smart cities

Article

Enhancing Service Quality of On-Demand Transportation Systems Using a Hybrid Approach with Customized Heuristics

Sonia Nasri [1,*], Hend Bouziri [2] and Wassila Aggoune Mtalaa [3]

[1] LARODEC Laboratory, Business Higher School of Tunis, Manouba University, La Manouba 2010, Tunisia
[2] LARODEC Laboratory, Higher School of Economics and Business, Tunis University, Tunis 1938, Tunisia; hend.bouziri@gmail.com
[3] Luxembourg Institute of Science and Technology, L-4362 Esch-Sur-Alzette, Luxembourg; wassila.mtalaa@list.lu
* Correspondence: sonia15.nasri@gmail.com

Abstract: As customers' expectations continue to rise, advanced on-demand transport services face the challenge of meeting new requirements. This study addresses a specific transportation issue belonging to dial-a-ride problems, including constraints aimed at fulfilling customer needs. In order to provide more efficient on-demand transportation solutions, we propose a new hybrid evolutionary computation method. This method combines customized heuristics including two exchanged mutation operators, a crossover, and a tabu search. These optimization techniques have been empirically proven to support advanced designs and reduce operational costs, while significantly enhancing service quality. A comparative analysis with an evolutionary local search method from the literature has demonstrated the effectiveness of our approach across small-to-large-scale problems. The main results show that service providers can optimize their scheduling operations, reduce travel costs, and ensure a high level of service quality from the customer's perspective.

Keywords: optimization method; evolutionary algorithm; tabu search; transportation systems; dial-a-ride problem; smart transportation; customized heuristics

check for updates

Citation: Nasri, S.; Bouziri, H.; Mtalaa, W.A. Enhancing Service Quality of On-Demand Transportation Systems Using a Hybrid Approach with Customized Heuristics. *Smart Cities* **2024**, *7*, 1551–1575. https://doi.org/10.3390/smartcities7040063

Academic Editor: Pierluigi Siano

Received: 27 April 2024
Revised: 31 May 2024
Accepted: 3 June 2024
Published: 26 June 2024

1. Introduction

The integration of on-demand transportation within smart cities represents a significant advancement, enhancing the flexibility, efficiency, and responsiveness of urban transport networks. By using advanced algorithms and optimized routing strategies, on-demand services contribute to optimizing resource utilization, reducing operational costs, and improving overall user satisfaction.

In this study, we focus on the dial-a-ride problem (DARP), which is a specific variant of on-demand transportation that focuses on passenger-oriented collective on-demand transportation [1]. It is classified as an NP-hard problem, as demonstrated by [2]. The optimization of service quality and cost-efficiency is crucial in this problem. High-quality service, characterized by reduced waiting times, reliable schedules, and customer-centric features, directly impacts user satisfaction and system efficiency. In smart cities, where technological advancements are pivotal in creating responsive and adaptive transportation networks, enhancing service quality can lead to more efficient resource utilization and substantial cost savings. In smart cities, optimizing cost-efficiency leads to reduced operational expenses, such as vehicle total riding time and travel distance, which directly impacts user satisfaction and system reliability. By minimizing costs, we not only ensure financial sustainability but also improve resource utilization and service responsiveness. Thus, prioritizing cost-efficiency alongside service quality supports the development of adaptive, efficient, and user-centric transportation networks, essential for smart city environments. Thus, enhancing service quality in on-demand transportation not only enhances the user experience but also supports the financial sustainability and operational efficiency of smart city transport networks. Therefore, the practical applicability of resolution methods for on-demand transportation problems has gained substantial attention.

In modern transportation systems, customer expectations have evolved to include several specific and nuanced aspects beyond traditional service quality. These new expectations include customers now expecting services tailored to their individual needs and preferences, such as specific pickup and drop-off points, preferred routes, and maximal riding time. To address these evolving expectations, authors [3] emphasize the importance of exploring problem variants that incorporate real-life service level criteria. Notable works in this area include studies of [4–12]. One specific problem variant that has gained attention is the customer-oriented dial-a-ride problem (CO-DARP), introduced by the authors in [12]. This problem incorporates constraints from a customer-oriented quality of service perspective and takes into account real-life customer expectations. In this model, customer preferences like preferred pickup and drop-off locations, as well as customer-dependent maximal riding time, are used for providing time windows constraints, and optimizing routes and schedules.

Meeting the individual preferences and needs of passengers contributes to an enhanced service quality. This entails providing choices for route customization, vehicle preferences, and accommodating passengers' special needs or preferences, such as their desired maximum travel time, preferred time windows, and arrival times. For example, time windows for delivery locations are adjusted based on the preferred maximum travel time and selected time windows for pickups. Regarding the consideration of real constraints on passenger and vehicle waiting times, we highlight the importance of balancing service efficiency with practical constraints. Time window constraints may produce limited passenger waiting times to ensure timely pickups and limited vehicle waiting times to minimize operational costs. Moreover, by incorporating these constraints into our approach, we aim to provide solutions that not only optimize service delivery times but also enhance overall service quality by addressing the diverse needs and preferences of passengers while considering real-world constraints on waiting times for both passengers and vehicles.

Furthermore, the efficiency of tabu search heuristics in solving a large set of dial-a-ride problems has been extensively demonstrated. Studies such as [1,13–15] have shown that the tabu search yields impressive results when applied to artificial cases of on-demand transportation problems. The ability of the tabu search to effectively explore the solution space and navigate complex constraints makes it a popular choice for tackling DARPs. Moreover, a substantial field of research has focused on the development of hybrid evolutionary methods, combining evolutionary computation with local search (LS) techniques. Contributions in this field include works by [4,7,16–19]. These hybrid evolutionary methods emphasize the strengths of both evolutionary algorithms and local search heuristics to improve the efficiency and effectiveness of optimization in solving on-demand transportation problems. Authors in [4] introduced a hybrid algorithm named the evolutionary local search (ELS) that combines the power of evolutionary computation and local search-based techniques.

In the existing literature, several approaches have been proposed to address the dial-a-ride problem in current transportation systems. These approaches have successfully tackled various aspects of the problem, such as optimizing routing and scheduling, integrating customer preferences, and managing resource allocation. These systems are increasingly incorporating customer exceptions and unique requirements into the design of new transportation problems. Therefore, new methods should focus on not only minimizing transportation costs but also enhancing service quality. These methods should be designed to support the new specifications of transportation systems, which introduce customer exceptions and prioritize customer satisfaction. By addressing both cost and service quality, these methods will provide transportation systems with the ability to meet evolving requirements and deliver an optimal customer experience.

We aim to build upon these advancements by offering an innovative approach that combines advanced optimization algorithms with customized techniques, providing even more efficient and flexible management of the CO-DARP in smart city transportation environments. Therefore, a novel method called the evolutionary tabu search (ETS) is proposed in this paper. This hybrid method combines the strengths of evolutionary algorithms and tabu search techniques, enabling the development of effective routing plans that optimize both the operational efficiency and the quality of service provided. New optimization techniques are proposed supporting advanced designs of service quality supported by the mathematical model. The latter involves the consideration of various factors such as wait times, travel times, and personalized preferences in the routing plans, ensuring a high-quality service that caters to the specific needs of customers.

- A customized insertion heuristic is specifically designed to define an initial feasible vehicle's plan with good quality. The use of this heuristic is claimed to be suitable for real-life on-demand transportation problems.
- A new tabu search method is introduced, which incorporates a novel customized neighborhood strategy. This neighborhood strategy is specifically designed to enhance both the operational costs and the service quality.
- Specific mutation operators are designed to address the characteristics and requirements of the problem at hand. These operators are dynamically exchanged to contribute to the discovery of high-quality solutions with improved service quality criteria.
- A specific crossover operator is proposed, which is carefully designed to inherit and preserve the advantageous service quality features discovered during the search. This contributes to the generation of new solutions that not only maintain the desired service quality features but also introduce variations that can potentially lead to further improvements.

The results of the comparative study indicate that the ETS method demonstrates a competitive performance when compared with the ELS method. Particularly, when it is applied to small-, medium- and large-scale realistic transport on-demand problems, the ETS method shows notable improvements in the service quality provided to the customers.

The paper is organized as follows: Section 2 presents a literature review of the contributions linked with the present study. Section 3 describes a customer-oriented DARP. Section 4 presents the proposed evolutionary tabu search approach for computing solutions. Section 5 reports the numerical results obtained with the new proposed metaheuristics on benchmark instances, together with a comparative study with another hybrid evolutionary method from the literature. A statistical analysis is conducted in Section 6. A discussion is provided in Section 7. Concluding remarks are set in Section 8.

2. Literature Review

Nowadays, the expectations of customers are increasingly complex, and anterior on-demand transport models should be enhanced [20]. Therefore, a few years ago, a new research field in DARP resolution emerged towards more realistic models with more rigorous constraints related to customer satisfaction. Authors in [3] reviewed various categories of real-life DARP features related to heterogeneity [21,22], routing properties [7,23], and quality specifications. In this field of quality specifications, the authors in [24] proposed a deeper survey on models being used by advanced research studies involving service quality designs from the customer's point of view. This research is in relation to maximal riding time (e.g., [4,11,25,26]) and time windows (e.g., [27–29]).

Authors in [30] highlighted the impact of variations in the level of quality of service provided to customers in the operational costs of a DARP. They studied the evolution of the operational costs through the investigation of the permutations of two sets of parameters describing the service level. The maximum deviation from the user preference time and the relative maximum exceedance of the user's direct journey time is measured according to the variation of certain parameters, such as the size of the transmission network and the diversity of customers. Moreover, the authors validated the potential trade-off between the costs and the quality of service according to variations in the service level parameters being frequently used, such as the size of the time windows and the maximal riding time. These two parameters are also emphasized in the works of [31,32], which reviewed and defined several measures related to the quality of service. These studies highlighted the maximal riding time and the time windows as the main criteria linked with the level of service that influence the total operational costs, and pointed out the advantage of combining them into the same model.

The quality of service can be expressed differently. In the works of [4,11,33], the maximum ride time is specified for each customer and used to calculate time windows. The need for the time windows redesign was emphasized by [28], using the maximal ride time as a parameter.

Other parameters can also be used in designing the service quality with a high level of customer-related criteria and preferences. In this regard, a problem named the customer-oriented DARP was proposed by [12], offering a design that provides more flexibility to service providers in producing routing plans closer to customer preferences.

Intelligent transportation problems have been addressed with techniques based on evolutionary computations. Beyond the efficiency of these tools, hybridization approaches combining evolutionary algorithms and local search methods were privileged. The union of these methods provides benefits in diverse settings [34]. Therefore, their use should be preferred to other methods when seeking better solutions for advanced on-demand transport problems, especially for more realistic DARPs designed from a customer-oriented viewpoint.

Recent advances in the tabu search showed the effectiveness of tabu search techniques in addressing dial-a-ride problems across various contexts and scales (e.g., [14,29,35–38]). These methods enhance the flexibility, efficiency, and adaptability of transportation systems, addressing the evolving needs of urban populations. Evolutionary computation for transportation problems is highlighted in [39]. Many works are provided for DARPs using evolutionary algorithms (e.g., [17,40–48]). These methods effectively handle the high-dimensional search spaces and diverse constraints of DARPs, leading to improved route efficiency and reduced operational costs. By simulating natural selection processes, they offer robust solutions adaptable to demands of on-demand transportation systems.

Authors in [34] emphasized the importance of combining genetic and tabu search algorithms. Thus, a large field of research is devoted to hybrid evolutionary methods for customized DARPs, see the existing related works of [4,7,16–19,46,49,50]. Authors in [4] proposed a hybrid evolutionary local search-based algorithm. This method was firstly proposed for solving DARPs of [1] with a customer-oriented maximal ride time. To analyze the service providers' cooperation better, more real-life characteristics are added to the artificial instances. This ELS method is also proposed for solving real-life instances of on-demand transportation [51]. However, to address real-life cases of on-demand transportation problems better, more advances are required when proposing optimization methods involving optimization techniques that support high-level designs.

