Memetic Algorithm for the Heterogeneous Fleet School Bus Routing Problem

Leonardo de Pádua Agripa Sales[1]; Cristiano Sousa Melo[2]; Tibérius de Oliveira e Bonates[3]; and Bruno de Athayde Prata[4]

Abstract: The school bus routing problem is a hard, widely studied combinatorial optimization problem. However, little attention has been paid in the literature to the integration between the school bus routing problem and the design of the underlying network. This paper aims to present a new variant of the problem in which the following issues are taken into consideration: the determination of the set of stops to visit, the allocation of students to stops, the generation of routes, and the utilization of a heterogeneous fleet, with different fixed costs and capacities. It is presented as an integer programming formulation, a lower-bound technique, as well a greedy genetic and a memetic algorithm for the heterogenous fleet school bus routing problem (HFSBRP). The integer programming formulation has shown limited application to the solution of large size instances. Computational results on a set of 100 instances provide evidence of the quality of the solutions found by the memetic algorithm on large instances. **DOI:** 10.1061/(ASCE)UP.1943-5444.0000454. *© 2018 American Society of Civil Engineers.*

Author keywords: Evolutionary algorithms; Vehicle-routing problems; School bus transportation; Combinatorial optimization; Metaheuristics.

Introduction

The vehicle-routing problem (VRP) is a hard and widely studied combinatorial optimization problem. In the last decades, several variants of the classical problem have been proposed.

A particular variant has received some attention over the last years: the school bus routing problem (SBRP) (Park and Kim 2010). In the SBRP, the process of route generation must consider bus stop locations (boarding and arrival points for students), the time window for the transport, and student allocation for the stops. Because of these characteristics, the SBRP introduces additional difficulties to a problem that is already notoriously difficult.

For the resolution of the SBRP, Table 1 summarizes several approaches proposed in the literature. On the basis of a literature review, one can observe that the considered premises of the modeling lead to a different variant of the SBRP. As Li and Fu (2002) pointed out, the SBRP seems to be problem dependent because of its many peculiarities; therefore, there is no approach that dominates the others in solving it. For example, whereas Dulac et al. (1980), Chapleau et al. (1985), Bowerman et al. (1995), Fügenschuh (2009), Riera-Ledesma and González (2012, 2013), and Schittekat et al. (2013), along with this study, consider he walking distance as an upper bound in the model, Bowerman et al. (1995) considered it not only an upper bound but also one of the objectives of their multi-objective optimization heuristic, which aims to minimize total student walking distance.

Most of the literature proposes heuristic approaches to solve the SBRP. The seminal paper by Newton and Thomas (1969) proposed the first heuristic for the SBRP, which consisted of a two-step procedure: first, the problem is solved as a traveling-salesman problem with a heuristic, generating a single route that visits all stops. Second, using two simple algorithms, this single route is partitioned to provide individual bus routes and schedules that satisfy all restraints. This method was improved by Newton and Thomas (1974) by considering multiple schools.

Bennett and Gazis (1972) proposed a method similar to the Clarke and Wright (1964) method, differing with it by considering asymmetries in the matrix of shortest distances between stops, differences in the origin and destination of buses, and requirements pertaining to the comfort and safety of students. Verderber (1974) also used the Clarke and Wright method to cluster stops to form the route structure. However, they also noted the practical utility of the method, which requires tools that a layperson can use to generate reports and modify and update machine-produced results. Gavish and Shlifer (1979) proposed a binary programming model for the SBRP, basing it on the Clarke and Wright method but changing the objective functions and some restrictions. They also applied their method to the delivery problem, the school bus problem, the assignment of buses to schedules, the combining truck trip problem, and the traveling salesman problem. Dulac et al. (1980) evaluated several techniques employed in SBRP, such as the Clarke and Wright method, the Yellow (1970), insertion (Rosenkrantz et al. 1974), and the Gillet and Miller (1974) methods, and compared results. They concluded that each instance should be solved several times with these different routing techniques, following with branch exchange procedures, to ensure reliable solutions.

Chapleau et al. (1985) noted that the Clarke and Wright method uses more buses than required because it minimizes the

[1]Graduate Student, Logistics and Network Infrastructure Laboratory, Federal Univ. of Ceará, Campus do Pici, Bl. 703, Fortaleza, CE, 60.455-960, Brazil. ORCID: https://orcid.org/0000-0002-7237-3418. Email: leonardosales@alu.ufc.br

[2]Graduate Student, Logistics and Network Infrastructure Laboratory, Federal Univ. of Ceará, Campus do Pici, Bl. 703, Fortaleza, CE, 60.455-960, Brazil. Email: cristianomelo_88@hotmail.com

[3]Professor, Dept. of Applied Statistics and Mathematics, Federal Univ. of Ceará, Campus do Pici, Bl. 910, Fortaleza, CE, 60.455-760, Brazil. Email: tb@ufc.br

[4]Professor, Dept. of Industrial Engineering, Federal Univ. of Ceará, Campus do Pici, Bl. 714, Fortaleza, CE, 60.440-554, Brazil (corresponding author). Email: baprata@ufc.br

Note. This manuscript was submitted on December 16, 2016; approved on December 19, 2017; published online on April 12, 2018. Discussion period open until September 12, 2018; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Urban Planning and Development***, © ASCE, ISSN 0733-9488.

