

Article

Approaching the Pareto Front in a Biobjective Bus Route Design Problem Dealing with Routing Cost and Individuals' Walking Distance by Using a Novel Evolutionary Algorithm

Herminia I. Calvete ^{1,*}, Carmen Galé ² and José A. Iranzo ³¹ Statistical Methods Department, IUMA, University of Zaragoza, Pedro Cerbuna 12, 50009 Zaragoza, Spain² Statistical Methods Department, IUMA, University of Zaragoza, María de Luna 3, 50018 Zaragoza, Spain; cgale@unizar.es³ Statistical Methods Department, IUMA, University of Zaragoza, Violante de Hungría 23, 50009 Zaragoza, Spain; joseani@unizar.es

* Correspondence: herminia@unizar.es

Abstract: This paper addresses a biobjective bus routing problem that pays attention to both the routing cost and the total distance walked by the individuals to reach their assigned pickup point. These two objectives are conflicting. Generally, the less the individuals walk, the more the number of visited pickup points and so the more the routing cost. In addition, the problem deals with finding the set of pickup points visited among the set of potential locations, identifying the set of individuals assigned to each visited pickup point, and designing the bus routes. Taking into account the highly combinatorial nature of the problem, an evolutionary algorithm is proposed to approach the associated Pareto front. Its main novelties are twofold. The first is the way in which the chromosomes are encoded since they only provide information about the number of routes and the visited pickup points. The second novelty lies in the procedure to construct a feasible solution from the chromosome, which involves a heuristic and several local search procedures to improve both objective functions. Computational experiments are carried out to check the performance of the algorithm in terms of the quality of the Pareto front yielded.

Keywords: bus routing problem; biobjective; evolutionary; local search; school; workplace

MSC: 90B06; 90C29; 90B50



Citation: Calvete, H.I.; Galé, C.; Iranzo, J.A. Approaching the Pareto Front in a Biobjective Bus Route Design Problem Dealing with Routing Cost and Individuals' Walking Distance by Using a Novel Evolutionary Algorithm. *Mathematics* **2022**, *10*, 1390. <https://doi.org/10.3390/math10091390>

Academic Editors: Antonio Jiménez-Martín and Shih-Wei Lin

Received: 9 March 2022

Accepted: 19 April 2022

Published: 21 April 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/>).

1. Introduction

Bus routing problems (BRP) consist of designing a set of routes for transporting individuals from pickup points to their destination. This broad description encompasses a very diverse set of problems that arise in a number of fields and give rise to formulations with different objectives and constraints. This paper addresses a variant of the BRP where, simultaneously: (1) the pickup points visited by the routes must be selected from a potential set of locations, (2) the individuals have to be assigned to the visited pickup points and (3) the bus routes must be designed. Problems with these characteristics are frequent in the design of school bus routes, in which the individuals are the students and the destination is the school. They also arise when the employees of a company are offered the possibility of being picked up at different points selected by the company to be transported to the workplace.

Although the BRP has major implications both from an economic point of view, since it involves spending on routes, and from the point of view of the quality of care, comfort or security of individuals, since they must go from their homes to their assigned pickup point and back, usually, only the economic issues are taken into account. This paper aims to fill this gap by studying a biobjective bus routing problem (BBRP) which addresses both issues in the objective functions. On the one hand, the problem will minimize the cost due

to the routes (routing cost). On the other hand, it will minimize the total distance walked by individuals (walking distance) as a measure of the cost of reaching the pickup points. These objectives are connected through the selection of the pickup points from the set of existing locations and the allocation of individuals to them. In general, the more pickup points that are selected, the shorter the distance walked by individuals because they will probably have a pickup point in the neighbourhood closest to their homes, but the routing costs will be higher. Bus capacity also plays a role as it prevents the free allocation of any number of individuals to the pickup points. Due to the intrinsically conflicting nature of the two objective functions, the BBRP formulated in this paper focuses on the construction of the Pareto front [1]. On the other hand, notice that this problem is highly combinatorial since after deciding which pickup points are visited and how the individuals are allocated to them, a biobjective vehicle routing problem still has to be solved. Hence, the aim will be to provide a good approximation of the Pareto front.

Besides the formulation of this new biobjective problem, the other relevant contribution of the paper is the development of a hybrid metaheuristic based on an evolutionary algorithm to approach the Pareto front. Evolutionary algorithms have been widely applied in the literature [2–6] since they have proved to be very effective in approaching the Pareto front. To cope with such a complex problem as the BBRP, the metaheuristic developed in the paper uses the general framework of evolutionary algorithms but has two key distinguishing features. The first one is chromosome encoding, which only partially identifies a feasible solution. Thus, the chromosome encoding provides the number of routes and which pickup points are visited but gives no information on the routes themselves and the allocation of individuals to these pickup points. Hence, based on a chromosome we can get feasible solutions with very different values of both objective functions. The second distinguishing feature is the procedure developed to associate a good feasible solution of the BBRP with the chromosome, which allows us to rule out feasible solutions which are known for sure not to be efficient. For this purpose, an initial feasible solution is computed using a heuristic and then several local search procedures are applied aiming to improve both objective functions. The first procedure focuses on the improvement of the walking distance by solving a transportation problem. The second procedure improves the routing cost by applying 2-opt. The third procedure, which is the most complex, aims to improve both objective functions by changing the selection of the pickup points. Finally, the fourth procedure simplifies the solution by removing unused pickup points or combining routes, if possible. The last procedure can alter the number of routes and visited pickup points. To see the influence of either updating or not the chromosome after this local search procedure, the computational experiment analyzes the performance of the algorithm under both levels of this factor. In the computational experiment, two crossover operators are also analyzed, based on the uniform and the single point crossovers. The results provided by the unary hypervolume indicator and the additive ϵ indicator lead us to conclude that the chromosome should not be updated, whatever the crossover operator selected. The rest of the paper is structured as follows. In Section 2 an outline of the relevant existing literature on the BRP is provided, focusing on papers that deal with the pickup points selection. Section 3 describes the biobjective BRP problem introduced in the paper. In Section 4 the characteristics of the evolutionary algorithm developed for solving it are described. Section 5 provides the results of the computational study carried out to analyze the performance of the proposed algorithm. Finally, Section 6 concludes the paper.

2. Literature Review

The design of school bus routes in the area in which the BRP has likely received the most attention. The paper by Newton and Thomas [7] is one of the pioneering works on the school bus routing problem (SBRP) using a computer to tackle a real-life problem. Since then, multiple papers which deal with both theoretical aspects of the problem and the solution of particular cases have appeared. Most studies assume that the locations of the pickup points are given and the individuals are allocated a priori. Thus, they concentrate on designing the bus routes, leading to variants of the well-known Capacitated Vehicle

Routing Problem (CVRP) [8–10]. Park and Kim [11] review studies on the SBRP published up until 2009. They classify problems based on their characteristics: number of schools, surroundings of service, problem scope, mixed loads, special-education students, fleet mix, objectives and type of constraints. Regarding the solution methods, most papers analyzed consider heuristic approaches due to the complexity of the problem. More recently, Ellegood et al. [12] have extended the review up to 2019, highlighting the fact that the topic is still relevant today. Their classification follows similar guidelines to Park and Kim [11], but shows that other objectives and constraints have been considered in recent papers. Peker and Eliyi [13] provides comprehensive literature for shuttle bus routing in different functional application areas. They highlight that transportation systems in the people domain appear in many areas such as public or intercity transportation, airports, tourism, student, or personnel transport, etc. These systems have a significant role in the quality of life, affecting factors such as time, comfort, stress and money.