Efficient and adaptive transportation solutions are required, driven by increasing urbanization and evolving user expectations. Hybrid evolutionary algorithms offer a promising approach to solving DARPs due to their ability to combine the strengths of multiple optimization techniques. They enhance the search process by the good exploration and exploitation capabilities of local search methods such as the tabu search. This is why we propose to combine the evolutionary schema with a tabu search to tackle customer-oriented DARPs in this present work.

3. The Customer-Oriented DARP

The customer-oriented DARP consists in satisfying on-demand transportation requests including the customer requirements in the decision taken. A passenger request is a transportation demand from a pickup location to a delivery one, which represents, respectively, the original location of the passenger and his/her destination. A set of passengers may be concerned with the same request which is also called a customer in this problem. This set is the same for the pickup and the delivery location. A homogeneous fleet of vehicles starts from the depot and arrives at the location. Note that neither demands nor waiting times are considered at this depot. In this problem, some constraints are aligned with those of the original DARP of [1], as follows.

- A request is served exactly once by one vehicle.
- The precedence between pickup and delivery stations is ensured.
- The number of pickups is equal to the number of deliveries in a vehicle tour.
- It is the same vehicle that ensures the pickup and delivery of a request.
- The depot is the departure and arrival station of all the tours.
- The capacity of a vehicle is respected when charging and discharging passengers.
- The maximal total tour duration is a time limit that has to be respected by all the vehicles.

Customer-oriented constraints are designed in the CO-DARP. These constraints are related to maximal riding time and time windows, which are commonly argued as the main attributes for the quality of service in on-demand transportation problems.

3.1. The Problem Formulation

The customer-oriented DARP is defined on a complete weighted digraph, $G(N,E)$, with the set N of nodes and the set E of connected arcs (i,j). The set of nodes, N, includes all $i \in \{0...2n+1\}$. This set includes the depot in two copies denoted by $(i=0)$ at the departure, and $(i=2n+1)$ at the arrival, the set of pickup nodes $\{1...n\}$, and the set of delivery nodes $\{n+1...2\text{n}\}$. The notations of the problem are indicated in Table 1. To each request corresponds a pickup node, i, which is its origin, and a delivery node $(i+n)$, which is its destination. Thus, the set of requests can be limited to that of the pickup nodes.

Table 1. Notations of the problem.

Notations	Meanings
n	Total number of requests.
m	Total number of vehicles.
v	Index of a vehicle, v, in $\{1...m\}$.
n_v	Total number of visited nodes by a vehicle, v, except the depot.
N_v	Set of nodes visited by a vehicle, v, from the depot to it.
$N_v(k)$	A node visited by v on a position $k \in \{0..n_v\}$.
λ	Positive penalty coefficient.
q_i	Number of passengers loaded at node i.
Q	Maximal capacity of the vehicles.
$t_{(i,j)}$	Transit time on an arc (i,j).
d_i	Time spent while loading passengers.
A_i	Arrival time at node i.
W_i	Waiting time at node i.
D_i	Departure time at node i.
TD_v	A total duration of a tour with the vehicle v.
Max_{TD}	Maximal total tour duration of the vehicles.
L_i	The maximal ride time for each request $i \in \{1...n\}$.
R_i	The riding time of a request $i \in \{1...n\}$.
B_i	Beginning of service at the origin of a request $i \in \{1...n\}$.
$B_{(i+n)}$	Beginning of service at the destination of a request $i \in \{1...n\}$.
$[e_i, l_i]$	Time window at the origin of a request $i \in \{1...n\}$.
$[e_{(i+n)}, l_{(i+n)}]$	Time window at the destination of a request $i \in \{1...n\}$.

A cost on an arc (i,j) is denoted by $C_{(i,j)} = (t_{(i,j)} + d_i)$, where $t_{(i,j)}$ is the transit time on an arc (i,j) and d_i is the time spent between the beginning of service and the departure of the vehicle for loading the passengers.

The total travel cost is the sum of all the costs of the visited arcs and the produced penalized waiting times at the visited nodes. A penalty coefficient, λ, is applied whenever a waiting time, W_i, occurs. Let us consider $f(S)$ as the objective function (1) to minimize. It consists of the total travel cost (TTC) related to all the vehicle tours considering waiting time penalties.

$$min\ f(S) = \sum_{v=1}^{v=m} \sum_{k=0}^{k=n_v} C_{(N_v(k),N_v(k+1))} + \lambda W_{N_v(k)} \tag{1}$$

The maximal capacity of a vehicle must be checked in the pickups and the deliveries. The number of passengers denoted by q_i is positively incremented in the case of a pickup and negatively otherwise. Thus, the inequality (2) explains that the number of passengers aboard a vehicle must not exceed its maximal capacity, Q.

$$\sum_{i \in N_v} q_i \le Q \qquad \forall v \in \{1 \ldots m\} \tag{2}$$

In this study, particular attention is given to the design of time window constraints. With this concern, the transportation plans are adjusted to the customers' needs while ensuring adaptability to the existing realistic on-demand transport situations. Time windows at the origin nodes are defined by the inequality (3), and time windows at the destination are obtained by (4) which allow also defining the lower bound and the upper bound for the beginning of service, B_i. In this customer-oriented DARP, the service duration, d_i, is supposed equal to the total number of passengers, q_i. Thus, we have ($d_i = q_i$) for all pickup and delivery nodes. However, it is usually considered positive when computing the quantities (3), (4). These time window equations contribute to reducing unnecessary waiting times as well as enhancing the total riding time. More details are provided in [12].

$$min(e_{(i+n)} - d_i - L_i; e_i) \le B_i \le min(l_{(i+n)} - d_i - R_i; l_i) \quad \forall \quad i \in \{1 \ldots n\} \tag{3}$$

$$min(e_i - d_i - L_i; e_{(i+n)}) \le B_{(i+n)} \le min(l_i - d_i - R_i; l_{(i+n)}) \quad \forall \quad i \in \{1 \ldots n\}; \tag{4}$$

The expression (5) limits the maximal riding time, L_i, as a bound linked with each customer $i \in \{1...n\}$. Thus, the riding time, R_i, which is equal to the time spent aboard a vehicle from an origin, i, to a destination $(i+n)$ must respect this customer-related bound.

$$R_i \le L_i \quad \forall i \in \{1 \ldots n\} \tag{5}$$

Equation (6) expresses the riding time at node i, being equal to the time between the departure time, D_i, at the origin and the beginning of service, $B_{(i+n)}$, at the destination. Equation (7) computes the departure time at node i. Note that the departure time at the depot is equal to the beginning of service. Thus ($B_0 = D_0$) and ($B_{(2n+1)} = D_{(2n+1)}$) since no person enters aboard the vehicle and no service time is assumed.

$$R_i = B_{(i+n)} - D_i \quad \forall i \in \{1 \ldots n\} \tag{6}$$

$$D_i = B_i + d_i \quad \forall i \in N \tag{7}$$

The quantity W_i in Equation (8) defines potential waiting times. Negative values of waiting times are not allowed since, in DARP, a vehicle can wait but a customer should not; i.e., $A_i <= B_i$. The depot is not concerned by any waiting time since we have ($B_0 = A_0$) and ($B_{(2n+1)} = A_{(2n+1)}$). The arrival time at node i is the travel time from the depot to it.

$$W_i = B_i - A_i \quad \forall i \in N \tag{8}$$

For each vehicle tour, we derive a beginning of service time, denoted as B_i, while adhering to a specified time window, i.e., $B_i \in [e_i, l_i]$, where i and j represent successive nodes visited within a vehicle tour, v. It is worth noting that the beginning of service at a pickup node $i \in \{1..n\}$ is calculated as $(B_i = B_{i+n} - d_i - R_i)$. The arrival time at node $j \in N_v$ is calculated as $(A_j = B_i + d_i + t_{(i,j)})$, where j is a successor node to node $i \in N_v$. The total tour duration, TD_v, for a vehicle is then defined as the time spent from the depot to it $(TD_v = A_{(2n+1)} - B_0)$. This TD_v is constrained by a maximal common total duration, Max_{TD}, as defined in the problem.

3.2. A Time Windows Application on a Realistic Case

To illustrate the concept of time windows, we draw upon a real-life example from problem instance $d71$ found in the data provided by Chassaing et al. [51], as detailed in Section 5. Initially, this problem was tackled using an evolutionary method known as ELS [4], as explained in Section 2.

From the solutions obtained, we selected a vehicle tour characterized by a notably high waiting time. Subsequently, we present two scenarios: one before the introduction of customized time window specifications and one after.

Within this tour, we address two specific requests, denoted as $(i = 1)$ and $(i = 2)$, each with respective loads of $(q_1 = 2)$ and $(q_2 = 4)$. These requests are subject to two maximum riding times, namely $(L_1 = 97)$ and $(L_2 = 76)$.

All vehicles are equipped with a maximum capacity constraint of $(Q = 8)$ and are bound by a maximal total tour duration of $(Max_{TD} = 480)$.

Figure 1 illustrates an on-demand transport problem involving two requests, denoted as $(i = 1)$ and $(i = 2)$, along with the depot. In the figure, we display the customers' input parameters, including their loads, maximal riding times, and time windows.

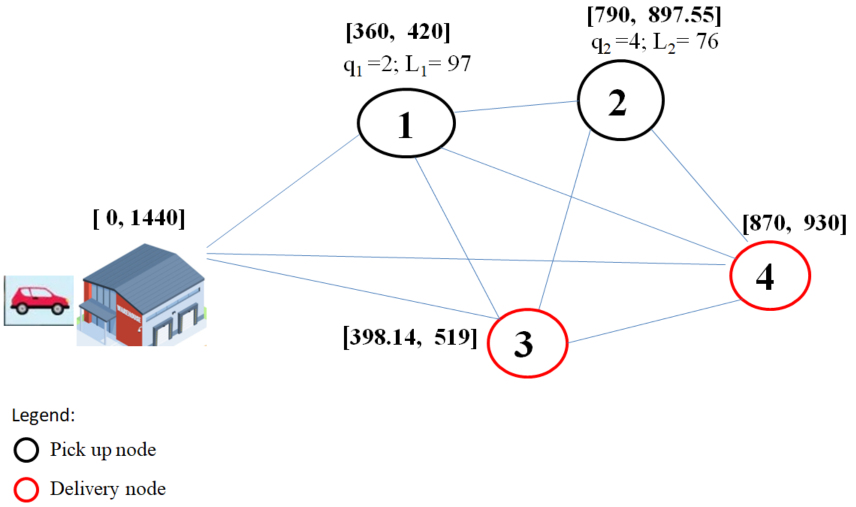

Figure 1. An on-demand transport network with two requests having time windows, maximal ride times, and loads.

In Table 2, we present the parameters associated with the vehicle's tour example before the computation of new customized time windows. The first request $(i = 1)$ assigned to the vehicle involves a pickup node $(i = 1)$ and a delivery node $(i = 3)$. Similarly, the second request comprises a pickup node $(i = 2)$ and a delivery node $(i = 4)$. For each visited node $(i \in N_v)$, the table displays load (q_i), transit time $(t_{(i,j)})$, arrival time (A_i), beginning of service (B_i), and departure time (D_i).

At the departure node $(i = 0)$, we have $(B_0 = A_0 = D_0)$, and, at the final arrival node $(i = 5)$, we have $(B_5 = A_5 = D_5)$, since no waiting time or load considerations are applicable at the depot.

Table 2. The vehicle tour specifications.

N_v	q_i	$t_{(i,j)}$	A_i	B_i	D_i
0	0	12.5	407.05	407.05	407.05
1	2	36.14	420	420	422
3	−2	9.48	458.14	458.14	460.14
2	4	28.45	469.62	837.55	841.55
4	−4	3.2	870	870	874
5	0	0	877.2	877.2	877.2

Arrival times are computed at each visited node. For example, the arrival time A_3 is calculated as ($A_3 = B_1 + d_1 + t_{(1,3)}$), resulting in ($420 + 2 + 36.14 = 458.14$). Departure times are determined using (7), with D_3 computed as ($D_3 = B_3 + d_3$), which equals ($458.14 + 2 = 460.14$).