Table 1. Approaches from the literature

References	Strategy	Objective function	Heterogeneous fleet	Fleet cost	Route cost	Time windows	Multiple schools	Stop multiple buses	Maximum walking distance
Newton and Thomas (1969)	Heuristic	Minimize total bus travel time	—	—	—	X	—	X	—
Bennett and Gazis (1972)	Heuristic	Minimize total bus and student travel time	X	—	—	—	—	X	—
Verderber (1974)	Heuristic	Minimize total bus travel time	X	—	—	X	—	X	—
Newton and Thomas (1974)	Heuristic	Minimize total bus travel time and number of routes	X	—	—	X	X	X	—
Gavish and Shlifer (1979)	Integer programming	Minimize fleet and routing costs	—	X	X	X	—	—	—
Dulac et al. (1980)	Heuristic	Minimize total bus travel length and number of routes	X	—	—	X	—	X	X
Chapleau et al. (1985)	Heuristic	Minimize number of routes	X	—	—	—	—	—	X
Bowerman et al. (1995)	Heuristic	Minimize number of routes, total bus travel length, total student walking distance, variation in number of students, and variation in total bus travel length	—	—	—	—	—	—	X
Fügenschuh (2009)	Linear programming	Minimize number of buses and deadhead trips	—	—	—	X	—	—	X
Park et al. (2012)	Mixed-integer programming	Minimize number of buses	X	—	—	X	X	—	—
Riera-Ledesma and González (2012)	Mixed-integer programming	Minimize number of routes, total bus travel length, and variation in total bus travel length	—	—	X	—	—	—	X
Riera-Ledesma and González (2013)	Mixed-integer programming	Minimize number of routes, total bus travel length, and total student walking distance	—	—	X	—	—	—	X
Schittekat et al. (2013)	Matheuristic	Minimize total bus travel length	—	—	—	—	—	—	X
Souza Lima et al. (2016)	Heuristic	Minimize fleet and routing costs	X	X	X		X	—	—
Caceres et al. (2017)	Dynamic programming	Minimize number of buses and total bus travel length	—		X	X	X	—	X
This study	Memetic algorithm	Minimize fleet and routing costs	X	X	X	—	—	X	X

total distance traveled by a bus rather than the number of routes. Therefore, they developed a heuristic that uses the "group first, route second" routing strategy as in the sweep method proposed by Gillet and Miller (1974). They reported results that indicate improvement in the overall design of the bus routes generated with this approach, as shown by a decrease in the number of routes without an increase in either the average walking distance or the average route length per student.

Bowerman et al. (1995) took a very different approach compared with earlier SBRP studies and proposed a multiobjective heuristic that minimized number of routes, bus travel length, student walking distance, varying number of students, and bus travel length. The algorithm first clusters students into groups using a districting algorithm and then generates a school bus route and bus stops for each cluster by combining a set covering and a traveling salesman algorithm.

Fügenschuh (2009) noted that earlier proposed methods did not consider the possibility of changing school start times in order to reduce the number of buses employed; nor did they consider students changing buses. Therefore, he developed an integer programming problem based on the vehicle-routing problem with time windows to consider these issues. With this approach, it is possible to reduce the number of deployed buses by 10–25%.

Riera-Ledesma and González (2012) addressed a family of school bus routing problems involving simultaneously assigning students to and designing bus routes for a set of bus stops. They employed the Clarke and Wright method to generate initial solutions and improved their model in Riera-Ledesma and González (2013) by considering more constraints on bus routes, such as maximum walking distance, maximum length a bus can travel, and minimum number of students that a vehicle has to pick up.

Schittekat et al. (2013) presented a mathematical formulation that integrates the student allocation and school bus routing problems in developing a quite robust approach for solving the SBRP. However, they did not take into account an important characteristic of the real-world problem: a fleet composed of different types of buses (heterogeneous fleet).

Souza Lima et al. (2016) proposed five heuristics to address the capacitated rural school bus routing problem featuring mixed loads, a heterogeneous fleet, and the same school starting time. However, they did not employ or compare their algorithms with exact methods. In contrast the study discussed in this paper focused on students living in high-density areas. In urban areas, the mixed-load problem is not usual, so we did not study it. However, the restraint employed in Souza Lima et al. (2016) of visiting a bus stop exactly once would clearly detract from solutions obtained for high-density areas. Therefore, we allowed a bus stop to be visited by more than one bus. Furthermore, an alternative for considering mixed loads is the mixed-load improvement algorithm proposed by Park et al. (2012), which can be applied to a single-load plan generated by traditional algorithms and converted into a mixed-load plan using a simple relocation operator.