It is worth mentioning the existence of papers that deal with more than one objective function. Corberán et al. [14] consider a biobjective SBRP in a sparsely populated rural district, whose objectives are minimizing the total number of buses and minimizing the maximum time that a student spends on the bus. Bus stops are fixed and bus capacity constraints are not taken into account. A set of efficient solutions is computed using scatter search with two constructive heuristics to generate solutions. In Pacheco et al. [15] the two objectives considered are minimizing the longest route and minimizing the total distance travelled. To approach the Pareto front, they develop a solution procedure that uses tabu search combined with Multiobjective Adaptive Memory Programming. In employees transportation, Dasdemir et al. [16] allow the existence of overbooking and model the problem as a biobjective open vehicle routing problem. They minimize the total distance and the total number of passengers who stand and hold onto a strap. For the purpose of generating the Pareto front they apply the ϵ -constraint method and also develop two heuristic approaches.

As mentioned in Section 1, the BBRP problem addressed in this paper, along with route design, also deals with the selection of the pickup points from the set of available locations and the allocation of individuals to them. The number of papers dealing simultaneously with the three sub-problems is less numerous and are restricted to the area of the SBRP. Dulac et al. [17] consider this location-allocation-routing problem and propose a two-stage approach to handle it. First, the bus stops are selected and the students are allocated to them. The second stage constructs the routes. There are constraints on the capacity of the buses, the number of stops and the length of the routes, and the distance walked by the students. Chapleau et al. [18] partition the students into a minimum number of clusters whose cardinality takes into account the route capacity. Then, for each cluster, a route is built after selecting the bus stops. Bowerman et al. [19] also follow this allocation-routing-location strategy in urban areas. The goal of minimizing the number of routes is the essential objective, but other criteria are combined into a single overall objective using the weighting method. The heuristic procedure proposed also partitions the students into clusters for which a single route will be constructed. Riera-Ledesma and Salazar-González [20] formulate the problem as a multiple-vehicle travelling purchaser problem whose objective minimizes the cost of the routes plus the cost of assigning users to bus stops. They solve the problem by using a branch-and-cut algorithm. Riera-Ledesma and Salazar-González [21] extend this problem to include additional constraints related to the quality of service. They present alternative formulations of the problem and a branch-and-cut-and-price algorithm. Schittekat et al. [22] minimize the routing cost and fix an upper bound on the distance walked by students. For solving the problem they develop a GRASP + VND matheuristic. Also for this problem, Kinable et al. [23] propose an exact branch-and-price algorithm and Calvete et al. [24] develop a matheuristic that constructs feasible solutions by successively selecting bus stops, partially assigning students, computing the routes and allocating all the students, and applying local search procedures. Bögl et al. [25] consider the SBRP with transfers in which students are allowed to change buses. Moreover, they aim to determine optimal school start times to give buses the flexibility of servicing multiple schools.

The algorithm proposed to solve the problem is based on a destroy and repair strategy. Calvete et al. [26] study a variant of the SBRP which allows students to select the bus stop among those visited by a bus. They formulate a bilevel optimization model and propose a metaheuristic that involves the solution of four mixed-integer-linear optimization problems to obtain tight upper bounds of the upper level optimal objective function value.

The above description shows that, in general, the distance walked by individuals is usually considered in the set of constraints [22–24], establishing a maximum walking distance. When this is considered in the objective function, it is summed with the routing costs [20,21], which can be controversial since both costs are not comparable unless they are proportional. As mentioned in Section 1, in this paper we propose to consider the routing cost and the walking distance separately, leading to a biobjective bus routing problem with pickup selection and allocation of individuals.

3. Problem Description

The main issues handled in the BBRP addressed in this paper are twofold. First of all, this is a routing problem in which, in addition to designing the routes, the pickup points and individual allocation must be determined. The second key aspect is that this problem simultaneously deals with minimizing the cost of the routes and minimizing the distance of users to their pickup point. Hence, it takes into account not only economic considerations but also pays attention to the safety and comfort of individuals. The BBRP can be described as follows. There is a set of individuals U , a set of potential pickup points W and a destination (workplace, school, etc.). Each individual is allocated to a pickup point, where he/she is picked up by a bus that takes him/her to his/her destination. Let H be the set of identical buses located at a depot. Let Q be the bus capacity. We assume that H is at least as large as the minimum number of buses needed to transport all the individuals ($HQ \geq |U|$). Buses visit only those pickup points to which individuals have been allocated. Moreover, the total number of individuals allocated to the pickup points in a bus route is limited by the capacity of the bus. Each bus route starts at the depot, visits a number of pickup points and finishes visiting in sequence the destination and the depot. Each bus may perform at most one route and each pickup point with allocated individuals is visited by at most one vehicle.

Let $G = (V, E \cup A)$ be a mixed graph, where $V = \{0\} \cup U \cup W$ and node 0 refers to the depot. The set of edges is defined as $E = \{[i, j] : i, j \in V \setminus U, i \neq j\}$. Edges are links that are used in the routes. The set of arcs is defined as $A = \{(k, i) : k \in U, i \in W\}$. Arcs allow allocating individuals to pickup points. It is assumed that there are neither loop edges nor loop arcs.

Each edge $[i, j] \in E$ has associated a nonnegative routing cost c_{ij} which refers to the cost of traveling from node i to node j . The cost c_{0i} , $i \in W$ includes the routing cost from the depot to the pickup point i and can also include the fixed cost of operating the bus. The cost c_{i0} , $i \in W$ includes the routing cost from the pickup point i back to the depot via the destination. It is assumed that the cost matrix satisfies the triangle inequality, i.e., $c_{il} + c_{lj} \geq c_{ij}$ for all $i, j, l \in V \setminus U$. Each arc $(k, i) \in A$ has associated a nonnegative walking distance d_{ki} referring to the distance from the location of individual k to pickup point i . In addition to the routing cost, which is relevant from the economic point of view, the cost due to the allocations of individuals to the pickup points (walking distance) is also significant. As mentioned above, the problem is to determine the set of visited pickup points, the allocation of individuals to them and the routes followed by buses in order to minimize both the total routing cost and the total distance walked by individuals to reach their allocated pickup point.

A feasible solution $X = \{W_v, A_v, R_1, \dots, R_m\}$ consists of a set $W_v \subset W$ of visited pickup points, a set of index pairs $A_v \subset A$, where for each individual $k \in U$, the corresponding pair $(k, i_k) \in A_v$ provides the index $i_k \in W_v$ of the pickup point to which the individual k is allocated, and a set of $m \leq H$ routes R_1, \dots, R_m that allow individuals to be picked up at their assigned pick up points and to be transported to the destination, taking into account the capacity of the bus. The routing cost and the walking distance are:

$$\text{Routing cost: } Z_1 = \sum_{h=1}^m \sum_{[i,j] \in R_h} c_{ij}, \tag{1}$$

$$\text{Walking distance: } Z_2 = \sum_{(k,i_k) \in A_v} d_{ki_k}.$$

Let \mathcal{X} denote the set of feasible solutions. The BBRP can be formulated through the following Biobjective Binary Integer Linear Programming model:

$$\begin{aligned} \min \quad & Z(X) = [Z_1(X), Z_2(X)] \\ \text{subject to} \quad & X \in \mathcal{X}. \end{aligned} \tag{2}$$

It is worth noting that, although both objective functions depend on different structures (routes versus allocations), they are related since both are interdependent. Indeed, the allocation depends on the pickup points visited and on the route constructed through the bus capacity constraint. Similarly, the routes depend on the pickup points visited and on the allocation of individuals to pickup points. Figure 1 shows that, from the same set of potential pick up points, feasible solutions with different objective function values can be obtained depending on how the routes are constructed and how the allocation is made. Figure 1a displays the node-set V . The red circle represents the depot. Blue squares are the potential pickup points that can be visited and black circles are the individuals. To make the figure more explanatory, an additional red square has been added which represents the destination. The bus capacity is equal to eight. The feasible solution in Figure 1b has less routing cost but more walking distance than the feasible solution in Figure 1c.