Riding times for the two requests are calculated as ($R_1 = 36.14$) and ($R_2 = 28.45$) using (6). The waiting time only applies to node ($i = 2$), with ($W_2 = 367.94$), determined as ($837.55 - 469.62 = 367.94$) using (8). This waiting time, W_2, affects the arrival time at the subsequent node ($i = 4$), as shown in Figure 2.

Figure 2 provides a visual representation of the vehicle, v, following the tour of nodes in N_v, as outlined in Table 2.

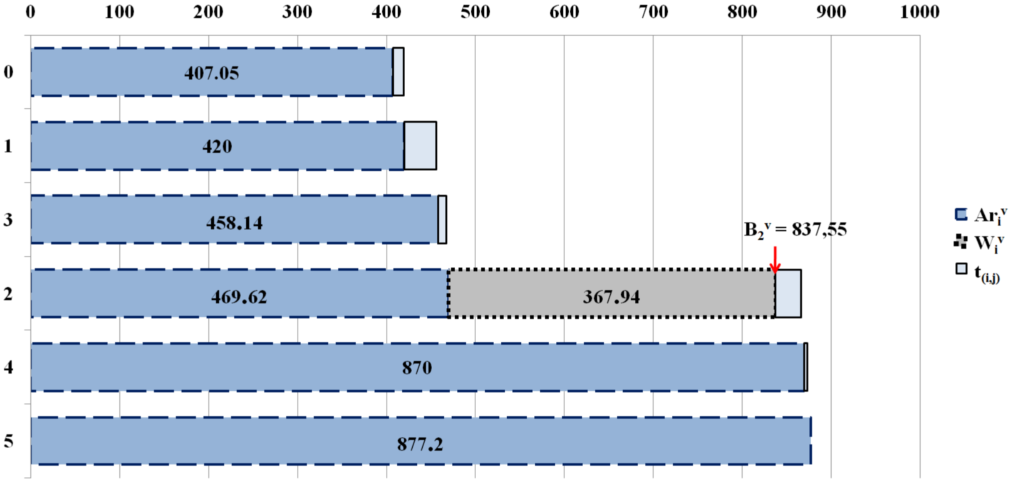

Figure 2. The vehicle tour schedule before the time windows calculation.

The total tour duration, TD_v, is computed as ($877.2 - 407.05 = 470.15$) at the conclusion of the tour, following the formula $(TD_v = A_{(5)} - B_0)$. This total duration ensures compliance with the maximum limit ($Max_{TD} = 480$).

In the case of customized time window specifications, new bounds are determined to restrict waiting times. Initially, time windows for delivery nodes are calculated using (4). These resulting bounds are then utilized to establish time windows for origin nodes through (3). Consequently, Table 3 displays new time windows derived from inequalities (3) and (4), resulting in the selection of updated early service times.

We specifically focus on the time windows associated with request ($i = 2$), as it was the node where waiting times occurred. The new beginning of services, denoted as B_2 and B_4, is determined at the pickup and delivery nodes of request 2. Let us note that the initial time windows for these nodes were initially set as $[790, 897.55]$ and $[870, 930]$, as indicated in Figure 1.

In Table 3, the new beginning of service $B_4 \in [710, 865.1]$ is selected as the earliest service time at the delivery node ($i = 4$). Consequently, the beginning of service B_2 is calculated as ($B_2 = 710 - 4 - 28.45$), given that ($B_2 = B_4 - d_2 - R_2$). Thus, $B_2 = 677.55 \in [630, 832.66]$ is the new beginning of service at the pickup node ($i = 2$).

This earlier beginning of service contributes to the reduction in the waiting time. The new value of W_2 is then ($B_2 - A_2 = 677.55 - 469.62 = 207.93$). The arrival time, A_4, is also impacted. As ($A_4 = 469.62 + 207.93 + 28.45$), where 28.45 is the transit time $t_{(2,4)}$ and 207.93 is the reduced waiting time ($W_2 = 207.93$), then ($A_4 = 706$). This new arrival time will produce a low waiting time ($W_4 = 4$). However, the new total waiting time is equal to ($W_2 + W_4 = 207.93 + 4 = 211.93$), which is much lower than the initial one equal to ($W_2 = 367.94$).

Table 3. The customized design of the time windows related to request 2.

Nodes	e_i	l_i	B_i
2	min (710 − 4 − 76;790) = 630	min (865.1 − 4 − 28.45;897.55) = 832.65	677.55
4	min (790 − 4 − 76;870) = 710	min (897.55 − 4 − 28.45;930) = 865.1	710

As reported in Table 3, with new time windows reduced, as compared with those proposed in Table 2, the quality of the service is enhanced in terms of waiting time. The vehicle tour after the new time windows calculation is illustrated in Figure 3. Only the beginning of service producing a waiting time is indicated in the figure.

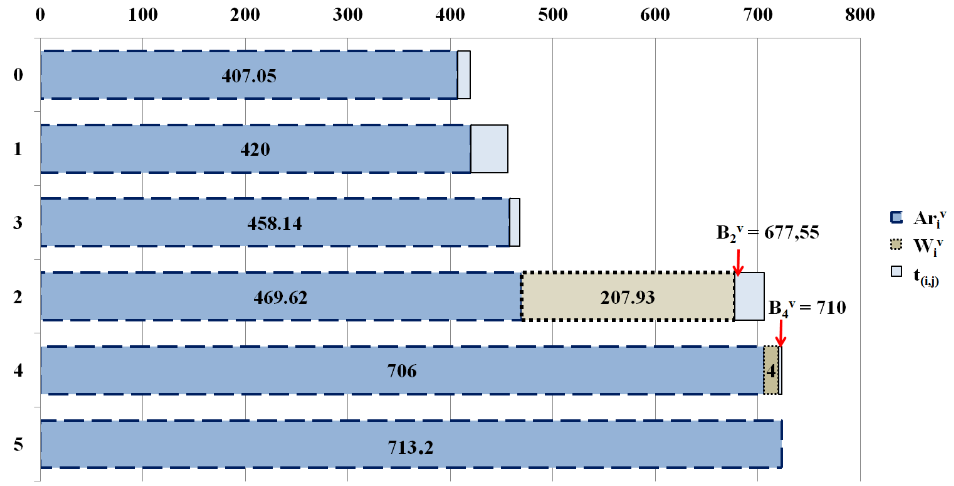

Figure 3. The vehicle tour schedule after the time windows calculation

The limitation of the waiting time leads to reducing the total duration, TD_v from ($TD_v = 470.15$) in Figure 2 to ($TD_v = 306.15$) in Figure 3. The enhanced TD_v is obtained as ($A_5 - B_0 = 713.2 - 407.05$), where ($A_5 = A_4 + W_4 + t_{(4,5)}$) and ($t_{(4,5)} = 3.2$), see Table 2.

Through the above example, we have analytically demonstrated how the application of customer-oriented constraints in problem resolution can improve service quality, specifically reducing waiting times. Furthermore, this enhancement in service quality contributes to minimizing the total duration, which is advantageous for the service provider. The following sections explain the investigation of time window constraints within the optimization techniques of the ETS.

4. The Evolutionary Tabu Search

This section defines the evolutionary tabu search algorithm with special genetic operators and a tabu search-based method.

4.1. Overview of the Evolutionary Tabu Search Algorithm

The evolutionary tabu search starts by generating an initial solution by the use of an insertion heuristic (see Section 4.2). In order to create individuals composing the first population, a mutation phase is performed where the initial solution is perturbed, producing a set of individuals. Next, each mutated solution in the population is modified by the use of a crossover named self crossover (SXO), provided in Section 4.5. In order to seek improvement in the neighborhoods of each solution, the TS method (see Section 4.6) is applied to each individual. To deal with some constraint violations, a repair procedure (see Section 4.3) is operated on each individual in the TS. Then, the obtained solutions are assessed and the best individual is selected and used for the next iteration in the ETS. This individual is also compared with the potential final solution in the search space. If it improves, it is saved as a potential new solution. In Figure 4, a chart presents the successive steps of the ETS algorithm.

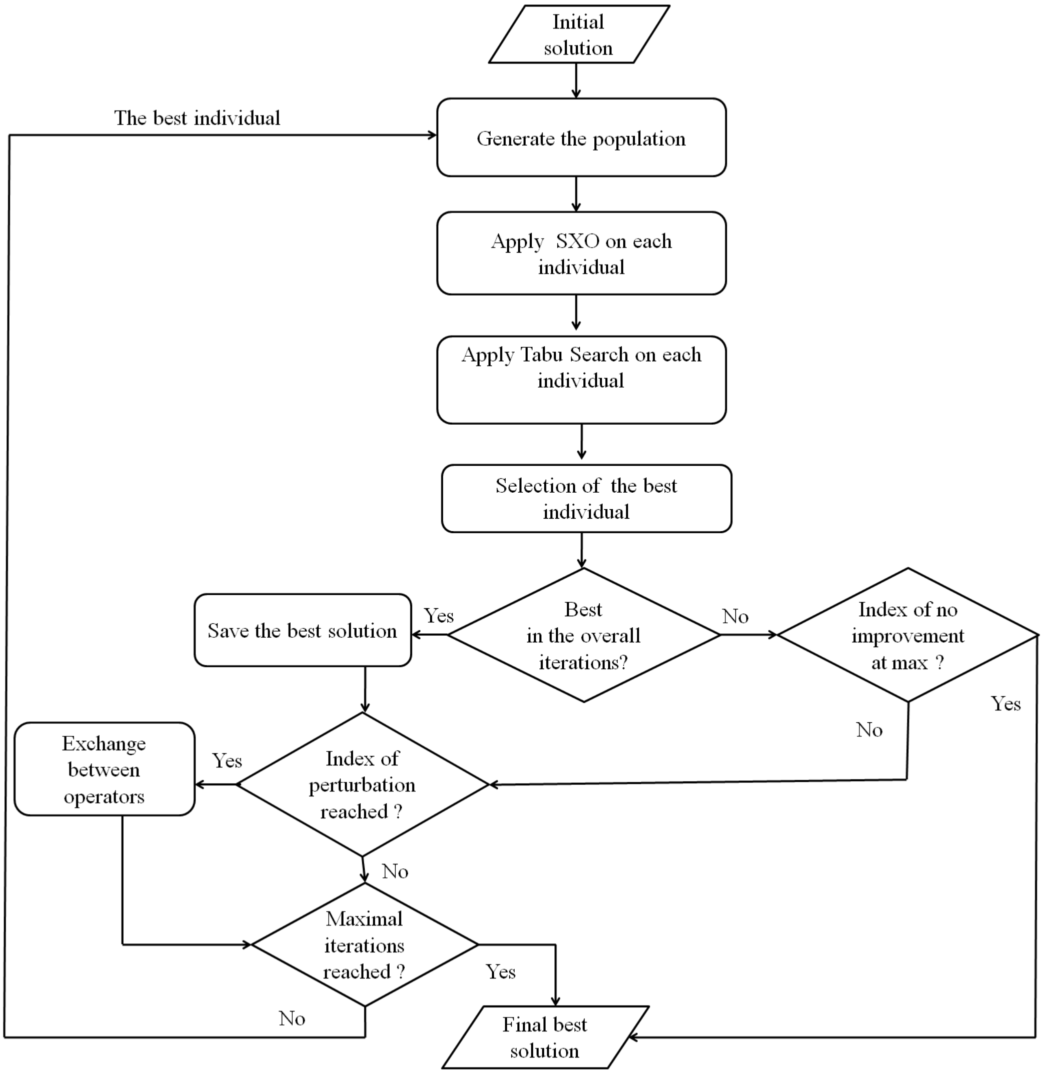
Initial solution
The best individual
Generate the population
Apply SXO on each individual
Apply Tabu Search on each individual
Selection of the best individual
Best in the overall iterations?
Yes
No
Save the best solution
Index of no improvement at max ?
Yes
No
Exchange between operators
Yes
Index of perturbation reached ?
No
Maximal iterations reached ?
Yes
No
Final best solution

Figure 4. Flow chart of the ETS algorithm.