Caceres et al. (2017) found that most earlier studies focused on deterministic routing problems that considered a known student demand and a fixed travel time. Thus, they considered these parameters stochastic and then overbooked the buses to decrease the number of buses needed. They developed a dynamic programming model to apply overbooking policies in a real-world school district. To consider the uncertainty of the buses' total travel time, they included a constraint limiting the probability of being late to school. Their results indicate desirable cost savings in terms of total number of buses used.

In order for public transport operators to generate optimal bus headways, perfect-competition and imperfect-competition models were formulated by Feng et al. (2016) for two bus routes to maximize every route's operating profit. A mixed-integer linear programming (MILP) model was developed by Pan et al. (2015) to optimize service area and transit route planning concurrently for a flexible feeder transit system serving irregularly shaped and gated communities. Khakbaz et al. (2016) studied a new bus routing problem formulation for the park-and-ride system. To determine high-quality routes in acceptable computational times, they developed an efficient hybrid genetic algorithm and an ant colony optimization algorithm. This study raised questions about whether employing metaheuristics such as a genetic algorithm to solve the SBRP could also return interesting results.

In this paper, we describe a new variant of the SBRP that takes these issues into consideration, which we refer as the heterogeneous fleet school bus routing problem, or simply the HFSBRP. To the best of the authors' knowledge, there is no variant of the HFSBRP in the literature that is similar to the one we have developed. The paper presents a memetic algorithm for the HFSBRP. Regarding the problem under study, an innovative integer programming formulation is described. By way of comparison with the developed memetic algorithm, both a lower-bound technique and a heuristic algorithm are presented. We carried out a series of computational experiments, solving the proposed model using a state-of-the-art integer programming solver. We used linear relaxation to obtain the lower bounds on the bigger instances, where integer programming had the worst performance. The computational evidence suggests the need for heuristic algorithms to solve the problem within acceptable computational times. We evaluated three algorithms with distinctive characteristics: greedy, genetic, and memetic. Our proposed memetic algorithm accomplishes this goal while producing near-optimal solutions.

The remainder of this paper is structured as follows. First, we discuss a novel integer programming formulation for the HFSBRP. Next, we present our memetic algorithm and its computational results. Finally, we offer conclusions and recommendations for future research.

Mathematical Formulation

The proposed mathematical model for the HFSBRP variant presents an objective function that corresponds to the global cost to be minimized, which can be divided into variable and fixed costs. The variable costs represent the total distance traveled by all buses selected in a given solution, taking into account both the capacity of each vehicle and the heterogeneity of the fleet. The number of buses used in a given solution reflects the fixed costs. In the following paragraphs, notation is introduced before the proposed model for the HFSBRP is introduced.

The sets are as follows:

- V = set of potential stops;
- V' = set of schools;
- S = set of students; and
- M = set of buses.

The parameters are as follows:

- c_{ij} = cost associated with arc (i, j);
- S_{il} = indicator variable that equals 1 if the lth student can walk to the ith stop, and 0 otherwise;
- C_k = capacity of the kth bus; and
- F_k = fixed cost for the kth bus.

The binary decision variables are as follows:

- $x_{ijk} = 1$ if the kth bus traverses the arc (i, j), and 0 otherwise;
- $y_{ik} = 1$ if the kth bus visits the ith stop, and 0 otherwise; and
- $z_{ilk} = 1$ if the lth student embarks at ith stop, and 0 otherwise.

The proposed variant can be formulated as follows:

[HFSBRP]

Minimize Z =

$$\sum_{i\in V}\sum_{j\in V}^{j\neq i} c_{\mathrm{ij}} \sum_{k\in M} x_{\mathrm{ij}}^k + \sum_{k\in M} F_k \sum_{j\in V'} x_{0j}^k \tag{1}$$

Subject to

$$\sum_{j\in V} x_{\mathrm{ij}}^k = \sum_{j\in V} x_{\mathrm{ji}}^k = y_{\mathrm{ik}} \quad \forall\ i\in V, \forall\ k\in M \tag{2}$$

$$\sum_{i,j\in Q}^{i\neq j} x_{\mathrm{ij}}^k \leq |Q| - 1 \quad \forall\ Q\subseteq V', \forall\ k\in M \tag{3}$$

$$\sum_{k\in M} z_{\mathrm{ilk}} \leq S_{\mathrm{il}} \quad \forall\ l\in S, \forall\ i\in V \tag{4}$$

$$\sum_{l\in S}\sum_{i\in V}^{S_{\mathrm{il}}=1} z_{\mathrm{ilk}} C_k \quad \forall\ k\in M \tag{5}$$

$$z_{\mathrm{ilk}} \leq y_{\mathrm{ik}} \quad \forall\ i\in V, \forall\ l\in S, \forall\ k\in M \tag{6}$$

$$\sum_{i\in V}\sum_{k\in M} z_{\mathrm{ilk}} = 1 \quad \forall\ l\in S \tag{7}$$

$$y_{\mathrm{ij}} \in \{0,1\} \quad \forall\ i,j\in V, i\neq j \tag{8}$$

$$x_{\mathrm{ij}}^k \in \{0,1\} \quad \forall\ i,j\in V, i\neq j, \forall\ k\in M \tag{9}$$

$$z_{\mathrm{ilk}} \in \{0,1\} \quad \forall\ i\in V, \forall\ l\in S, \forall\ k\in M \tag{10}$$