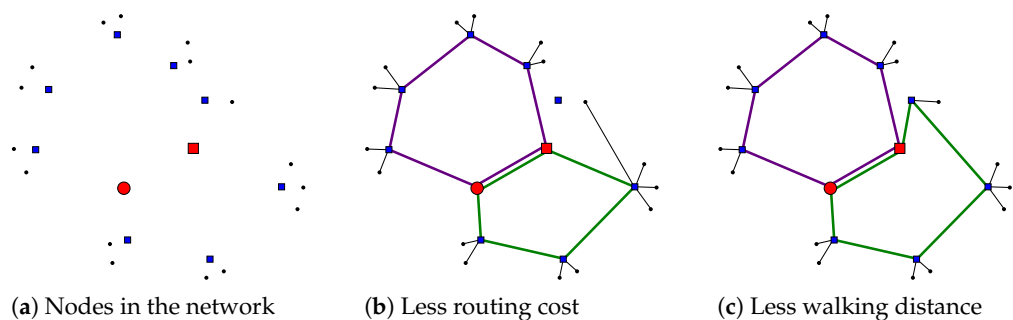


Figure 1. Two feasible solutions with the same potential pickup points, the same number of routes and different costs. The red circle represents the depot. The red square represents the destination. Blue squares are the potential pickup points and black circles are the individuals. The bus capacity is equal to eight.

For a nontrivial biobjective problem, i.e., a problem in which both objective functions are conflicting, a single solution that simultaneously minimizes each objective cannot be found. Improving the value of one of the objective functions usually has a negative effect on the other [1]. From the different approaches to multi-objective optimization problems, we focus here on the construction of the Pareto front. A feasible solution is said to be efficient if it is impossible to find another feasible solution with a better performance of one objective without degrading the performance of the other objective. Mathematically, a feasible solution X is efficient if and only if there is no other feasible solution \tilde{X} so that

$$\begin{aligned} Z_s(\tilde{X}) &\leq Z_s(X), \quad s = 1, 2, \\ Z_t(\tilde{X}) &< Z_t(X), \quad \text{for at least one index } t \in \{1, 2\}. \end{aligned}$$

A feasible solution X is weakly efficient if and only if there is no other feasible solution \tilde{X} so that $Z_s(\tilde{X}) < Z_s(X)$, $s = 1, 2$. The image in the objective space of an efficient (weakly

efficient) feasible solution is a nondominated (weakly nondominated) point. That is to say, if $X \in \mathcal{X}$ is an efficient (weakly efficient) solution, $Z(X) = [Z_1(X), Z_2(X)]$ is a nondominated (weakly nondominated) point. The set of all nondominated points is the Pareto front. Figure 2 displays the image in the objective space of a set of feasible solutions. The Pareto front is shown in red.

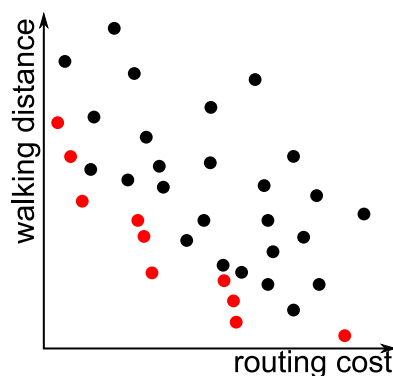


Figure 2. Image in the objective space of a set of feasible solutions. The Pareto front is shown in red.

Without additional information on the preferences of the decision-maker, all Pareto optimal solutions are considered equally good. Finding the Pareto front is usually prohibitive from the computational point of view. Therefore, the purpose is to provide a useful approximation of the Pareto front. As previously mentioned, evolutionary algorithms are very effective in approaching the Pareto front. Next, we develop a metaheuristic based on an evolutionary algorithm that has the definition of the chromosome and the construction of a feasible solution from it as novelties.

4. HEABBR: A Hybrid Evolutionary Algorithm for the BBRP

Evolutionary Algorithms are based on the ideas of reproduction and selection in populations. These algorithms use such ideas to provide close to optimal solutions to challenging optimization problems in small computational time [27–29]. They encode potential solutions as a string of symbols, called a chromosome, and associate with it a fitness value that gives an evaluation of the solution quality, usually based on the objective function values of the optimization problem in the encoded solution. After generating an initial population of chromosomes and evaluating their fitness, through successive iterations the evolutionary algorithm maintains a population of chromosomes, usually of the same size. By applying a crossover operator and a mutation operator to the chromosomes of the current population, offspring are generated. After assessing the fitness of these chromosomes, the new population is selected from the incumbent population (current population and offspring) by applying the selection operator. This process continues through successive iterations of the algorithm until the stopping criterion is met.

The algorithm HEABBR proposed in this paper maintains the general framework of a classical evolutionary algorithm but presents two relevant novelties. The first one is the definition of the chromosome. This does not directly encode a feasible solution. Instead, only the number of routes and the visited pickup points are identified. The idea of identifying only the nodes visited by the route in the chromosome has already been proposed in the context of the ring star problem [30,31]. However, it is worth mentioning that in the ring star problem all nodes have the same characteristics, while in the BBRP there are two kinds of nodes, pickup points and individuals. Moreover, in [31] there is only one ring (route) and there is no capacity constraint. Therefore, when a node has to be allocated, it can always be allocated to the closest visited ring node. In contrast, in the problem addressed in this paper, not all individuals will be able to be allocated to their nearest pickup point because of the bus capacity constraint. On the other hand, in the capacitated m -ring star problem dealt with in [30] there is a capacity constraint, but the number of rings (routes) is fixed. The second relevant novelty of the algorithm HEABBR

is the method applied to construct a feasible solution from the chromosome. It involves local search procedures allowing us to explore the neighbourhood of a tentative solution to try to improve its fitness. Moreover, the algorithm maintains a file with potentially efficient solutions. Next, we present a detailed description of the characteristics of the algorithm developed.

4.1. Chromosome Encoding

The chromosome is encoded as a $1 + |W|$ -dimensional vector

$$B = (b_0, b_1, \dots, b_{|W|}).$$

The first component b_0 is an integer which indicates the number of routes of the encoded solution. To be able to construct a feasible solution,

$$\left\lceil \frac{|U|}{Q} \right\rceil \leq b_0 \leq H.$$

The remaining components are binary and indicate if the corresponding pickup point i is available to be visited by a route or not, i.e., if individuals can be allocated to it:

$$b_i = \begin{cases} 1, & \text{if pickup point } i \text{ is available to be visited by a route} \\ 0, & \text{otherwise} \end{cases}, \quad i = 1, \dots, |W|.$$

As explained below, the algorithm always guarantees that $\sum_{i \in W} b_i \geq b_0$, i.e., b_0 routes can be constructed. Note that, as mentioned previously, chromosome B gives information on the number of routes and the visited pickup points but does not provide the routes themselves. Moreover, there is no information on how the individuals are allocated to the available pickup points. Hence, in general, we can construct several feasible solutions of the BBRP associated with the same chromosome. The purpose of the algorithm is to construct a good feasible solution, i.e., a solution that allows us to discard other feasible solutions which are dominated by it.

4.2. Construction of a Feasible Solution

The construction of a feasible solution involves two steps. The first one constructs a tentative solution, which is improved in the second step by applying local search procedures.

4.2.1. Tentative Solution

We propose to adapt the main ideas of the clustering algorithm developed by Fischetti et al. [32] for the generalized travelling salesman problem. First, the initial skeleton of the b_0 routes is built by sequentially selecting the pickup point ‘furthest’ from the depot and all the pickup points previously considered. For this purpose, first the pickup point i_1 , such that $b_{i_1} = 1$ and is the furthest from the depot in terms of routing cost, is singled out:

$$i_1 = \arg \max \{c_{0i} : i \in W, b_i = 1\}.$$

The first route is formed by connecting the depot and the node i_1 . Next, the pickup point i_2 , such that $b_{i_2} = 1$ and is the furthest from nodes $\{0, i_1\}$, is singled out:

$$i_2 = \arg \max \{\min \{c_{0i}, c_{i_1i}\} : i \in W \setminus \{i_1\}, b_i = 1\}.$$

The second route connects the depot and the node i_2 . The process is continued until the pickup point i_{b_0} , such that $b_{i_{b_0}} = 1$ and is the furthest from nodes $\{0, i_1, \dots, i_{b_0-1}\}$, is singled out:

$$i_{i_{b_0}} = \arg \max \{\min \{c_{0i}, c_{i_1i}, \dots, c_{i_{b_0-1}i}\} : i \in W \setminus \{i_1, \dots, i_{b_0-1}\}, b_i = 1\}.$$

The b_0 -th route connects the depot and i_{b_0} . Up to this point, we have the initial skeleton of the b_0 routes, each connecting the depot and a node $i \in \{i_1, \dots, i_{b_0}\}$.