Furthermore, in order to increase the perturbation of the search within the ETS, the mutation operators (see Section 4.4) are exchanged after a fixed number of ETS iterations controlled by an index of perturbation. In addition, another index stops the search when no improvement is made during a set of iterations. Otherwise, the overall algorithm stops when the maximal number of iterations is reached.

4.2. The Insertion Heuristic

To construct the initial solution, we propose a new insertion heuristic, see Algorithm 1.

Algorithm 1 The insertion_heuristic

```
 1: Initialize the list of non-allocated requests sorted in the increasing order of the earliest
    service time;
 2: Initialize the set of tours as empty and all the vehicles as available;
 3: repeat
 4:     Step 1: Select a request from the list of the sorted non-allocated ones;
 5:     Step 2: Select an available vehicle;
 6:     if there is a valid case of insertion in the tour then
 7:         Insert the pickup node in a valid position in the tour;
 8:         Update the vehicle's capacity and total duration;
 9:         Delete the request from the list of non-assigned requests;
10:         Update the time schedule; go to Step 1;
11:     else
12:         if there are potential violations of capacity or MRT then
13:             Deliver some requests ;
14:             Update the capacity and total duration;
15:             Update the time schedule ; go to Step 1;
16:         else
17:             if The vehicle is at its maximal total duration then
18:                 Set the vehicle as non-available; go to Step 2;
19:             else
20:                 go to Step 2;
21:             end if
22:         end if
23:     end if
24: until All the requests are satisfied;
25: returns a solution as a result;
```

The heuristic starts by allocating the requests according to the ascending order of their earliest service times. Therefore, it proceeds to the selection of an appropriate vehicle related to its availability capacity and total duration. The requests are inserted in the tour of the vehicles within an available time schedule and when the problem constraints are respected. When a request is allocated (line 7), all the related parameters in relation to vehicles, requests, and time schedules are updated (lines 8–10). In this regard, time windows are computed by the use of Equations (3) and (4) in the problem formulation of Section 3.

In order to release some vehicles' capacity, deliveries are performed in accordance with their earliest service times and all necessary updates are performed (lines 14–15). This action is also ensured when there are maximum riding time violations (which therefore can be violated during the new request's insertion) in the tour. Furthermore, a vehicle is set as non-available when its maximal total tour duration is reached. Finally, when all the requests are satisfied, the process comes to an end.

4.3. Repair Operator

Since the metaheuristics are stochastic during the search for better solutions, constraints may be violated. The constraints that may be violated can be linked to the vehicle capacity, the maximal riding time, and the maximal total duration. Therefore, to ensure the feasibility of the solutions, a repair procedure is performed.

To avoid changing the structure of the solution too much, the repair procedure tries to apply a change in the same tour. Thus, in the case of a maximum riding time violation, the delivery schedule is altered by reinserting the destination's node into another position. With respect to the problem constraints, the same repair action is operated when there is a violation of capacity. Whenever the repaired constraint, the procedure updates the time schedule involving the time windows computation. However, if it is not possible to deal with the feasibility of a tour, the request is reinserted in another route.

In the case of the violation of maximal total duration, lasted requests in the tour are moved to other routes, ensuring the feasibility of the problem constraints in the routes.

4.4. The Mutation Operators

This section introduces both an interruption operator and a swap operator for the mutation phase. Two illustrative examples based on seven requests and two vehicles are provided.

4.4.1. The Interruption Operator

The interruption operator randomly selects a request with waiting time at either the pickup or delivery node. It then cuts the tour before the pickup node of the chosen request and removes that request and the subsequent ones from the route. Subsequently, it attempts to relocate other requests from different routes while adhering to problem constraints and scheduling times.

To facilitate the insertion of requests into another schedule, the operator relies on the customized time windows discussed in Section 3. An illustrative example, based on seven requests and two vehicles, is presented in Figure 5. In this example, the beginning of service for each node is indicated, with node ($i = 7$) initially assigned to vehicle $v1''$.

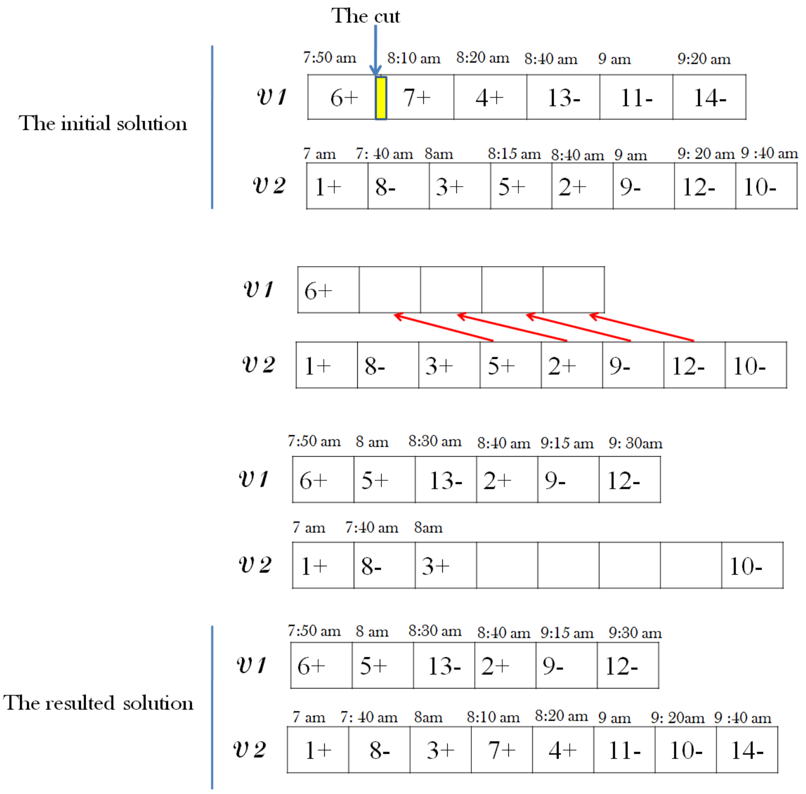
The cut
The initial solution
7:50 am 8:10 am 8:20 am 8:40 am 9 am 9:20 am
v1 6+ 7+ 4+ 13- 11- 14-
7 am 7: 40 am 8am 8:15 am 8:40 am 9 am 9: 20 am 9 :40 am
v2 1+ 8- 3+ 5+ 2+ 9- 12- 10-
v1 6+
v2 1+ 8- 3+ 5+ 2+ 9- 12- 10-
7:50 am 8 am 8:30 am 8:40 am 9:15 am 9: 30am
v1 6+ 5+ 13- 2+ 9- 12-
7 am 7:40 am 8am
v2 1+ 8- 3+ 10-
The resulted solution
7:50 am 8 am 8:30 am 8:40 am 9:15 am 9:30 am
v1 6+ 5+ 13- 2+ 9- 12-
7 am 7: 40 am 8am 8:10 am 8:20 am 9 am 9: 20am 9 :40 am
v2 1+ 8- 3+ 7+ 4+ 11- 10- 14-

Figure 5. Construction of a new solution with the interruption operator.

Initially, the tour of vehicle $v1$ is cut before node ($i = 7$), and the subsequent requests are removed from the tour. The procedure then reallocates requests from the tour of $v2$ to $v1''$, respecting constraints related to requests, vehicles, and scheduling.

Next, node 7 is inserted after node ($i = 3$) in the tour of $v2$. The procedure continues allocating requests in the tour of vehicle $v2$ until all requests are assigned. Time scheduling is managed through time window equations, ensuring a feasible routing plan with high-quality service in the resulting solution. This enhanced service quality is evident in the elimination of the waiting time previously observed at node 7 in $v1$.

4.4.2. The Swap Operator

The swap operator involves exchanging two consecutive delivery locations along the same route, provided that the beginning of service is respected. The procedure manages scheduling time using time window constraints (see Equations (3) and (4) in Section 3) and selects the maximum value between the beginning of service and the new arrival times. This technique enhances service quality when waiting times occur.

Figure 6 illustrates an example of a solution perturbed by the swap operator. The operation involves the two delivery nodes ($i = 13$) and ($i = 12$) in an illustrative example with seven requests and two vehicles. Beginning of service times are indicated for each node, with node ($i = 13$) having a service start time of 8:20 am and node ($i = 12$) starting at 8:40 am.

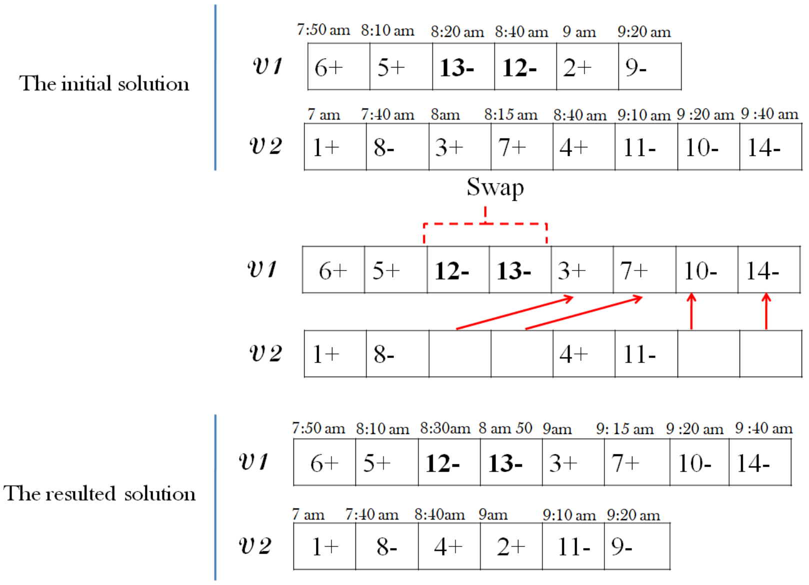
The initial solution
v1
7:50 am 8:10 am 8:20 am 8:40 am 9 am 9:20 am
6+ 5+ 13- 12- 2+ 9-
v2
7 am 7:40 am 8am 8:15 am 8:40 am 9:10 am 9 :20 am 9 :40 am
1+ 8- 3+ 7+ 4+ 11- 10- 14-
Swap
v1
6+ 5+ 12- 13- 3+ 7+ 10- 14-
v2
1+ 8- 4+ 11-
The resulted solution
v1
7:50 am 8:10 am 8:30am 8 am 50 9am 9: 15 am 9 :20 am 9 :40 am
6+ 5+ 12- 13- 3+ 7+ 10- 14-
v2
7 am 7:40 am 8:40am 9am 9:10 am 9:20 am
1+ 8- 4+ 2+ 11- 9-

Figure 6. Construction of a new solution with the swap operator.

As shown in Figure 6, the swap operation is executed while ensuring compliance with problem constraints, including the new time windows computation. Consequently, new beginnings of service are assigned to the two swapped nodes, resulting in nodes ($i = 12$) and ($i = 13$) having new beginnings of service at 8:30 am and 8:50 am, respectively.

Consequently, requests ($i = 3$) and ($i = 7$) are transferred to the vehicle tour of $v1$. The tour of vehicle $v2$ is adjusted to accommodate request 2 in its second route, with request 2 having pickup and delivery nodes ($i = 2$) and ($i = 9$), respectively. The time schedule is managed and updated using time window equations, considering the maximum values between the beginning of service and the newly provided arrival times at the nodes.

4.5. The Crossover Operator

The crossover operator aims to ensure that the offspring solution inherits positive features from both parents, particularly the good quality of service achieved in the preceding mutation phase.

In our SXO, the parents are not individual solutions, as in classical evolutionary algorithms. Instead, the parents are two selected routes from the same individual. The main steps of the crossover process include:

- Positioning: choose two positions on the two routes, linked to the first pickup nodes in the routes that have similar or close beginnings of service.
- Mapping: temporarily remove the remaining sequences of requests from routes outside the selected cut points in each parent.
- Switching: exchange the two parts between the cut positions between the two routes, respecting the same positions in the time schedule.

- Filling: copy the deleted requests into the child, trying to maintain their initial positions from the parent without introducing redundancy of nodes.