The objective function [Eq. (1)] minimizes the sum of variable costs (total distance traveled) and fixed costs (buses used). The set of constraints [Eq. (2)] imposes that, if stop i is visited by bus k, the arc (i, j) is traversed by bus k. The set of constraints in Eq. (3) imposes the route connectivity of the kth bus. This set of constraints also guarantees the elimination of subroutes. The set of constraints in Eq. (4) guarantees that each student embarks only at the selected stops. The set of constraints in Eq. (5) guarantees that the capacity of the kth bus is not exceeded. The set of constraints in Eq. (6) requires that student l does not embark at stop i and in bus k if bus k does not visit stop i. The set of constraints in Eq. (7) guarantees that each student is picked up at most once by a bus. Finally, the constraints in Eqs. (8–10) define the domain of the decision variables.

Proposed Memetic Algorithm

The HFSBRP can be divided into three subproblems: student allocation to stops, bus-routing generation and bus selection, considering a heterogeneous fleet of buses (different capacities). Based on the proposed mathematical formulation, preliminary computational experiments with the ILOG CPLEX solver (version 12.6.1) showed the necessity of heuristic algorithms for the solution of the problem. Those experiments are described in a later section.

Evolutionary algorithms have been used with broad success in numerous domains of combinatorial optimization. Several approaches have been reported in the literature, highlighting that the hybridization of such algorithms with specific heuristics can be beneficial for the solution of a given optimization problem. A form of hybridization of an evolutionary algorithm, such as a genetic algorithm (GA), is the inclusion of local search procedures. These algorithms are referred to as memetic algorithms (MAs) (Moscato 1989; Moscato and Cotta 2003, 2010).

For the resolution of the VRP variants, memetic algorithms were presented by Prins et al. (2006), Fallahi et al. (2008), Prins (2009), Mendoza et al. (2010), Nagata et al. (2010), Ngueveu et al. (2010), Ke and Feng (2013), Cattaruzza et al. (2014), Karaoglan and Altiparmak (2015), Matei et al. (2015), Qi et al. (2015), and Wang et al. (2015). In addition to the wide use of memetic algorithms in routing problems, we justify their use in the SBRP based on the building blocks hypothesis (Goldberg 1989), which appears to be very promising for the SBRP because this problem can benefit from obtaining better school bus routes based on routes suggested earlier.

The proposed genetic and memetic algorithm for the new variant under study was based on the memetic algorithm proposed by Lima et al. (2004) for the heterogeneous fleet vehicle routing problem. Next, we describe our proposed MA for the HFSBRP.

Student Allocation

Each student is allocated to the stop that is nearest to the school and that does not exceed the maximum walking distance. Each student has at least one available stop; therefore, each student is allocated to a stop. After the allocation of all students, the number of students allocated to each stop is calculated and a list of stops with at least one student is created.

Chromosome Encoding

The chromosome is encoded as a sequence of all stops with at least one student [e.g., for the set of stops with students (1, 2, 4, 5, 6), a valid chromosome would be (6, 2, 4, 5, 1)], as illustrated in Fig. 1.

Initial Population Generation

Adapting the Lima et al. (2004) procedure to the HFSBRP, in the construction phase of a given solution (chromosome), the first stop is randomly selected from the list of stops with at least one student. Therefore, the ith stop is randomly selected from a restricted list containing the three closest stops to the $(i-1)$th stop that receives students. If there are ties during selection, the first stop in the list is selected. From this point on, the steps consist of selecting buses and generating a complete route.

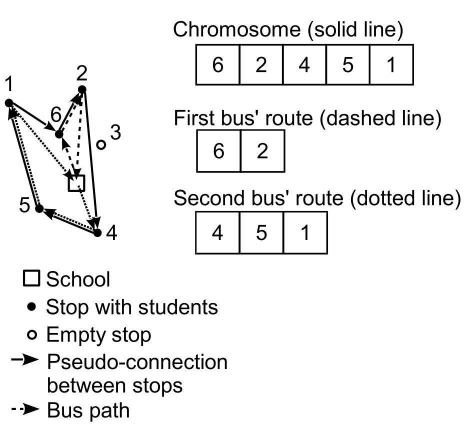

Fig. 1. Bus-routing example.

Bus Selection

Bus selection is performed by the function *define_bus* in the order in which the stops appear in the chromosome. Let us consider q buses with capacities $C_1, C_2, C_3, \ldots, C_q$ and fixed costs $F_1, F_2, F_3, \ldots, F_q$, respectively. The bus with the lower value of $(C_k - D_k) \times F_k$ is selected, where D_k is the number of students picked up by the kth bus (Ochi et al. 1998). The procedure keeps selecting buses until there are no more students at the stop, and the procedure ends when there are no more students left to be picked up. An example solution of the bus selection procedure is shown in Fig. 1. The first bus attends Stops 4, 5, and 1 (in this order), and the second bus attends Stops 6 and 2 (in this order).