Then, in sequence, each remaining node, i.e., $i \in V \setminus \{0, i_1, \dots, i_{b_0}\}$, is randomly taken and the following procedure is applied:

- if $i \in W$ and $b_i = 1$, the pickup point i is inserted in the route which provides the minimum insertion cost. At this time, i will be a visited pickup point.
- If $i \in W$ and $b_i = 0$, the pickup point is discarded.
- If $i \in U$, the individual located at node i is allocated to the pickup point visited by a route that provides the minimum walking distance, bearing in mind the bus capacity constraint.

Note that, at the end of this process, every pickup point with $b_i = 1$ in the chromosome will be visited by a route and every individual will be allocated to a visited pickup point (the closest pickup point at the time of the allocation).

It is worth mentioning that, after constructing the skeleton of the routes, all the remaining nodes (individuals and pickup points) must be randomly selected to apply the previous procedure. Otherwise, if we decided, for example, to consider first the pickup points to construct the routes and then consider the individuals' locations to allocate them, i.e., we decided to follow a location-routing-allocation strategy, we would not be able to get efficient solutions with less total walking distance. For instance, let us consider the network represented in Figure 1a. If we follow the location-routing-allocation strategy in which all the potential pickup points are selected to be visited, the routes would be the ones shown in Figure 3a, and so the allocation would be the one presented in Figure 3b. As a consequence, it would be impossible to get the solution shown in Figure 1c, which has less walking distance. Therefore, in general, a location-routing-allocation strategy does not work when trying to construct efficient solutions.

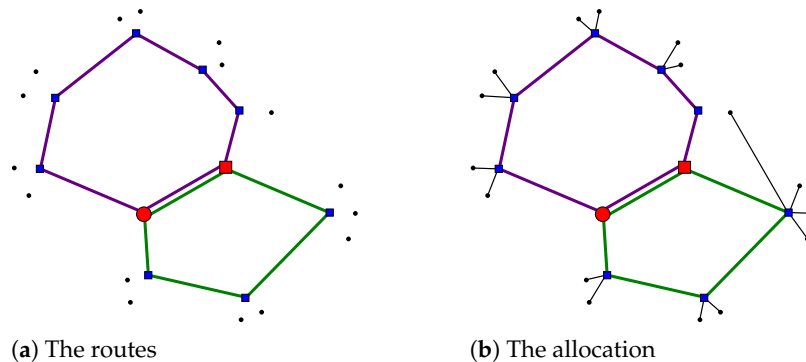


Figure 3. The location-routing-allocation strategy.

Having constructed the tentative feasible solution, denoted by X_1 , next we seek to improve it by applying four local search procedures, aiming to obtain a feasible solution that dominates X_1 . Three of the procedures affect only one of the objective functions and are applied to every solution. The other one involves simultaneously both objective functions and it is only applied to some solutions. The procedures are applied in the order in which they are explained in the paper. At the end of this process, if the feasible solution obtained is efficient with respect to the incumbent Pareto front, it is included in the Pareto front and the solutions that are now dominated are removed. Next, we explain in detail the four local search procedures.

4.2.2. Common Local Search Procedure: Reducing the Walking Distance

To improve the walking distance of the feasible solution X_1 , the individuals are optimally allocated to the visited pickup points by solving a transportation problem in which the sources are the current routes, each supplying Q units (the bus capacity), and the demand points are the individuals, each demanding 1 unit. The cost a_{mj} associated

with the source m and the demand point j is the walking distance from the location j to the closest pickup point of the route m :

$$a_{mj} = \min\{d_{ji} : i \in W_m\},$$

where W_m is the set of pickup points visited in the route m , $m = 1, \dots, b_0$.

By defining the variable f_{mj} equal to 1 if the individual j is allocated to route m , and 0 otherwise, this problem can be formulated as:

$$\begin{aligned} \min_{f_{mj}} \quad & \sum_{m=1}^{b_0} \sum_{j \in U} a_{mj} f_{mj} \\ \text{subject to} \quad & \sum_{j \in U} f_{mj} \leq Q, \quad m = 1, \dots, b_0, \\ & \sum_{m=1}^{b_0} f_{mj} = 1, \quad j \in U, \\ & f_{mj} \geq 0, \quad m = 1, \dots, b_0, j \in U. \end{aligned} \tag{3}$$

In accordance with the definition of a_{mj} , this problem implicitly assumes that when the individual located at j is allocated to the route m , he/she is allocated to the nearest pickup point in this route. This operation does not alter the routes, but the allocation of individuals. Therefore, it provides a feasible solution with the same or lower walking distance and with the same routing cost. Hence, the updated solution X_2 weakly dominates X_1 .

4.2.3. Common Local Search Procedure: 2-opt

Now, let us consider the feasible solution X_2 . To improve its routing cost, we apply the 2-opt local search procedure [33] to each of its routes. This operation provides a feasible solution X_3 with the same or lower routing cost and with the same walking distance. Hence, the updated solution X_3 weakly dominates X_2 , and thus X_1 .

4.2.4. Specific Local Search Procedure: Changing the Role of Nodes

The specific local search procedure is only applied to some solutions X_3 , which are selected to undergo the procedure with probability p_l . If the incumbent solution X_3 is not selected, X_3 goes directly to the next local search procedure explained in Section 4.2.5. Otherwise, in this case, we allow simultaneously a change in both objective functions by changing the role of a pickup point in the chromosome. Since there are two objective functions involved, the comparison of the solutions is not straightforward and a criterion must be provided to select a solution.

The two operations considered in this procedure are cycle reduction and cycle augmentation. For this purpose, we randomly select a pickup point $i \in W$ in the incumbent solution X_3 .

Cycle Reduction

If the pickup point i is visited by a route, i.e., $b_i = 1$, it is removed from the route (if it is not the only pickup point in the route) and the best allocation of the individuals is obtained by solving the updated transportation problem (3). Let X_4 be the updated solution, which coincides with X_3 if $b_i = 0$.

The termination condition explained below is checked. If the procedure continues, we consider the feasible solution X_4 , in which the pickup point i is not in any route, and the cycle augmentation operation is applied.

Cycle Augmentation

Let R_1, \dots, R_{b_0} be the routes in the solution X_4 . The pickup point i is inserted in the best position of the route R_1 and the best allocation of the individuals is obtained by solving the updated transportation problem (3). This provides the updated solution X_5 and the termination condition is checked.

If the process is not yet finished, we consider again the solution X_4 and insert the pickup point i in the best position of the route R_2 and optimally allocate the individuals by solving the updated transportation problem (3). This provides the updated solution X_6 and the termination condition is checked. If needed, the process continues with all the routes.

If the local search procedure is not finished after having completely analyzed the pickup point i , the incumbent solution remains X_3 and a new pickup point (without repetition) is randomly selected to undergo the cycle reduction and cycle augmentation operations.

Termination Condition for the Specific Local Search Procedure

Let X_s be the incumbent solution. The specific local search procedure is finished as soon as either the solution X_s dominates X_3 or the solution X_s is efficient with respect to the incumbent Pareto front. Since the number of pickup points is finite, this specific local search procedure will terminate in a finite number of iterations. Hence, at the end of this procedure, either the solution X_3 remains or it has been replaced by X_s . Whatever the case, the incumbent solution undergoes the following local search procedure. Notice that, as a consequence of the operations involved in this local search procedure, the pickup points which are visited by a route could have been changed. Hence, if needed, the chromosome is updated to match the characteristics of the incumbent solution.