In the example of Figure 7, we consider seven requests and two vehicles. The beginning of service is indicated for each node. The cut is operated on two positions by considering a set of requests with their related pickup and delivery nodes. This criterion aims to ensure the feasibility of the precedence, vehicle capacity, and maximal riding times constraints.

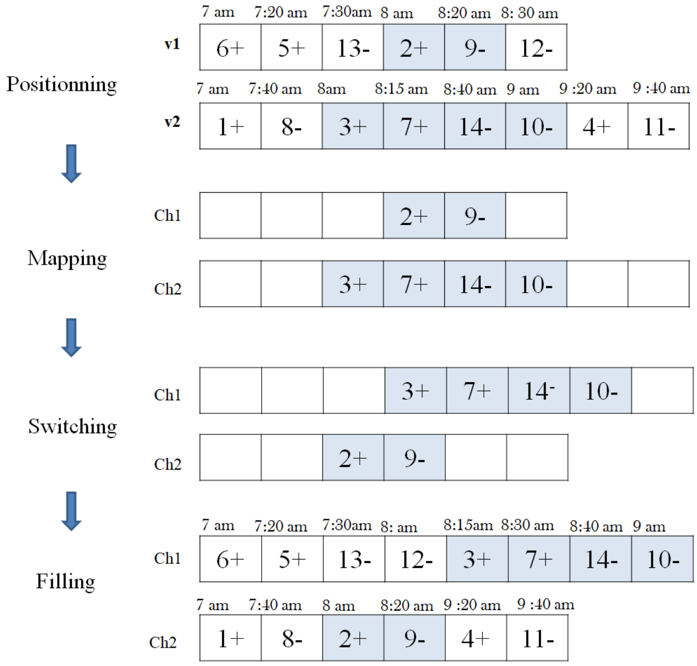
Positionning
v1
7 am 7:20 am 7:30am 8 am 8:20 am 8: 30 am
6+ 5+ 13- 2+ 9- 12-
v2
7 am 7:40 am 8am 8:15 am 8:40 am 9 am 9 :20 am 9 :40 am
1+ 8- 3+ 7+ 14- 10- 4+ 11-
Mapping
Ch1
2+ 9-
Ch2
3+ 7+ 14- 10-
Switching
Ch1
3+ 7+ 14- 10-
Ch2
2+ 9-
Filling
Ch1
7 am 7:20 am 7:30am 8: am 8:15am 8:30 am 8:40 am 9 am
6+ 5+ 13- 12- 3+ 7+ 14- 10-
Ch2
7 am 7:40 am 8 am 8:20 am 9 :20 am 9 :40 am
1+ 8- 2+ 9- 4+ 11-

Figure 7. An illustrative example of the SXO.

In Figure 7, the two blocs of requests {2,9} and {3,7,14,10} are selected from parents $v1$ and $v2$, respectively. The beginning of service in the first pickup, node 2, is 8 am, which is also the case for node 3. In the mapping step, only the two blocs exist in the children $Ch1$ and $Ch2$. Next, the blocs are switched between children and the corresponding positions are fulfilled. In the next step, vacant positions are filled by the remaining nodes, ensuring respect for the problem constraints. Whenever a waiting time occurs, the crossover proceeds to the time windows computation seeking its minimization.

4.6. The Tabu Search Method

The tabu search method is a well-known heuristic that was successfully used for DARPs (e.g., [1,13]). Therefore, we propose to develop a TS especially adapted to the customer-oriented DARP. Thus, a neighborhood strategy supporting the customized design is provided involving service quality enhancement.

In the TS solving method, the move operation involves altering the allocation of requests by changing the position of the request responsible for the largest waiting time within a trip. The request with the highest waiting time value is selected from a trip. Customized time windows for this request are computed using equations provided in Section 3, resulting in a new beginning of service.

Next, the procedure attempts to reinsert the request into another possible position that minimizes the total waiting time. If no waiting time is present in the solution, a random request is selected and moved to a different position. This process involves updates to parameters related to requests, vehicles, and the time schedule.

To maintain solution feasibility after the move operation, the repair procedure (Section 4.3) is applied to restore any violated constraints.

Figure 8 depicts an example of the move operation proposed in our TS. We present a case before and after the move operation. Two vehicles' routes are considered to satisfy three requests. These latter have origins 1, 2, and 3, with destinations 4, 5, and 6. The beginning of the service is marked on each node. In addition, there are two waiting times that are supposed for nodes 3 and 5, with (W_3^1 = 20) and (W_5^2 = 10), respectively.

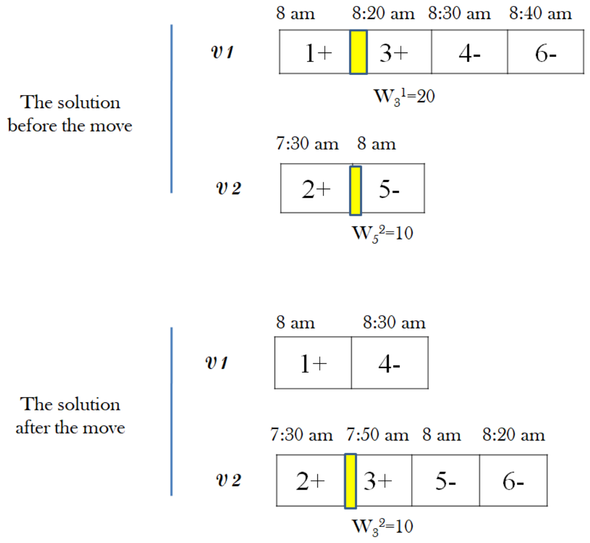

Figure 8. An illustrative example of the neighborhood strategy of the TS.

The node with the highest waiting time is node 3, corresponding to the pickup node of request 3 in the route of $v1$. Initially, this request had a beginning of service at 8:20 am for pickup and 8:40 am for delivery. After computing the time windows, new bounds are determined, which are 7:50 am for pickup and 8:20 am for delivery. To ensure the resulting time schedule, request 3 is inserted in route $v2$ to minimize waiting times.

This move operation effectively reduces or eliminates unnecessary waiting times, as seen in the case of node 5. Consequently, the total waiting time is reduced to 10 in the resulting solution.

In Figure 8, a new solution is presented, where nodes 1 and 4 are allocated to $v1$ consecutively, while nodes 2, 3, 5, and 6 are visited in succession by $v2$.

5. Computational Experiments

The experiments were conducted on an Intel(R) Core(TM) computer (2.4 GHz) with 4 GB RAM. The evolutionary tabu search algorithm is implemented in C++ and uses a single thread. In this section, a fair comparative study is performed between two hybrid evolutionary methods, which are the ETS and the ELS [4] (Section 2) using the same data set. Thus, we first explain the benchmark data set that is used in [51]. Next, the resolution of one particular instance is detailed to highlight the relevance of the customized design of time windows supported by the ETS. Then, larger experimentation is conducted on small, medium, and large instances of on-demand transportation. Finally, the main outputs are discussed based on a deeper analysis.

5.1. Benchmark Instances

The data set used in all the experimentation is based on the 96 instances of [51]. These instances represent realistic on-demand transport situations in urban areas. The main data features are reported here.

- The locations are marked by the real distances between them based on the GPS positions of French cities.
- The transportation network density is not the same for all the instances.
- The customers are distributed over a large area, varying according to the instances.

- Time windows are separately defined on both the pickup and destination locations. These time windows are of different sizes according to the customers.
- The maximum riding time for each customer depends on the customer and the distance they have to travel.
- The number of passengers in a location may be greater than 1 up to 4.
- All the vehicles have the same average speed equal to 1.33 km per minute.
- The maximal vehicle tour duration is equal to 480.
- The maximal capacity of the vehicles is equal to 8.

The distances between the origins and the destinations of the customers are distributed so that the majority of the requests (75%) need transportation of between 30 km and 70 km. In addition, the distribution of the maximum riding times is in relation to the travel distances requested by the customers. Thus, the majority of the customers (80%) require a maximal riding time equal to a markup of 100% of their distance. This defines precisely that the maximal riding time is customer-dependent but also depends on the distance of the journey, which reflects realistic situations.

Time windows are supposed to vary between 1 and 4 h. In addition, 50% of the time windows are of 1 h.

5.2. The Relevance of Customer-Oriented Time Windows in the Resolution by ETS

In this part, we analyze the effects of the customer-oriented design of time windows on both the service quality and the travel cost of a solution provided by the ETS. Precisely, we focus on showing the positive impact of the customized time windows formulation provided in Equations (3) and (4) in Section 3 on the waiting times and the total duration. Thus, we compare in-depth two solutions provided by our ETS and the ELS algorithms on the problem instance *d*75. As reported later in Table 4, the total number of requests is equal to 10 and the total number of vehicles is equal to 2. The solution obtained with the ELS has a travel cost equal to 150.91. However, with the ETS, we obtain a better cost equal to 143.22. Note that, for this instance, we have 20 nodes including ($i = 0$) and ($i = 21$), the two copies for the depot. Figure 9 presents the solution obtained with the ELS for problem *d*75. The travel cost of this solution, which is 150.91, is the sum of the cost 40.91 and 110 of the tours achieved with the two vehicles ($v = 1$) and ($v = 2$), respectively.

$v = 1$	5	8	18	15	9	19								
$v = 2$	10	3	20	1	13	2	11	4	14	6	12	16	7	17

Figure 9. The solution of the *d*75 instance obtained with the ELS.

Two cases of waiting times are produced in the solution of the ELS. These waiting times are ($W_9 = 107.44$) in ($v = 1$) and ($W_7 = 260.15$) in ($v = 2$). To demonstrate the positive effects of the customized time windows design, we emphasize one vehicle tour having a waiting time ($W_9 = 107.44$), as it is illustrated by Figure 10. In this figure, we show some of the nodes' time schedule related to the tour of ($v = 1$) of Figure 9. For more clarity, we focus on request 9 with only the depot.

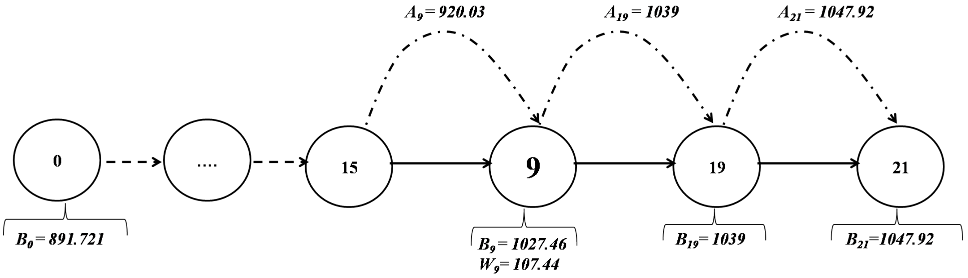

Figure 10. The schedule time for request 9 in the vehicle ($v = 1$) obtained with the ELS.

When we consider the solution provided by the ETS on the same problem instance $d75$ (see Figure 11), we can note that request 7 is moved from ($v = 2$) in the solution of the ELS (Figure 9) to ($v = 1$) in the solution of the ETS. This new schedule allows a reduced cost equal to 143.22 in the solution of the ETS, as compared with 150.91 for the ELS. Indeed, this request had a high waiting time ($W_7 = 260.15$) at the pickup node in ($v = 2$) of the ELS. This waiting time is totally deleted in the solution of the ETS when inserting request 7 in the schedule of ($v = 1$), as reported by Figure 12.

$v = 1$	7	17	5	8	18	15	9	19				
$v = 2$	10	3	20	1	13	2	11	4	14	6	12	16

Figure 11. The solution of the $d75$ instance obtained with the ETS.

Moreover, the waiting time for request 9 is also reduced. This enhancement is due to the time windows design using Equations (3) and (4) and the new bounds $[989, 1135.93]$ and $[964, 1124.39]$ derived for the pickup node ($i = 9$) and the delivery one ($i = 19$) of request 9. Consequently, two beginnings of service time are then proposed as ($B_9 = 977.46$) and ($B_{19} = 989$). This has also an impact on the overall system performance, producing new values for the vehicle's arriving times ($A_{19} = 989$) and ($A_{21} = 997.92$) in Figure 12, as compared with ($A_{19} = 1039$) and ($A_{21} = 1047.92$) in the ELS, see Figure 10.