The fitness of each chromosome is calculated in *define_fitness*, which sums the cost of the buses used and the cost of each route.

Parent Selection

Similar to Lima et al. (2004), two parents are selected through the binary tournament method, in which two distinct pools are created, with two chromosomes in each pool, and the parent with the higher fitness is selected for the crossover phase (Beasley and Chu 1996).

Crossover

The edge recombination operator (ERX) is applied to the best parents selected in the binary tournament, aiming generation of the stop scheduling toward the generated chromosome (Whitley et al. 1991). In the ERX crossover, a list is created containing both selected chromosomes, indicating which stops are connected, and the stops of the offspring are scheduled considering this list.

The first stop is randomly selected from the parents' stops. Subsequently, the next stop of the generated offspring is selected considering the list of stops connected with the current stop. The next stop is the one having the lowest number of connections with other stops. If two or more stops have the same minimum number of possible connections, one of those stops is randomly selected.

In case the current stop does not have more connections that can be used, the next stop is randomly selected from the stops that are not connected. The consequence of this random selection is that a limited amount of mutation is likely to occur. After all stops of both parents are connected sequentially, the crossover is finished and an offspring is generated. Then the function *define_bus* is applied to define the buses and routes for the generated offspring. A crossover example for six stops is shown in Fig. 2.

We did not include a mutation operator in our proposed MA.

Local Search

The adopted local search procedure is the lambda Interchange algorithm (Wassan and Osman 2002). It is run until a better chromosome is found or until a maximum number of attempts, which is a parameter of the proposed MA, is reached. A route is defined as a sequence of stops visited by the kth bus. Let the coding of a given solution be expressed by a set of routes $\{R_1, R_2, R_3, R_{k,} \ldots, R_l\}$, in which l is the number of buses used.

Let (R_p, R_q) be a pair that belongs to the solution. This operator swaps a subset S_p of size $|S_p| \leq \lambda$ from R_p with another subset S_q of size $|S_q| \leq \lambda$ from R_q. Thus, two new routes—$(R_p - S_p) \cup S_q$ and $(R_q - S_q) \cup S_p$—are generated. The neighborhood size is

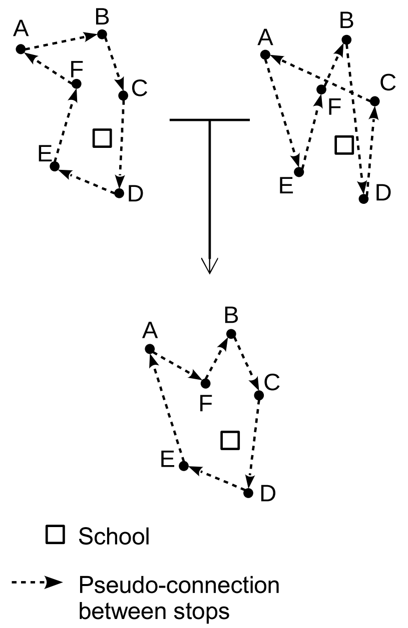

Fig. 2. ERX crossover example.

randomly selected from the interval $[0, \lambda]$, where λ = integer parameter.

The elements from S_p and S_q are randomly selected and placed at the end of the route that receives the set. If one of the sets S_p or S_q is empty, a transfer is executed. If both sets are empty, an exchange is performed between the two routes. After this operation, the number of students in each selected bus is calculated, and if a bus has its capacity constraint violated, this solution is discarded. If the new route configuration is feasible and a better chromosome is generated, it replaces the incumbent solution. This is the mechanism by which neighboring solutions are explored. Fig. 3 shows a lambda interchange example with $n = 7$ stops, $p = 1$, $q = 2$, $|S_p| = 1$, $|S_q| = 2$.

Population Replacement

If the offspring has a higher fitness (i.e., a better value in the objective function) than one of the parents, it replaces the worst parent. Elitism is adopted; that is, the better solution in the current generation is maintained in the next generation.

To escape local optima, a mechanism for the acceptance of worst solutions was designed. There is a probability p for the acceptance of an offspring that is worse than one of the parents, in which p is a parameter of the proposed MA.

Restart

According to the authors' experience, a problem that might happen with evolutionary algorithms is the premature convergence of the population. Aiming to reduce this loss of diversity, a procedure for population restart was designed. Every time that r generations are completed without improvement, the best solution in the current population is stored and all the other solutions are replaced by new chromosomes, generated as previously described. This procedure plays a key role in the exploration of several regions in the search space.

In Algorithm 1, the pseudocode version of the MA developed for the HFSBRP is presented.