4.2.5. Common Local Search Procedure: Removing Unused Pickup Point Locations and Combining Routes

At this time, we remove from the incumbent feasible solution all the pickup points with no individuals allocated to them and update the routes. Taking into account the triangular inequality, the solution finally obtained generally dominates the incumbent solution.

Finally, aiming to reduce the routing cost, routes with low occupancy are combined. For this purpose, by pairs, routes with less than Q individuals are combined, if possible, to get a single route with the same pickup points visited. This operation provides a feasible solution with the same or lower routing cost and with the same walking distance. Hence, the updated solution weakly dominates the original one.

This local search procedure, in general, reduces the number of visited pickup points as well as the number of routes. If the chromosome is updated to match the characteristics of this final solution, in the long term the population of chromosomes could consist of a set of chromosomes that would have a low number of routes and of available pickup points to be visited. Hence, to evaluate the impact of this issue on the performance of the algorithm, two variants of the algorithm are developed. The first one proposes to update the chromosome at this time, if needed. The second one does not update it, i.e., the chromosome involved in the crossover and mutation operators is the one available at the beginning of Section 4.2.5. This is one of the algorithm characteristics that will be analyzed in Section 5.

4.3. Initial Population

To construct the initial population, we randomly generate P_{size} chromosomes, where P_{size} refers to the population size. To build a chromosome, b_0 is randomly generated as an integer in the interval $\left[\left\lceil \frac{|U|}{Q} \right\rceil, H\right]$. To obtain the remaining $b_i, i = 1, \dots, |W|$, a number $p \in [0, 1]$ is randomly generated. Then, each pickup point is selected to be available to be visited in a route, $b_i = 1$, with probability p .

If a chromosome has fewer than b_0 pickup points that can be visited, i.e., $\sum_{i \in W} b_i < b_0$, the chromosome is repaired by switching the allele of $b_0 - \sum_{i \in W} b_i$ pickup points i currently having $b_i = 0$.

4.4. Crossover and Mutation Operators

For each parent population, the crossover and mutation operators provide us with offspring. We propose two different crossover operators, which are based on the uniform crossover and a variant of the one-point crossover. Regarding the uniform crossover, pairs of parents are randomly selected and one offspring is generated from each pair by taking each gene from one of the parents with a probability of 0.5. As an example:

Parent 1:	2	1	1	0	1	1	0	1	0	1	1	1
Parent 2:	3	0	1	0	0	1	1	0	0	1	0	1
Random Number:	0.11	0.60	0.96	0.29	0.07	0.75	0.48	0.65	0.85	0.65	0.38	0.26
Offspring:	2	0	1	0	1	1	0	0	0	1	1	1

With regard to the single point crossover, we distinguish the first gene corresponding to the number of routes from the remaining genes associated with pickup points. Hence, from a pair of parents randomly selected from the parent population, one offspring is generated. The first gene is taken from one parent with a probability of 0.5. Then, after randomly generating a crossover point, the genes to the left of the crossover point are taken from the first parent and the genes to the right are taken from the second parent. Assuming that the crossover point is after the 7th gene corresponding to a pickup point:

Parent 1:	2	1	1	0	1	1	0	1	0	1	1	1
Parent 2:	3	0	1	0	0	1	1	0	0	1	0	1
Random Number:	0.11											
Offspring:	2	1	1	0	1	1	0	1	0	1	0	1

Whichever crossover is applied, with probability p_m each chromosome is chosen to undergo mutation. After a chromosome is selected, a gene corresponding to a pickup point is randomly selected and its allele value is switched. The gene associated with the number of routes is never changed in this step.

After crossover and mutation operators have been applied, if needed the chromosome is repaired as indicated above to have at least as many pickup points that can be visited as b_0 . Finally, the feasible solutions corresponding to each of the offspring are computed as indicated in Section 4.2.

4.5. Fitness Evaluation and Population Handling

To evaluate the quality of the chromosomes, the fitness function computes both objective functions (the routing cost Z_1 and the walking distance Z_2) at the feasible solution associated with the chromosome.

To handle the populations, the best members of the current population and the offspring, according to the NSGA-II (Nondominated Sorting Genetic Algorithm) [34], are selected to be part of the next population. NSGA-II ranks the current population plus offspring into several non-dominated fronts. All nondominated points are in the first front. Then, these points are eliminated and the second nondominated front is formed with all the non-dominating points among the remainder. This process continues until all chromosomes have been classified and are assigned a nondomination rank. Moreover, the crowding distance is also computed for each chromosome to estimate the density of solutions around its corresponding solution. By using the nondomination rank, the new population selects the best different P_{size} chromosomes. If there are ties, the crowding distance is used.

Moreover, in the implementations of HEABBR, an archive population is maintained. This archive includes solutions computed by the algorithm which can be nondominated. It is updated at each iteration.

5. Computational Experiment

A computational experiment has been carried out to analyze the performance of HEABBR. The numerical experiments have been performed on a PC Intel Core i7-9700 with 3.0 gigahertz, having 32.0 gigabytes of RAM and Windows 10 64-bit as the Operating

System. The code has been written in C++, TDM-GCC 4.9.2. For solving problem (3) we have selected the simplex algorithm for Transportation Problems [35].

In order to test HEABBR, we have selected the set of instances proposed in [22]. This set consists of 112 instances. The number of potential pickup points ranges between 5 and 80, the number of individuals varies from 25 to 800 and the bus capacity is 25 or 50. These instances also include a maximum walking distance parameter which has been omitted in this study. Since there is no parameter H , we have set $H = 2 \lceil \frac{|U|}{Q} \rceil$. This upper bound allows us to avoid having for sure at least two routes that can be combined into a single route with at most Q individuals allocated to it. The instances have been classified into five sets in accordance with the number of potential pickup points. Their characteristics are described in Table 1. Instances with odd ID have bus capacity equal to 25, and those with even ID have bus capacity equal to 50. The destination and the depot are located at the same place.

Table 1. Sets of instances.

Set	ID	Number of Pickup Points	Computing Time (min)
S_1	1 to 24	5	2
S_2	25 to 48	10	4
S_3	49 to 72	20	8
S_4	73 to 96	40	16
S_5	97 to 112	80	32

We have studied the effect and influence of two factors on the quality of the Pareto front yielded by HEABBR. The first factor is the crossover operator used. The second factor refers to whether to update or not the chromosome at the end of Section 4.2.5. Each combination of these factors provides a configuration of HEABBR. They are shown in Table 2. Each test instance has been solved under each configuration, i.e., it has been solved 4 times. The remaining parameters of the algorithm have been set at $P_{size} = 50$ chromosomes, $p_m = 0.5$ and $p_l = 0.5$. The termination condition has been established in terms of total computing time. In accordance with the previous classification of instances, the computing time has been set at the minutes indicated in the last column of Table 1.

Table 2. HEABBR configurations in the computational experiment.

Conf.	Crossover	Chromosome Update
1	Single point	No update
2	Single point	Update
3	Uniform	No update
4	Uniform	Update

For the purpose of evaluating the quality of the Pareto front approximations, we have used the unary hypervolume indicator I_H^- [36] and the additive ϵ indicator $I_{\epsilon+}^1$ [37]. In order to compute both indicators, for each instance, the reference set Z_N^* is computed, which consists of the union of the outputs obtained throughout the whole experiment. i.e., all the nondominated points available. Then, for output set A , I_H^- computes the area of the objective space which is weakly dominated by Z_N^* and not by A . On the other hand, $I_{\epsilon+}^1$ computes the minimum factor by which A has to be translated in the objective space to weakly dominate Z_N^* . The closer the indexes to zero, the better the approximation. Both indexes have been computed by using the performance assessment tool suite provided in PISA [38]. Since these numbers are, in all cases, near-zero, we have multiplied the indicators by 100 to be displayed in the following tables.