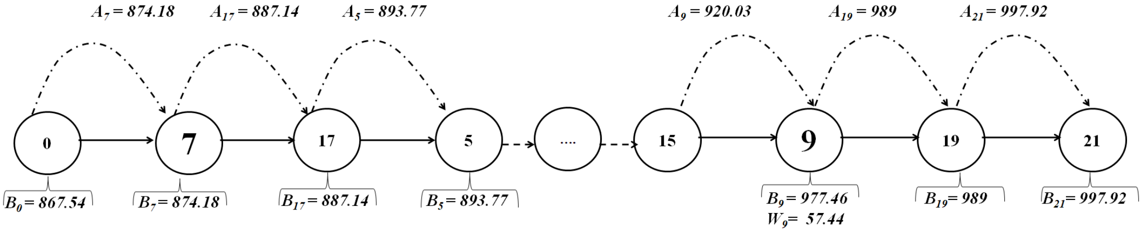

Figure 12. The details related to requests 7 and 9 in ($v = 1$) in the solution obtained with the ETS.

Consequently, we can argue that the customized design of the service quality term is beneficial for both the customer and the service provider. More precisely, in the solution produced by the ETS method, two levels of enhancement are highlighted: the solution cost in terms of completion time and the service quality in terms of waiting time.

5.3. Results Obtained with the Two Evolutionary Techniques on the Benchmark Problems

In this subsection, we show the ETS performance regarding on-demand transport problems of [51]. To analyze the behaviour of the ETS better, we propose to divide the instances into three groups, $C1$, $C2$, and $C3$, which, respectively, correspond to small, medium, and large instances. The sets of instances $C1$, $C2$, and $C3$ contain 32 instances each. The total number of requests is in $\{10..60\}$ for $C1$, $\{62..84\}$ for $C2$, and $\{85..128\}$ for $C3$.

In all the experimentation, the comparison is based on the deviation or (gap) between the total travel cost and the total waiting time (TWT). The latter is the sum of all waiting times related to all the vehicles' tours. A negative deviation means a reduction in the values. The computational times are not compared here because those of the ELS [4] were not reported.

Tables 4, 5, and 6 are, respectively, related to the benchmark problems from the $C1$, $C2$, and $C3$ groups defined previously. In these tables, we successively present the instances' names, the number of vehicles, and the number of customers, and, for each evolutionary method, we separately present the resulting total travel travel cost and total waiting time. The gaps between the results obtained with the ETS and the ELS techniques are also reported. The gap is set as $(value(ETS) - value(ELS))/value(ELS) * 100$ for computing the level of improvement of the ETS, as compared with the ELS. Improved values are marked with bolt font.

From the observation of Tables 4–6, the results obtained with the ETS algorithm compete well with the ones of the ELS algorithm. First, the results show the capability of the ETS algorithm to address real-life and expected on-demand transportation problems with considerable sizes. This is the case of the instances *d*64 and *d*35 in Table 5, and *d*18 and *d*22 in Table 6.

In Table 4, there is only one instance, *d*05, that is not improved by the ETS, as compared with the ELS. However, the gap obtained between the two algorithms in this instance is still low (0,79) in TTC and (5,19) in TWT.

In Figure 13, the diagram shows a comparison of the two evolutionary methods considered. This comparison is based on the number of instances where the TTC or the TWT is improved or not by the ETS, as compared with the ELS.

Table 4. Results of the ETS vs. ELS on small-sized problems.

Instances	*m*	*n*	**ELS**		**ETS**		Gap_{TTC}**(%)**	Gap_{TWT}**(%)**
			TTC	**TWT**	**TTC**	**TWT**		
d75	2	10	150.91	398.54	**143.22**	**57.44**	**−5.09**	**−85.59**
d92	2	17	347.01	250.08	**321.7**	**23.17**	**−7.3**	**−90.73**
d93	3	20	418.76	177.34	**410.91**	**8.24**	**−1.87**	**−95.35**
d94	2	23	352.25	153.79	**325.98**	**102.21**	**−7.46**	**−33.54**
d55	5	28	1516.7	202.63	**1506.29**	**80.56**	**−0.69**	**−60.24**
d52	4	29	1607.74	70.81	**1580.09**	90.14	**−1.72**	27.3
d10	4	34	1341.02	377.87	**1333.58**	**241.74**	**−0.56**	**−36.03**
d39	6	38	2030.44	524.72	**1964.28**	**423.87**	**−3.26**	**−19.22**
d70	6	39	2006.05	144.29	**2003.69**	**100.54**	**−0.12**	**−30.32**
d82	6	39	1842.84	95.27	**1839.52**	112.88	**−0.18**	18.48
d08	7	42	1857.35	482.47	**1762.50**	**275.14**	**−5.11**	**−42.97**
d36	6	42	2139.52	390.74	**2136.22**	**323.24**	**−0.15**	**−17.27**
d43	6	43	2002.15	189.57	**1842.04**	210,12	**−8**	10.84
d01	7	46	2396.55	212.51	**2157.81**	**102.99**	**−9.96**	**−51.53**
d11	7	47	2538.18	141.18	**2252.7**	**25.36**	**−11.25**	**−82.04**
d90	6	51	1133.45	355.15	**1126.29**	**213.44**	**−0.64**	**−39.9**
d17	8	52	2861.93	234.74	**2846.24**	**53.17**	**−0.55**	**−77.35**
d84	8	52	2368.34	312.03	**2366.65**	353.46	**−0.07**	13.28
d81	7	53	2456.13	333.38	**2258.38**	**215.46**	**−8.05**	**−35.37**
d96	11	53	3593.6	279.91	**3263.4**	**204.12**	**−9.19**	**−27.08**
d07	8	54	3082.8	167.8	**3056.51**	**93.27**	**−0.85**	**−44.42**
d87	8	54	2714.31	339.88	**2709.77**	**120.65**	**−0.17**	**−64.5**
d47	7	55	2636.07	301.51	**2491.12**	**96.25**	**−5.5**	**−68.08**
d48	8	55	3083.19	418.27	**3037.58**	578.63	**−1.48**	38.34
d61	8	55	2855.09	342.11	**2743.69**	**114.04**	**−3.9**	**−66.67**
d12	9	56	3614.27	341.82	**3399.68**	**214.05**	**−5.94**	**−37.38**
d20	9	56	3567.32	394.84	**3537.15**	**295.65**	**−0.85**	**−25.12**
d30	8	56	2678.58	286.48	**2547.48**	**227.74**	**−4.89**	**−20.5**
d53	7	57	2484.6	104.83	**2391.08**	**12.03**	**−3.76**	**−88.52**
d05	9	58	3393.09	400.98	3419.89	421.78	0.79	5.19
d13	9	59	3183.19	441.76	**3095.3**	**400.33**	**−2.76**	**−9.38**
d06	10	60	3412.34	523.99	**3403.59**	**366.15**	**−0.26**	**−30.12**

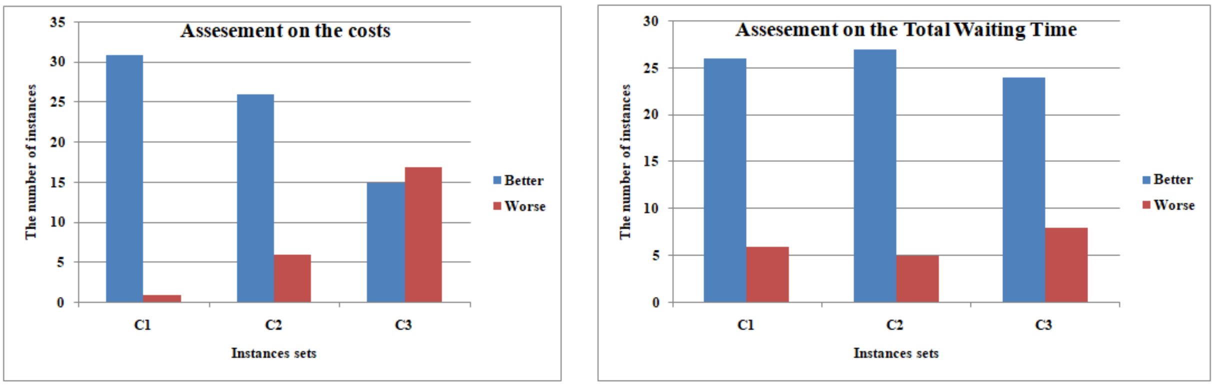

Figure 13. Number of solutions obtained with the ETS that are better or worse than the solutions of the ELS.

Table 5. Results of the ETS vs. ELS on medium-sized problems.

Instances	m	n	**ELS**		**ETS**		Gap_{TTC}(%)	Gap_{TWT}(%)
			TTC	**TWT**	**TTC**	**TWT**		
d03	10	62	3185.61	599.6	3384.89	**428.33**	6.26	−**28.56**
d68	10	62	2606.51	308.12	**2558.42**	**112.25**	−**1.84**	−**63.57**
d74	10	62	3335.58	646.47	**3041.86**	**247.33**	−**8.81**	−**61.74**
d83	9	62	3275.42	241.64	**3168.87**	**111.63**	−**3.25**	−**53.8**
d21	9	63	3218.86	626.41	**3113.14**	**455.12**	−**3.28**	−**27.34**
d26	9	63	2880.21	198.24	2974.04	260.96	3.26	31.64
d88	10	63	3476.11	457.77	**3384.82**	**348.62**	−**2.63**	−**23.84**
d16	10	64	2709.25	435.13	**2684.38**	**241.51**	−**0.92**	−**44.5**
d51	10	64	3035.71	461.75	**2960.35**	**436.54**	−**2.48**	−**5.46**
d31	9	65	3112.39	255.51	**3068.16**	487.88	−**1.42**	90.94
d40	10	66	3815.96	253.79	**3783.61**	**243.41**	−**0.85**	−**4.09**
d41	16	67	3198.74	597.91	**3161.7**	**355.31**	−**1.16**	−**40.57**
d89	10	67	3583.62	390.16	**3204.92**	**192.22**	−**10.57**	−**50.73**
d34	10	68	3066.26	368.69	3260.88	**134.31**	6.35	−**63.57**
d60	9	68	2816.27	729.96	**2759.79**	**611.98**	−**2.01**	−**16.16**
d73	10	68	4278.51	194.21	**4272.07**	**124.56**	−**0.15**	−**35.86**
d28	10	70	3137.18	298.29	**2964.66**	**166.14**	−**5.5**	−**44.3**
d25	10	71	3693.55	436.17	**3552.75**	**377.58**	−**3.81**	−**13.43**
d79	11	73	3206.91	421.22	**3196.82**	456.1	−**0.31**	8.28
d85	12	73	3785.93	847.95	**3624.68**	**722.52**	−**4.26**	−**14.79**
d66	11	75	3350.46	1160.49	3689.34	**912.06**	10.11	−**21.41**
d56	13	76	4287.3	300.36	**4261.59**	**207.47**	−**0.6**	−**30.93**
d69	10	76	2691.37	928.69	2948.11	**633.25**	9.54	−**31.81**
d76	11	76	3467.29	813.88	**3375.31**	**693.94**	−**2.65**	−**14.74**
d86	11	76	3624.95	414.58	**3386.78**	**156.07**	−**6.57**	−**62.35**
d37	10	80	4164.12	208.95	**4083.44**	333.55	−**1.94**	59.63
d64	12	80	4971.05	248.99	**4862.47**	**152.82**	−**2.18**	−**38.62**
d24	11	81	4643.65	178.15	4960	**94.63**	6.81	−**46.88**
d57	12	81	3758.09	500.04	**3718.1**	**422.66**	−**1.06**	−**15.47**
d29	13	82	4705.89	687.13	**4620.41**	**489.67**	−**1.82**	−**28.74**
d09	12	83	3827.63	179.98	**3588.41**	185.13	−**6.25**	2.86
d35	13	84	4142.25	589.31	**4009.54**	**451.39**	−**3.2**	−**23.4**

Table 6. Results of the ETS vs. ELS on large-sized problems.