Algorithm 1. Memetic Algorithm for the HFSBRP

```
FOR STUDENT I TO N DO:
  ALLOCATE_STUDENT(I);
FOR CHROMOSOME I TO MAX_GEN DO:
  GENERATE_CROMOSOME(I);
  DEFINE_BUS(I);
  DEFINE_FITNESS(I);
FOR GENERATION I TO K DO:
  BINARY_TOURNAMENT(P1, P2, P3, P4);
  ERX_CROSSOVER(P1, P2);
  DEFINE_BUS(OFFSPRING);
  FITNESS_OFFSPRING:=DEFINE_FITNESS(OFFSPRING);
  FOR GENERATION I TO P DO:
    LAMBDA_INTERCHANGE(OFFSPRING);
    FITNESS_OFFSPRING_LAMBDA:=DEFINE_FITNESS
    (OFFSPRING);
    IF FITNESS_OFFSPRING > FITNESS_OFFSPRING_LAMBDA
      FITNESS_OFFSPRING:=FITNESS_OFFSPRING_LAMBDA;
      OFFSPRING:=OFFSPRING_LAMBDA;
      BREAK;
  IF FITNESS_OFFSPRING < FITNESS_WORST_PARENT
    OFFSPRING ENTERS IN POPULATION.
```

Computational Experiments

For evaluating the proposed HFSBRP mathematical formulation, computational experiments were conducted involving the developed exact methods (integer programming and its linear relaxation) and the heuristics (greedy, genetic, and memetic algorithms). Because there were no other methods known to the authors for solving this problem, we restricted our comparisons to the algorithms proposed in this paper.

Bearing in mind that no benchmark instances exist for the proposed variant, we randomly generated a set of instances using our

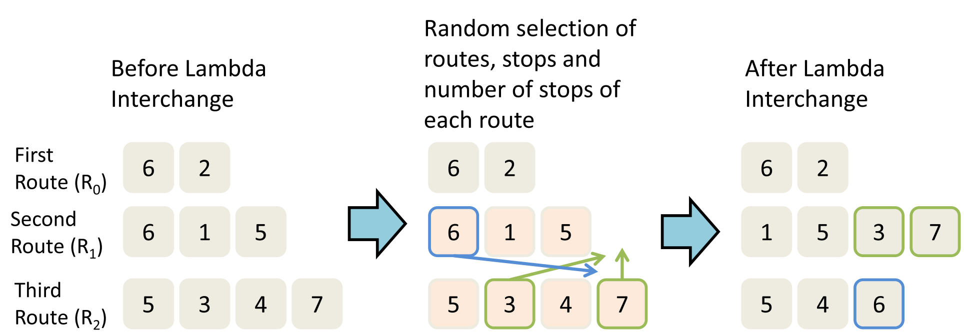

Fig. 3. Lambda interchange procedure.

own instance generator. The input parameters for the instance generator are the number of stops, the number of students at a stop, and the maximum distance that a student can walk.

The position of stops is chosen randomly along the predefined region, and the position of students is calculated by means of two random variables. The first variable is the distance between the student and the stop, which can be no greater than the maximum walking distance established, thereby guaranteeing that each student has at least one stop for walking to. The second variable is the direction of the student around the stop. This is the process used for generating stop and student positions reported by Schittekat et al. (2013).

Adapting the instance generation of Schittekat et al. (2013) for the HFSBRP, a bus fleet is added to the instance. From a finite number of bus varieties, buses of random types are added to the fleet until the fleet size is 50% bigger than needed, allowing the algorithm to make a nonbiased bus choice.

The integer programming mathematical formulation, presented in Eqs. (1)–(10), was implemented using the commercial software ILOG CPLEX, as previously noted. As seen in the original formulation, the set of constraints in Eq. (3) involves an exponential number of subroute elimination inequalities ($O(2^n)$). This set of constraints was replaced by Miller et al.'s (1960) constraints in the actual implementation, so

$$u_i + u_j + \mathrm{nx}_{\mathrm{ij}} \leq n - 1 \quad \forall\ i, j = 2, \ldots, ni \neq j \tag{11}$$

$$u_1 = 1 \tag{12}$$

Initially, we generated a set of 10 small instances (5–10 stops), aiming at validation of the proposed algorithm. Then the MA solved each instance 10 times. We also registered the optimal solution obtained by *CPLEX* in these instances, thus supporting the validity of the proposed metaheuristic. Given that this initial set of instances does not have practical significance (there are no real problems of this dimension), we opted for not reporting those experiments.

A set of 100 instances was generated according to authors' experience in observations of real problems. The size of each instance varied from 25 stops and 500 students to 250 stops and 5,250 students, with the maximum walking distance varying from 5 to 25 distance units. The maximum walking distance defines indirectly the amount of stops to which the student can go. The generated instances used here were presented in Sales et al. (2015).

To evaluate the proposed mathematical formulation and the complexity of the generated instances, we opted to define the lower bound obtained by linear relaxation (zLP) and the value of the optimal solution for the problem (zIP). Linear relaxation was suggested by Beasley (1993) as a technique for obtaining lower bounds in combinatorial optimization problems. The solution obtained in this way is usually infeasible for the whole problem. Nevertheless, it allows the establishment of a distance measure between the lower and upper bounds (in this case, the values obtained by heuristic techniques).