Table 3 displays, for each configuration and sets S_1 to S_5 , the mean value, the standard deviation and the maximum of the number of points in the Pareto front and the indicators I_H^- and $I_{\epsilon^+}^1$ values. For set S_1 , all configurations provide the same Pareto front, thus providing a value of zero for both indicators. Therefore, we eliminate this set from the remaining analysis as all the configurations are equally good.

Figure 4a,b display the value of the corresponding indicators for each of the 88 instances of sets S_2 to S_4 , according to the configuration. Regarding the I_H^- indicator, for sets S_2 to S_4 , configurations 1 and 3 provide the lesser values. Set S_5 shows larger variability and for certain instances, configuration 2 provides the best results. Similar findings are obtained for the $I_{\epsilon^+}^1$ indicator, although there is a higher variability. Figure 5a,b display, for each set and configuration, the value of the corresponding indicator as a blue ball. A solid red ball is associated with the mean value of the corresponding indicator over all instances in the set. From these figures, we can also observe the differences among the mean values of the four configurations as well as how the variability increases as the number of pickup points increases.

Table 3. Aggregated results by a set of instances (indicators I_H^- and $I_{\epsilon^+}^1$ have been multiplied by 100).

Conf.	Set	Size of the Pareto Front			I_H^-			$I_{\epsilon^+}^1$		
		Mean	Stdev	Max	Mean	Stdev	Max	Mean	Stdev	Max
1	S_1	10.83	3.92	20.00	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	41.88	14.69	71.00	0.03	0.08	0.38	0.49	0.83	2.88
	S_3	109.00	41.30	182.00	0.20	0.24	0.81	1.23	0.95	3.48
	S_4	142.33	67.49	272.00	1.13	1.03	3.60	2.66	2.01	8.98
	S_5	120.44	66.77	269.00	2.48	2.30	8.84	5.59	4.56	14.23
2	S_1	10.83	3.92	20.00	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	40.46	13.49	69.00	0.32	0.57	2.53	1.68	2.28	9.11
	S_3	108.54	41.55	185.00	0.82	1.39	6.54	2.54	2.81	11.78
	S_4	149.13	70.18	274.00	1.80	1.26	4.31	4.00	2.36	8.32
	S_5	139.56	73.05	268.00	2.18	1.38	5.55	4.16	2.04	8.09
3	S_1	10.83	3.92	20.00	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	41.67	14.73	72.00	0.03	0.06	0.22	0.41	0.64	2.40
	S_3	112.71	42.59	194.00	0.11	0.14	0.57	0.94	0.68	2.51
	S_4	141.04	75.35	286.00	0.92	0.75	2.46	2.29	1.51	4.95
	S_5	106.44	66.40	273.00	2.67	2.32	9.95	5.42	3.83	14.93
4	S_1	10.83	3.92	20.00	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	40.71	13.73	70.00	0.34	0.61	2.66	1.73	2.27	9.11
	S_3	109.50	43.18	190.00	0.78	1.45	6.85	2.27	2.73	11.78
	S_4	139.42	73.79	265.00	2.03	1.53	6.13	4.32	2.72	11.33
	S_5	113.38	68.08	244.00	3.06	1.32	5.51	5.73	2.04	8.53

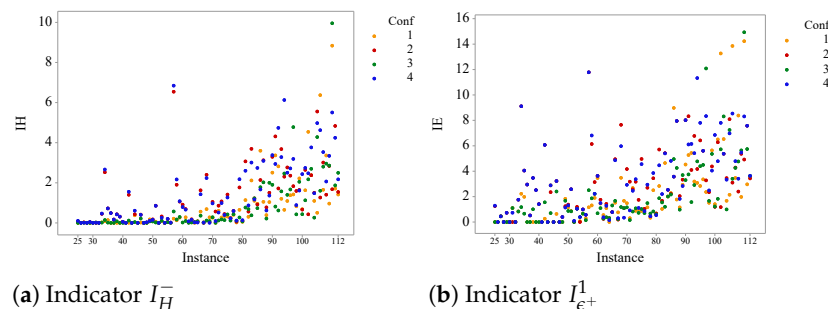


Figure 4. Scatterplot of the indicator values of the four configurations versus the instance.

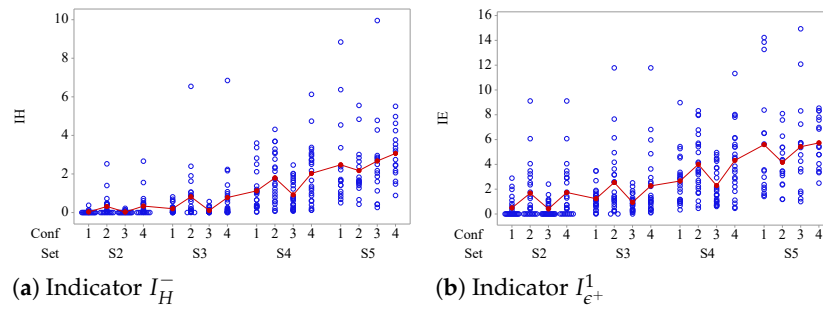


Figure 5. Value of the corresponding indicator for the instances in each set and configuration in blue. The red ball shows the mean value of the corresponding set instances.

Figure 6a,b shows the main factor for the indicators I_H^- and I_{e+}^1 , respectively. Figure 7a,b show the interaction plot for both indicators. Looking at them, we can conclude that the chromosome update is the most relevant factor. Whatever the crossover selected, the best choice is not to update the chromosome. Moreover, the configuration which globally provides the best mean results is configuration 3, i.e., to combine uniform crossover with no chromosome update. However, it should be noted that, when distinguishing by set, set S_5 , which includes the larger instances, performs slightly differently, since, only, in this case, configuration 2 would show slightly better results.

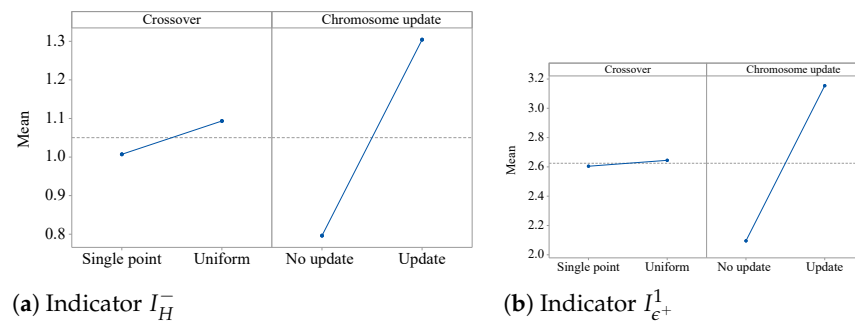


Figure 6. Main plot of both indicators.

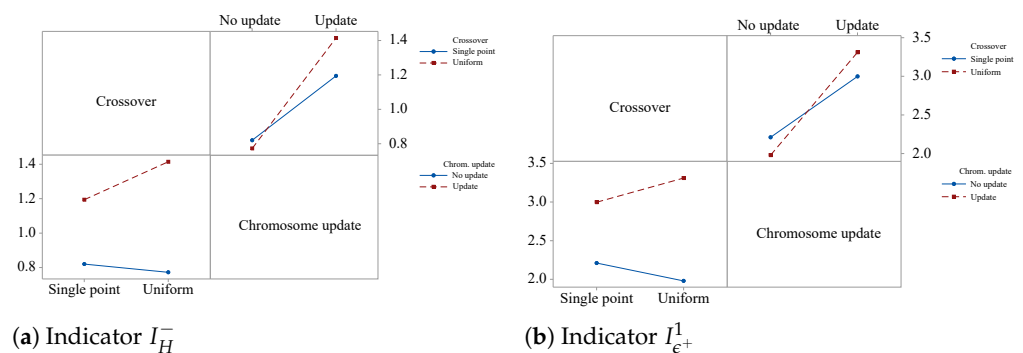


Figure 7. Interaction plot of both indicators.