Instances	m	n	**ELS**		**ETS**		Gap_{TTC}(%)	Gap_{TWT}(%)
			TTC	**TWT**	**TTC**	**TWT**		
d45	11	85	4057.07	563.75	4379.7	**388.11**	7.95	−**31.16**
d80	11	85	4057.1	774.38	4502.42	**537.42**	10.98	−**30.6**
d44	11	86	4240.85	420.22	**4161.15**	**254.57**	−**1.88**	−**39.42**
d54	15	86	3826.1	476.69	4286.63	**248.27**	12.04	−**47.92**
d67	11	86	3615.08	450.2	**3600.21**	**331.86**	−**0.41**	−**26.29**
d63	12	87	3719.97	615.15	**3715.39**	702.39	−**0.12**	14.18
d14	12	88	3710.25	408.21	3969.26	**145.03**	6.98	−**64.47**
d42	13	89	4351.21	354.2	4781.87	**270.65**	9.9	−**23.59**
d02	13	90	4025.16	1090.6	4259.12	**852.14**	5.81	−**21.87**
d04	16	91	5972.08	488.93	**5730.6**	**255.36**	−**4.04**	−**47.77**
d95	10	92	2453.72	375.18	2775.73	**207.96**	13.12	−**44.57**
d50	12	93	4650.02	425.61	5091.04	502.61	9.48	18.09
d71	14	93	4820.43	732.09	**4783.1**	934.4	−**0.77**	27.63
d72	13	93	3978.86	660.08	**3887.55**	**596.14**	−**2.29**	−**9.69**
d15	14	94	5230.89	871.41	5921.25	947.05	13.2	8.68
d33	15	94	4880.13	358.64	**4733.35**	**322.80**	−**3.01**	−**9.99**
d77	13	95	4099.41	621.98	4376.93	935.96	6.77	50.48
d78	13	95	2924.31	554.96	3298.66	**312.44**	12.8	−**43.7**
d59	13	96	5383.8	598.78	**5136.7**	**240.16**	−**4.59**	−**59.89**
d91	13	98	2846.48	866.48	3061.08	**644.37**	7.54	−**25.63**
d23	14	101	4816.45	1045.64	5104.06	**976.05**	5.97	−**6.66**
d38	16	102	5439.44	528.48	**5387.12**	611.68	−**0.96**	15.74
d27	14	110	4664.12	540.4	**4270.23**	**420.97**	−**8.45**	−**22.1**
d58	16	110	5127.06	547.88	5470.56	**354.1**	6.7	−**35.37**
d65	15	111	4431.84	818.49	4864.45	**620.9**	9.76	−**24.14**
d19	15	112	5561.84	800.74	**5394.81**	**593.69**	−**3**	−**25.86**
d62	17	112	4458.1	1021.37	5023.64	**855.34**	12.69	−**16.26**
d22	18	119	6153.15	678.49	**6082.28**	**491.02**	−**1.15**	−**27.63**
d32	17	122	347.47	5824.23	**5794.77**	433.59	−**0.51**	24.78
d49	16	123	5450.74	1273.54	**5449.86**	**1012.63**	−**0.02**	−**20.49**
d46	16	125	5637.37	710.22	5941.78	836.78	5.40	17.82
d18	20	128	6341.12	859.64	**6219.75**	**561.9**	−**1.91**	−**34.64**

Figure 13 demonstrates the significant improvement in total travel cost (TTC) achieved by the ETS compared with the ELS in the C1 instances, with 31 out of 32 instances showing improvement. For the *C*2 instances, there is improvement in 26 out of 32 cases. However, in the last group (C3 instances), the improvement rate is 46.87%, affecting 15 instances. In total, the ETS improves 72 out of 96 on-demand transportation problems tested. This improvement results from the combination of the evolutionary method with TS and the utilization of customized time windows in request scheduling.

In terms of total waiting time, Figure 13 illustrates a substantial enhancement when comparing the ETS with the ELS. The ETS outperforms ELS in 77 out of 96 cases, showcasing the significant contribution of customized time windows in ETS resolution.

Overall, the service quality is well addressed across all instances of TOD. TWT is improved for 26, 27, and 24 instances out of 32 in the $C1$, $C2$, and $C3$ classes, respectively.

6. Statistical Analysis of the Results Provided for the Problem

In order to assess well the performance of our new hybridized evolutionary method ETS, a deeper analysis of the results is required. We assess the behavior of the combined methods, which are the TS and the evolutionary algorithm (EA), before their hybridization. In addition, the analysis concerns the hybridized ones, which are the ETS and the ELS. Therefore, statistical significance between the results of the compared algorithms is performed based on the TTC obtained, being the values to not deteriorate when improving the quality of service in the problem.

We follow the work of [52], which proposed guidelines to conduct a statistical analysis of experimental studies. To this end, a set of steps is prescribed for statistically comparing the algorithms. However, tests of normality are needed to select the appropriate statistical tests: parametric or non-parametric associated with normal or non-normal data distribution. Many tests are provided in the literature to ensure the distribution of the data used in the experimentation, such as the Shapiro–Wilk test [53] or the Kolmogorov–Smirnov [54] test.

According to the Shapiro–Wilk test, the results of the algorithms do not follow a normal distribution, as is the case for the majority of the metaheuristics and bio-inspired algorithms [52]. Therefore, non-parametric tests are selected to derive statistical tests.

6.1. Tests of Differences

In order to detect differences between algorithms, the Friedman's test is used. It consists in accepting or rejecting the null hypothesis, H_0. This hypothesis considers that the algorithms have a similar performance. Given k is the number of algorithms equal to 4, b the number of instances equal to 96, and T_j^2 the squared sum of the ranks for an algorithm, the Friedman equation is given by (9).

$$F_T = \left[\frac{12}{b(k)(k+1)} \sum_{j=1}^{j=k} T_j^2\right] - 3b(k+1) \tag{9}$$

The average ranking of the compared algorithms is provided based on the average of solutions within 10 runs. Next, the sum and the average of the ranking of each algorithm are indicated in Table 7.

Table 7. Indications of the Friedman test.

	TS	*EA*	*ELS*	*ETS*
Sum of ranks	348	257	226	128
Average rank	3.63	2.67	2.35	1.33
Sum of squared ranks	121,104	66,049	51,076	16,384

Using the values produced in Table 7, the Friedman test (9) produces a value of 151,33. In addition, the $CHI - square$ critical value (CV) is 7.81 according to the $Chi - square$ χ^2 distribution. Thus, if we have $F_T > CV$, the rejection of the null hypothesis is YES, and NO otherwise. In our case, 151.33 > 7.81. Consequently, we deduce that there are differences between the compared algorithms, and the null hypothesis is rejected with a confidence level of 95%.

Moreover, the algorithm is considered to be the best one if it deals with the best average ranking value, as is the case with ETS presenting a less average rank equal to 1.33. However, this indication is not enough to conclude which algorithm has the best performance. Therefore, we proceed by pairwise comparisons to effectively observe the significance of the algorithms' performance.

6.2. Pairwise Comparisons

In this part, a statistical significance between the compared algorithms is addressed regarding both the obtained solutions values and the distribution of the obtained solutions in the search space. Thus, to analyze the performance of the algorithms in terms of solution values, we compute the absolute difference of the sums of ranks $|S_i - S_j|$ between two algorithms i and j. The difference is significant when it exceeds the critical value equal to 7.81. The winner algorithm is that having the smallest sum of ranks between the two compared ones. In Table 8, the absolute difference between the sums of ranks (previously computed in Table 7) is indicated.

Table 8. Pairwise comparisons of the ranked solutions value.

Groups	$|S_i - S_j|$	**Significance**	**Winner**	**Loser**
TS vs. EA	91	YES	EA	TS
TS vs. ELS	122	YES	ELS	TS
TS vs. ETS	220	YES	ETS	TS
EA vs. ELS	31	YES	ELS	EA
EA vs. ETS	129	YES	ETS	EA
ETS vs. ELS	98	YES	ETS	ELS

Based on the difference of ranking sums in terms of solutions values, all the differences are significant since $|S_i - S_j| > CV$ for each pairwise algorithm. Moreover, these outputs highlight the dominance of the ETS, which occupies the largest number of winner occurrences in Table 8.

Whenever the significance in terms of solution values is analyzed, the one based on the solution value distributions in the search space is mandatory. Therefore, the p-value is calculated for each group of comparisons using the Student t-test distribution in Table 9. Next, the Bonferroni correction method is applied for α as α/k. The obtained adjusted p-value is equal to 0.008. A significance is mentioned as Yes when the obtained p-value of a group is less than the adjusted one. Otherwise, it is mentioned as No.

Table 9. Pairwise comparisons of the solution value distributions.

Groups	p**-Value**	**Significance**
TS vs. ELS	0.008	Yes
TS vs. EA	0.198	No
TS vs. ETS	0.007	Yes
EA vs. ELS	0.197	No
EA vs. ETS	0.172	No
ETS vs. ELS	0.921	No

The cases that indicate the significance of the differences in terms of distribution are TS vs. ELS and TS vs. ETS. Nevertheless, all the other groups indicate no significant result of the solution value distributions in the search space.

6.3. Interpretation of the Statistical Results

According to the results provided in Tables 8 and 9, we point out that the results' significance is not the same in terms of solution values and their distribution in the search space. Regarding the guidelines of [52], the following interpretations are highlighted.

The TS algorithm indicates no significant results in terms of solution values, as compared with EA, ELS, and ETS (see Table 8). But, the distribution of the values of the solutions provided by this losing method has a statistical significance, as compared with the hybrid evolutionary algorithms, which are the ELS and ETS, see Table 9. Obviously, these advanced methods involve a larger exploration mechanism than the TS as well as the EA, since they present no significant distribution of solution values (Table 9).

As we see in Table 8, the algorithms EA, ELS, and ETS have an increased significance in terms of solution values but not their distribution (Table 9). However, according to the indications of Table 8, the EA is losing against the ELS and ETS regarding the solution values' significance. Therefore, the EA has a lower performance than the hybridized ones.

Moreover, the two hybridized algorithms ELS and ETS have the same exploration power but with a different exploitation efficiency. Given that the ETS does not provide any losing case in terms of ranking solution values, see Table 8, we can deduce that it dominates the ELS. Thus, we emphasize the effectiveness of the TS when it is used as an exploitation mechanism in the hybrid evolutionary algorithm ETS.

7. Discussion

Our findings highlight the significance of enhancing the service quality in on-demand transportation systems. By reducing waiting times, improving schedule reliability, and implementing customer-centric features, we not only enhance the user satisfaction but also increase the system efficiency. This aligns with the goals of smart cities, where technological innovations are required to create more responsive and adaptive transportation networks. The implications of our findings extend beyond theoretical considerations to practical applications within smart city transportation planning and management. By optimizing the service quality, we can achieve more efficient resource utilization and cost savings.

To tackle DARPs, which are more applicable to real-world scenarios, it is relevant to consider dynamic elements such as variable capacities and real-time transit data.

However, we have started by assuming fixed capacities, which allows us to focus on optimizing the routing and scheduling aspects without the additional complexity of variable capacities. This is a common practice in initial methodological studies, providing a clear framework to demonstrate the core algorithmic innovations. Future works can extend the method to accommodate variable capacities, making it more applicable to real-world scenarios. Similarly, assuming constant transit times simplifies the model, making it more tractable for computational experiments. In real-world applications, transit times can vary due to traffic conditions, roadworks, and other factors. When considering dynamic factors, such as variable transit times and vehicle capacities, the computational complexity of the hybrid evolutionary tabu search method for transport on demand scenarios, like the dial-a-ride problem, increases significantly. Therefore, there are effective strategies to address complexity as efficient dynamic data integration techniques, such as stream processing, to manage real-time transit data effectively. Moreover, a dynamic parameter adjustment would be beneficial to address the complexity. The method should adapt parameters like mutation rates, crossover rates, and neighborhood sizes based on current solution quality and real-time system changes. It would be interesting to use machine learning techniques to predict optimal parameter settings based on historical data and current system states. In addition, we should also integrate heuristics that provide quick, approximate solutions to guide the evolutionary algorithms and tabu search, reducing the search space and improving convergence rates.