Next, we opted for relaxation of the integrality of Type-x_{ijk} decision variables, considering them continuous in the interval [0, 1]. Computational experiments realized with the *CPLEX* solver showed that, even for instances of small size, memory error made it impossible to obtain the lower bounds. For example, in Instance 1 (25 stops and 500 students), there was a memory error after two hours of processing the relaxed problem. Therefore, we concluded that adopting heuristic techniques to solve the HFSBRP was justified, given the need for computing high-quality solutions within acceptable computational times.

According to Barr et al. (1995), when no results from a more general method are available, such as linear or integer programming, a greedy algorithm can be used to find a reference solution. Hence, a greedy algorithm was developed, which initially selects the farthest stop from school that has at least one student and then always chooses the closest stop to the last visited stop that has at least one student. This procedure simulates how routes can be generated in practical situations by planners or even by bus drivers. In Algorithm 2, the pseudocode of the greedy algorithm developed for the HFSBRP is presented.

Algorithm 2. Greedy Algorithm for the HFSBRP

```
FOR STUDENT I TO N DO:
   ALLOCATE_STUDENT(I);
SELECT_FARTHEST_STOP(CHROMOSOME);
FOR STOP I TO N − 1 DO:
   SELECT_CLOSEST_STOP(LAST_STOP, CHROMOSOME);
DEFINE_BUS(CHROMOSOME);
DEFINE_FITNESS(CHROMOSOME);
```

Along with comparing the MA with the aforementioned greedy algorithm, the authors opted for comparing the proposed metaheuristic with a GA that consists of the MA without the local search procedure (lambda interchange). In this manner, we could evaluate how the local search affects both solution quality and computational times. Fig. 4 shows a solution returned by the proposed MA for Instance 12.

The parameters employed in the computational experiments were tuned and presented as follows:

- Population: 100 chromosomes;
- Maximum number of iterations: 5,000 generations;
- Maximum number of tries in the lambda interchange procedure: 50;
- λ parameter: 2;
- Probability of acceptance of a worse solution: 2%; and
- Number of generations without improvement for restart: 9,000 generations.

Each instance was solved 10 times with the MA, 10 times with the GA, and just once with the greedy algorithm. All tests were done on a computer with an Intel i7-3770S processor and 8GB RAM, with the algorithms implemented in C on Windows 8.1, using the compiler TDM-GCC, version 4.8.1 (32 bits).

Table 2 provides the results from the computational experiments. The first column shows the identification of the instance (*ID*); the succeeding columns show the number of stops (*sto*), the number of students (*stu*), the maximum walking distance (*MWD*), and the fleet size available (*fs*). We then had the average result on the test run (GA_{average}) and the average computational time required (*GA CTR*). The MA results are presented in the same order. The results for the greedy algorithm and its computational time are then presented. The last three columns respectively show the relative gap between the results from the GA and the greedy algorithm (Δ_{GA}), the relative gap between the results from the MA and the greedy algorithm (Δ_{MA}), and the difference in the gaps ($\Delta_{MA} - \Delta_{GA}$).

The relative gaps Δ_{GA} and Δ_{MA} are calculated as the difference between the average result obtained with the algorithm (GA or MA) and the result obtained with the greedy algorithm, divided by the result obtained with the greedy algorithm. The last row of Table 2 shows the average value of the last three columns.

The results presented in Table 2 indicate that it is possible to determine which instance variables deeply influence the difficulty of the problem. The higher the maximum walking distance, maintaining the other parameters constant, the lower the sum of the bus routes. This is because, with the higher student mobility, the algorithm can concentrate students at stops closer to the school.