As an illustration, Table 4 shows the number of points in the Pareto front and the value of both indicators provided by configuration 3 in each instance. Also, Figure 8 shows the network and the Pareto front for instances 35 and 78.

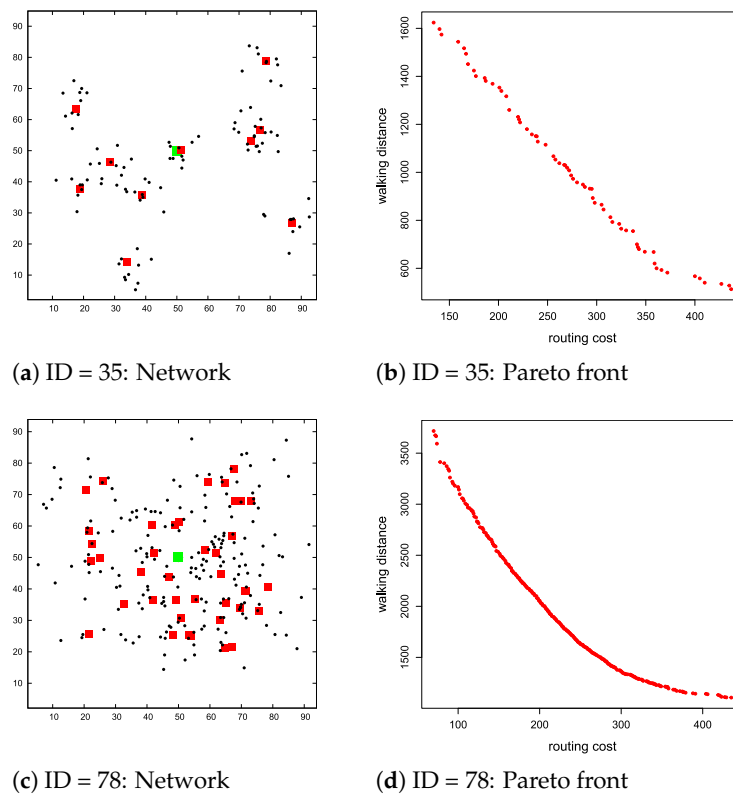


Figure 8. Network and Pareto front of instances 35 and 78. The green square represents the depot. The red squares represent the pickup points. The black circles are the individuals.

Table 4. Configuration 3: Number of points in the Pareto front (Pf) and indicator I_H^- and $I_{\epsilon^+}^1$ values (multiplied by 100).

Ins.	Pf	I_H^-	$I_{\epsilon^+}^1$	Ins.	Pf	I_H^-	$I_{\epsilon^+}^1$	Ins.	Pf	I_H^-	$I_{\epsilon^+}^1$	Ins.	Pf	I_H^-	$I_{\epsilon^+}^1$	Ins.	Pf	I_H^-	$I_{\epsilon^+}^1$
1	17	0.00	0.00	25	47	0.00	0.00	49	134	0.07	1.67	73	202	0.17	1.08	97	97	4.78	12.08
2	11	0.00	0.00	26	55	0.00	0.00	50	178	0.07	0.87	74	274	0.14	0.72	98	273	0.43	1.67
3	16	0.00	0.00	27	59	0.01	0.45	51	167	0.02	0.84	75	268	0.42	1.02	99	117	2.05	5.36
4	10	0.00	0.00	28	36	0.00	0.00	52	169	0.00	0.00	76	286	0.12	0.70	100	234	0.43	1.22
5	9	0.00	0.00	29	38	0.00	0.00	53	109	0.11	1.04	77	197	0.43	1.49	101	125	1.12	1.82
6	11	0.00	0.00	30	32	0.00	0.00	54	143	0.00	0.39	78	243	0.11	0.67	102	161	2.17	5.72
7	9	0.00	0.00	31	35	0.02	1.10	55	88	0.03	0.66	79	113	0.27	0.83	103	100	2.70	8.31
8	16	0.00	0.00	32	37	0.00	0.00	56	73	0.00	0.14	80	113	0.07	0.60	104	121	0.26	0.99
9	20	0.00	0.00	33	72	0.01	0.83	57	133	0.57	2.51	81	130	0.77	1.87	105	63	4.28	7.24
10	14	0.00	0.00	34	56	0.14	1.17	58	159	0.17	1.86	82	233	0.69	1.54	106	66	1.60	3.40
11	11	0.00	0.00	35	64	0.01	0.66	59	126	0.29	2.22	83	129	0.38	1.20	107	46	2.80	4.26
12	10	0.00	0.00	36	43	0.00	0.00	60	194	0.02	0.69	84	175	1.22	2.01	108	70	2.96	4.77
13	11	0.00	0.00	37	50	0.00	0.00	61	122	0.09	1.10	85	102	0.73	2.16	109	57	2.86	5.64
14	10	0.00	0.00	38	48	0.08	1.01	62	132	0.01	0.28	86	159	1.78	4.95	110	48	9.95	14.93
15	11	0.00	0.00	39	35	0.00	0.00	63	76	0.02	0.38	87	75	2.02	4.26	111	55	1.87	5.74
16	9	0.00	0.00	40	55	0.00	0.00	64	78	0.04	0.31	88	104	0.21	0.98	112	70	2.50	3.64
17	6	0.00	0.00	41	19	0.00	0.00	65	86	0.07	1.30	89	76	2.00	3.69				
18	10	0.00	0.00	42	47	0.12	1.55	66	100	0.36	1.67	90	62	1.90	3.30				
19	7	0.00	0.00	43	19	0.00	0.00	67	57	0.01	0.24	91	104	0.81	3.74				
20	11	0.00	0.00	44	40	0.04	0.70	68	99	0.12	1.18	92	91	0.62	1.57				
21	5	0.00	0.00	45	21	0.00	0.00	69	51	0.09	1.09	93	76	1.57	2.91				
22	13	0.00	0.00	46	38	0.22	2.40	70	113	0.28	1.15	94	57	2.46	4.77				
23	2	0.00	0.00	47	13	0.00	0.00	71	30	0.00	0.15	95	53	1.40	4.39				
24	11	0.00	0.00	48	41	0.00	0.00	72	88	0.12	0.83	96	63	1.67	4.49				

6. Conclusions

Most of the literature on bus routing problems, especially school bus routing problems, assumes the routing cost as the objective to be minimized, leaving aside the distance travelled by individuals or at most imposing certain upper limits on it. This paper addresses

a variant of the bus routing problem in which both the routing cost and the distance walked by individuals to reach their allocated pickup point are taken into account. Moreover, in this problem neither the pickup points which are visited in the bus routes nor the allocation of individuals to them are fixed a priori, i.e., these are decisions that must be made together with the construction of the routes themselves. This biobjective approach whose purpose is to provide the Pareto front allows the decision-maker to have available a variety of efficient solutions.

The problem considered is highly combinatorial since three simultaneous decisions have to be made: which pickup points will be visited by a route, to which pickup point each individual will be allocated, and which routes will be designed. Therefore, the paper focuses on approaching the Pareto front. For this purpose, a metaheuristic is developed which uses the ideas of an evolutionary algorithm. The proposal is new in the sense that each chromosome does not refer to a feasible solution but to a set of feasible solutions. From this set, a potentially good solution is chosen from a tentative solution by using several local search procedures. A computational experiment has been carried out to select the best configuration of the algorithm and to show its performance.

Challenging future lines of research include considering different objective functions related to environmental issues, such as, reducing carbon footprint or achieving buses as fully loaded as possible. Regarding other measures of the individuals' comfort, minimizing the maximum distance walked by an individual or minimizing the ride time could be considered.

Author Contributions: Conceptualization, H.I.C., C.G. and J.A.I.; methodology, H.I.C., C.G. and J.A.I.; software, J.A.I.; statistical analysis, C.G.; formal analysis, H.I.C., C.G. and J.A.I.; writing—original draft preparation, H.I.C.; writing—review and editing, H.I.C., C.G. and J.A.I. All authors have read and agreed to the published version of the manuscript.