Implementing these strategies can significantly improve the scalability of the hybrid evolutionary tabu search method for medium and large networks in transport on-demand scenarios. These enhancements ensure that the method remains robust and efficient, making it suitable for real-world applications in smart cities. Future work will continue to refine these strategies, further improving scalability and practical applicability in diverse urban transport environments.

8. Conclusions

This paper introduces an efficient hybrid algorithm called the evolutionary tabu search that addresses the challenges of solving a specific variant of the dial-a-ride problem with customer-oriented constraints. By combining the evolutionary computation framework with the tabu search, the ETS algorithm offers a powerful approach to optimizing transportation solutions. The primary objective of the ETS method is to provide tailored optimization techniques that support customized designs for the DARP. To this end, the algorithm incorporates specific mutation operators that are designed to address the unique requirements of the problem. These operators are dynamically exchanged during the search process, allowing for the exploration of diverse solution spaces and leading to an improved service quality. To validate the effectiveness of the ETS algorithm, extensive experiments were conducted, comparing its performance with another hybrid evolutionary method from the literature using real-life transportation problems. The results of these experiments demonstrate the superior performance of the ETS algorithm in terms of improvement of the service quality. The introduction of the new tabu search method with its customized neighborhood strategy and tailored evolutionary operators represents a significant contribution to the DARP field. It presents a promising approach for solving the problem at hand by effectively addressing both operational and service quality objectives, thereby providing a valuable tool for decision-making in relevant domains. The findings of the study highlight the strengths and advantages of the ETS method in addressing the challenges and requirements of transport on-demand problems. The method's ability to deliver competitive results, coupled with the observed improvements in service quality, signifies its potential as an effective solution for optimizing on-demand transportation systems. In future research, it would be valuable to explore the integration of additional criteria, such as waiting time, riding time, and tour cost, within a multi-objective framework. This would further enhance the algorithm's capability to deliver good solutions that consider multiple aspects of the problem.

Furthermore, future works will extend the model to incorporate dynamic elements such as variable capacities and real-time transit data, making it more relevant for real-world scenarios. Enhancing the method to accommodate real-time data on traffic conditions and vehicle availability should also improve the accuracy and responsiveness of the system. Thus, extending the method to handle variable vehicle capacities and other dynamic constraints is encouraged, making the method more applicable to diverse urban transport scenarios in smart cities.

In conclusion, the presented hybrid optimization method provides a significant advancement in solving DARPs for smart cities. It not only enhances service quality and operational efficiency but also sets the stage for further research and development to adapt the method to the dynamic and complex nature of real-world transportation systems.

Author Contributions: Conceptualization, S.N. and H.B.; methodology, S.N.; software, S.N.; validation, S.N., H.B. and W.A.M.; formal analysis, S.N.; investigation, S.N.; resources, S.N.; data curation, S.N.; writing—original draft preparation, H.B.; writing—review and editing, S.N.; visualization, W.A.M.; supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data required to reproduce the above findings are available to download from Scopus Document Searcher using the keywords specified at the methodology.

Acknowledgments: We would like to express our gratitude to the administrative staff of the Business Higher School of Tunis for their assistance in coordinating logistics and facilitating communication. I express my sincere gratitude to the technical support team of the LARODEC laboratory for their expertise in maintaining equipment and providing troubleshooting assistance. I also thank LARODEC laboratory for generously providing access to their facilities and resources, which significantly contributed to the success of our experiments.

Conflicts of Interest: The authors declare that they have no conflicts 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 Methodol.* **2003**, *37*, 579–594. [CrossRef]
2. 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]

3. Molenbruch, Y.; Braekers, K.; Caris, A. Typology and literature review for dial-a-ride problems. *Ann. Oper. Res.* **2017**, *259*, 295–325. [CrossRef]
4. 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]
5. Parragh, S.N. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transp. Res. Part Emerg. Technol.* **2011**, *19*, 912–930. [CrossRef] [PubMed]
6. Braekers, K.; Kovacs, A.A. A multi-period dial-a-ride problem with driver consistency. *Transp. Res. Part Methodol.* **2016**, *94*, 355–377. [CrossRef]
7. Masmoudi, M.A.; Braekers, K.; Masmoudi, M.; Dammak, A. A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. *Comput. Oper. Res.* **2017**, *81*, 1–13. [CrossRef]
8. Agrawal, V.; Lightner, C.; Lightner-Laws, C.; Wagner, N. A bi-criteria evolutionary algorithm for a constrained multi-depot vehicle routing problem. *Soft Comput.* **2017**, *21*, 5159–5178. [CrossRef]
9. Bahadori-Chinibelagh, S.; Fathollahi-Fard, A.M.; Hajiaghaei-Keshteli, M. Two constructive algorithms to address a multi-depot home healthcare routing problem. *IETE J. Res.* **2019**, *68*, 1108–1114. [CrossRef]
10. Masson, R.; Lehuédé, F.; Péton, O. The dial-a-ride problem with transfers. *Comput. Oper. Res.* **2014**, *41*, 12–23. [CrossRef]
11. Molenbruch, Y.; Braekers, K.; Caris, A. Benefits of horizontal cooperation in dial-a-ride services. *Transp. Res. Part Logist. Transp. Rev.* **2017**, *107*, 97–119. [CrossRef]
12. Nasri, S.; Bouziri, H.; Aggoune-Mtalaa, W. An evolutionary descent algorithm for customer-oriented mobility-on-demand problems. *Sustainability* **2022**, *14*, 3020. [CrossRef]
13. 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]
14. Paquette, J.; Cordeau, J.F.; Laporte, G.; Pascoal, M.M. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transp. Res. Part Methodol.* **2013**, *52*, 1–16. [CrossRef]
15. 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]
16. Carotenuto, P.; Paradisi, L.; Storchi, G. A flexible transport service for passengers. *Transp. Res. Procedia* **2014**, *3*, 442–451. [CrossRef]
17. Viana, R.J.; Santos, A.G.; Martins, F.V.; Wanner, E.F. Optimization of a demand responsive transport service using multi-objective evolutionary algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, 13–17 July 2019; pp. 2064–2067.
18. Belhaiza, S. A hybrid adaptive large neighborhood heuristic for a real-life dial-a-ride problem. *Algorithms* **2019**, *12*, 39. [CrossRef]
19. Chu, J.C.Y.; Chen, A.Y.; Shih, H.H. Stochastic programming model for integrating bus network design and dial-a-ride scheduling. *Transp. Lett.* **2022**, *14*, 245–257. [CrossRef]
20. Nguyen, J.; Powers, S.T.; Urquhart, N.; Farrenkopf, T.; Guckert, M. Modelling the impact of individual preferences on traffic policies. *Comput. Sci.* **2022**, *3*, 365. [CrossRef]
21. 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]
22. Zhang, Z.; Liu, M.; Lim, A. A memetic algorithm for the patient transportation problem. *Omega* **2015**, *54*, 60–71. [CrossRef]
23. Braekers, K.; Caris, A.; Janssens, G.K. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transp. Res. Part Methodol.* **2014**, *67*, 166–186. [CrossRef]
24. 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.
25. Wong, K.I.; Han, A.; Yuen, C. On dynamic demand responsive transport services with degree of dynamism. *Transp. Transp. Sci.* **2014**, *10*, 55–73. [CrossRef]
26. 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]
27. 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 Emerg. Technol.* **2015**, *55*, 156–165. [CrossRef]
28. 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]
29. 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.
30. 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]
31. Paquette, J.; Cordeau, J.F.; Laporte, G. Quality of service in dial-a-ride operations. *Comput. Ind. Eng.* **2009**, *56*, 1721–1734. [CrossRef]
32. 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]

33. Parragh, S.N.; Pinho de Sousa, J.; Almada-Lobo, B. The dial-a-ride problem with split requests and profits. *Transp. Sci.* **2015**, *49*, 311–334. [CrossRef]
34. Glover, F.; Kelly, J.P.; Laguna, M. Genetic algorithms and tabu search: hybrids for optimization. *Comput. Oper. Res.* **1995**, *22*, 111–134. [CrossRef]
35. Nasri, S.; Bouziri, H. Improving total transit time in dial-a-ride problem with customers-dependent criteria. In Proceedings of the IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017 ; pp. 1141–1148.
36. Cheikh-Graiet, S.B.; Dotoli, M.; Hammadi, S. A Tabu Search based metaheuristic for dynamic carpooling optimization. *Comput. Ind. Eng.* **2020**, *140*, 106217. [CrossRef]
37. Gmira, M.; Gendreau, M.; Lodi, A.; Potvin, J.Y. Tabu Search for the Time-Dependent Vehicle Routing Problem with Time Windows on a Road Network. *Eur. J. Oper. Res.* **2020** , *288*, 129–140. [CrossRef]
38. Aziz, H.A.; Garikapati, V.; Rodriguez, T.K.; Zhu, L.; Sun, B.; Young, S.E.; Chen, Y. An optimization-based planning tool for on-demand mobility service operations. *Int. J. Sustain. Transp.* **2022**, *16*, 45–56. [CrossRef]
39. Chen, Z.G.; Zhan, Z.H.; Kwong, S.; Zhang, J. Evolutionary computation for intelligent transportation in smart cities: A survey. *IEEE Comput. Intell. Mag.* **2022**, *17*, 83–102. [CrossRef]
40. Cubillos, C.; Urra, E.; Rodríguez, N. Application of genetic algorithms for the DARPTW problem. *Int. J. Comput. Commun. Control.* **2009**, *4*, 127–136. [CrossRef]
41. 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]
42. 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]
43. 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]
44. Nasri, S.; Bouziri, H. Towards a fair mobility: An evolutionary algorithm for customers-dependent transport on demand problems. In Proceedings of the 2019 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Tunis, Tunisia, 20–22 December 2019; pp. 34–40.
45. Molenbruch, Y.; Braekers, K.; Hirsch, P.; Oberscheider, M. Analyzing the benefits of an integrated mobility system using a matheuristic routing algorithm. *Eur. J. Oper. Res.* **2021**, *290*, 81–98. [CrossRef]
46. Chevrier, R.; Liefooghe, A.; Jourdan, L.; Dhaenens, C. Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport. *Appl. Soft Comput.* **2012**, *12*, 1247–1258. [CrossRef]
47. Ghoseiri, K.; Ghannadpour, S.F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Appl. Soft Comput.* **2010**, *10*, 1096–1107. [CrossRef]
48. Li, J.; Tomita, K.; Kamimura, A. A Novel Genetic Algorithm for a Multi-Vehicle Dial-a-Ride Problem. In Proceedings of the 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 3–5 July 2022; pp. 682–689.
49. Perera, T.; Prakash, A.; Gamage, C.N.; Srikanthan, T. Hybrid genetic algorithm for an on-demand first mile transit system using electric vehicles. In Proceedings of the International Conference on Computational Science, Las Vegas, NV, USA, 12–14 December 2018 ; Springer: Berlin/Heidelberg, Germany, 2018; pp. 98–113.
50. Godart, A.; Manier, H.; Bloch, C.; Manier, M.A. Hybrid metaheuristic for the Pickup and Delivery Problem designed for passengers and goods transportation. *IFAC-PapersOnLine* **2019**, *52*, 2584–2589. [CrossRef]
51. Chassaing. Instances of Chassaing. 2020. Available online: http://fc.isima.fr/~lacomme/Maxime/ (accessed on 19 July 2020).
52. LaTorre, A.; Molina, D.; Osaba, E.; Poyatos, J.; Del Ser, J.; Herrera, F. A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm Evol. Comput.* **2021**, *67*, 100973. [CrossRef]
53. Shaphiro, S.; Wilk, M. An analysis of variance test for normality. *Biometrika* **1965**, *52*, 591–611. [CrossRef]
54. Razali, N.M.; Wah, Y.B. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *J. Stat. Model. Anal.* **2011**, *2*, 21–33.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.