Funding: Gobierno de Aragón: E41-20R; AEI, Spanish Ministry of Science, Innovation and Universities: PID2019-104263RB-C43

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

Abbreviations

The following abbreviations are used in this manuscript:

BBRP	Biobjective Bus Routing Problem
BRP	Bus Routing Problem
CVRP	Capacitated Vehicle Routing Problem
GRASP + VND	Greedy Randomized Adaptive Search + Variable Neighborhood Descent
HEABBR	Hybrid Evolutionary Algorithm for the BBR problem
NSGA-II	Nondominated Sorting Genetic Algorithm-II
PISA	A Platform and Programming Language Independent Interface for Search Algorithms
SBRP	School Bus Routing Problem

References

1. Ehrgott, M. *Multicriteria Optimization*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2005.
2. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: Chichester, UK, 2001.
3. Coello Coello, C.A.; González Brambila, S.; Figueroa Gamboa, J.; Castillo Tapia, M.G.; Hernández Gómez, R. Evolutionary multiobjective optimization: Open research areas and some challenges lying ahead. *Complex Intell. Syst.* **2020**, *6*, 221–236. [[CrossRef](#)]
4. Ehrgott, M.; Gandibleux, X. Approximative solution methods for multiobjective combinatorial optimization. *Top* **2004**, *12*, 1–89. [[CrossRef](#)]
5. Jozefowicz, N.; Semet, F.; Talbi, E.G. Multi-objective vehicle routing problems. *Eur. J. Oper. Res.* **2008**, *189*, 293–309. [[CrossRef](#)]
6. Potvin, J.Y. State-of-the-art review—Evolutionary algorithms for vehicle routing. *INFORMS J. Comput.* **2009**, *21*, 517–656. [[CrossRef](#)]
7. Newton, R.M.; Thomas, W. Designing of school bus routes by computer. *Socio-Econ. Plan. Sci.* **1969**, *3*, 75–85. [[CrossRef](#)]

8. Braekers, K.; Ramaekers, K.; Van Nieuwenhuysse, I. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2016**, *99*, 300–313. [[CrossRef](#)]
9. Eksioglu, B.; Volkan Vural, A.; Reisman, A. The vehicle routing problem: A taxonomic review. *Comput. Ind. Eng.* **2009**, *57*, 1472–1483. [[CrossRef](#)]
10. Tan, S.Y.; Yeh, W.C. Vehicle Routing Problem: State-of-the-Art Classification and Review. *Appl. Sci.* **2021**, *11*, 295. [[CrossRef](#)]
11. Park, J.; Kim, B.I. The school bus routing problem: A review. *Eur. J. Oper. Res.* **2010**, *202*, 311–319. [[CrossRef](#)]
12. Ellegood, W.A.; Solomon, S.; North, J.; Campbell, J.F. School bus routing problem: Contemporary trends and research directions. *Omega* **2020**, *95*, 102056. [[CrossRef](#)]
13. Peker, G.; Eliiyi, D.T. Shuttle bus service routing: A systematic literature review. *Pamukkale Univ. J. Eng. Sci.* **2022**, *28*, 160–172.
14. Corberán, A.; Fernández, E.; Laguna, M.; Martí, R. Heuristic solutions to the problem of routing school buses with multiple objectives. *J. Oper. Res. Soc.* **2002**, *53*, 427–435. [[CrossRef](#)]
15. Pacheco, J.; Caballero, R.; Laguna, M.; Molina, J. Bi-objective bus routing: An application to school buses in rural areas. *Transp. Sci.* **2013**, *47*, 397–411. [[CrossRef](#)]
16. Dasdemir, E.; Testik, M.C.; Öztürk, D.T.; Şakar, C.T.; Güleriyüz, G.; Testik, O.M. A multi-objective open vehicle routing problem with overbooking: Exact and heuristic solution approaches for an employee transportation problem. *Omega* **2022**, *108*, 102587. [[CrossRef](#)]
17. Dulac, G.; Ferland, J.A.; Forgues, P.A. School bus routes generator in urban surroundings. *Comput. Oper. Res.* **1980**, *7*, 199–213. [[CrossRef](#)]
18. Chapleau, L.; Ferland, J.A.; Rousseau, J.M. Clustering for routing in densely populated areas. *Eur. J. Oper. Res.* **1985**, *20*, 48–57. [[CrossRef](#)]
19. Bowerman, R.; Hall, B.; Calamai, P. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transp. Res. Part A Policy Pract.* **1995**, *29*, 107–123. [[CrossRef](#)]
20. Riera-Ledesma, J.; Salazar-González, J.J. Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Comput. Oper. Res.* **2012**, *39*, 391–404. [[CrossRef](#)]
21. Riera-Ledesma, J.; Salazar-González, J.J. A column generation approach for a school bus routing problem with resource constraints. *Comput. Oper. Res.* **2013**, *40*, 566–583. [[CrossRef](#)]
22. Schittekat, P.; Kinable, J.; Sörensen, K.; Sevaux, M.; Spieksma, F.; Springael, J. A metaheuristic for the school bus routing problem with bus stop selection. *Eur. J. Oper. Res.* **2013**, *229*, 518–528. [[CrossRef](#)]
23. Kinable, J.; Spieksma, F.C.R.; Berghe, G.V. School bus routing—a column generation approach. *Int. Trans. Oper. Res.* **2014**, *21*, 453–478. [[CrossRef](#)]
24. Calvete, H.I.; Galé, C.; Iranzo, J.A.; Toth, P. A partial allocation local search matheuristic for solving the school bus routing problem with bus stop selection. *Mathematics* **2020**, *8*, 1214. [[CrossRef](#)]
25. Bögl, M.; Doerner, K.F.; Parragh, S.N. The school bus routing and scheduling problem with transfers. *Networks* **2015**, *65*, 180–203. [[CrossRef](#)] [[PubMed](#)]
26. Calvete, H.I.; Galé, C.; Iranzo, J.A.; Toth, P. The bilevel school bus routing problem with student choice: A bilevel approach and a simple and effective metaheuristic. *Int. Trans. Oper. Res.* **2021**. [[CrossRef](#)]
27. Affenzeller, M.; Wagner, S.; Winkler, S.; Beham, A. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*; Chapman & Hall/CRC: London, UK, 2009.
28. Chion, R.; Weise, T.; Michalewicz, Z. (Eds.) *Variants of Evolutionary Algorithms for Real-World Applications*; Springer: Berlin/Heidelberg, Germany, 2012.
29. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 1996.
30. Calvete, H.I.; Galé, C.; Iranzo, J.A. An evolutionary algorithm for the biobjective capacitated m-ring star problem. In *Algorithmic Decision Theory AFT 2013; Lecture Notes in Artificial Intelligence*; Perny, P., Pirlot, M., Tsoukiàs, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8176, pp. 116–129.
31. Calvete, H.I.; Galé, C.; Iranzo, J.A. MEALS: A multiobjective evolutionary algorithm with local search for solving the bi-objective ring star problem. *Eur. J. Oper. Res.* **2016**, *250*, 377–388. [[CrossRef](#)]
32. Fischetti, M.; Salazar-González, J.J.; Toth, P. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Oper. Res.* **1997**, *45*, 378–394. [[CrossRef](#)]
33. Croes, G.A. A method for solving traveling-salesman problems. *Oper. Res.* **1958**, *6*, 791–812. [[CrossRef](#)]
34. Deb, K.; Pratap, A.; Agrawal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
35. MacDonald, D.T. C++ Implementation of the Transportation Simplex Algorithm. Available online: <https://github.com/engine99/transport-simplex> (accessed on 29 March 2022).
36. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [[CrossRef](#)]
37. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.M.; Grunert da Fonseca, V. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132. [[CrossRef](#)]
38. Bleuler, S.; Laumanns, M.; Thiele, L.; Zitzler, E. PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In *Evolutionary Multi-Criterion Optimization (EMO 2003); Lecture Notes in Computer Science*; Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 494–508.