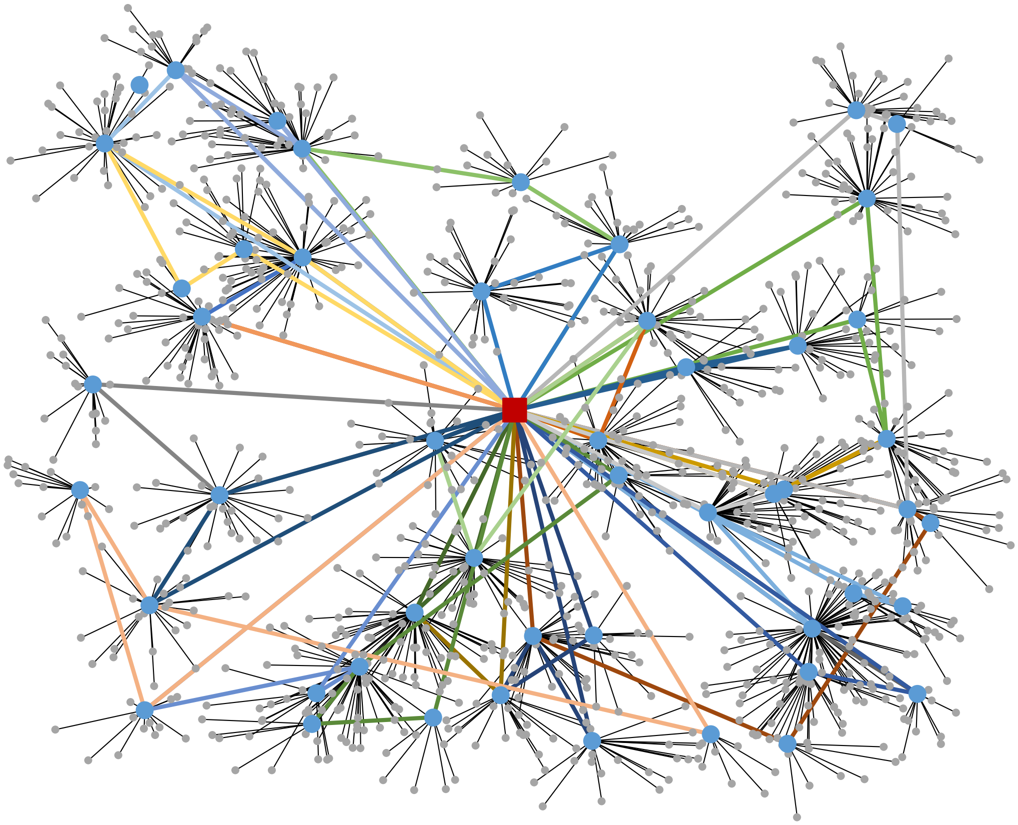

Fig. 4. Solution obtained for Instance 12. Square = school; large circles = bus stops; small circles = students; and lines = proposed route for each bus.

The higher the number of students, keeping other parameters constant, the larger the sum of the routes, because more buses and stops are required to satisfy demand.

We observe in Table 2 that the results obtained by the greedy algorithm are always worse than those obtained by the GA and the MA. Because the greedy algorithm is similar to procedures commonly used by planners or even bus drivers, this shows the necessity of more advanced heuristics such as the evolutionary algorithms presented in this paper.

The proposed MA can be interpreted as a sampling process in which the solution obtained is a random variable. In this context, evaluating the stability of the solutions obtained by the

Table 2. Computational results

Identifier	Stops	Students	MWD	fs	GA_{average}	GA CTR (s)	MA_{average}	MA CTR (s)	Greedy solution	Greedy CTR (s)	ΔGA (%)	ΔMA (%)	$\Delta MA - \Delta GA$ (%)
1	25	500	5	21	4,751	0.1	4,745	1.3	6,135	0.000	29.1	29.3	0.2
2			10	21	4,446	0.1	4,442	2.0	5,651	0.000	27.1	27.2	0.1
3			15	18	4,180	0.1	4,176	1.0	5,130	0.000	22.7	22.8	0.1
4			20	21	4,015	0.1	4,012	3.3	4,671	0.000	16.3	16.4	0.1
5			25	18	3,837	0.1	3,831	4.3	4,397	0.000	14.6	14.8	0.2
6		750	5	29	6,629	0.1	6,630	1.5	7,566	0.000	14.2	14.1	0.0
7			10	30	6,589	0.1	6,392	0.9	7,586	0.000	15.1	18.7	3.6
8			15	29	6,018	0.1	6,017	5.5	7,028	0.000	16.8	16.8	0.0
9			20	26	5,730	0.1	5,726	5.8	6,552	0.000	14.4	14.4	0.1
10			25	30	5,452	0.1	5,446	7.4	5,973	0.000	9.6	9.7	0.1
11	50	1,000	5	38	8,925	0.3	8,806	3.0	11,349	0.000	27.2	28.9	1.7
12			10	36	8,786	0.3	8,788	7.5	11,095	0.000	26.3	26.2	0.0
13			15	38	8,251	0.3	8,127	5.3	10,213	0.000	23.8	25.7	1.9
14			20	39	7,692	0.3	7,698	12.2	9,094	0.000	18.2	18.1	0.1
15			25	38	7,456	0.3	7,279	5.0	8,601	0.000	15.4	18.2	2.8
16		1,250	5	45	10,836	0.3	10,823	13.6	13,459	0.000	24.2	24.3	0.2
17			10	48	10,853	0.3	10,737	6.3	13,278	0.000	22.3	23.7	1.3
18			15	48	10,071	0.3	10,078	13.1	11,953	0.000	18.7	18.6	−0.1
19			20	48	9,573	0.3	9,563	15.9	11,098	0.015	15.9	16.0	0.1
20			25	47	9,082	0.3	8,925	12.2	10,081	0.015	11.0	13.0	2.0

Table 2. (*Continued.*)

Identifier	Stops	Students	MWD	fs	$GA_{average}$	GA CTR (s)	$MA_{average}$	MA CTR (s)	Greedy solution	Greedy CTR (s)	ΔGA (%)	ΔMA (%)	$\Delta MA - \Delta GA$ (%)
21	75	1,500	5	59	13,492	0.6	13,492	29.6	17,800	0.000	31.9	31.9	0.0
22			10	54	12,778	0.6	12,767	19.9	16,273	0.000	27.4	27.5	0.1
23			15	59	12,071	0.6	12,066	50.1	15,073	0.015	24.9	24.9	0.0
24			20	57	11,115	0.6	11,114	58.5	13,166	0.000	18.5	18.5	0.0
25			25	57	10,658	0.5	10,662	204.0	12,098	0.015	13.5	13.5	0.0
26		1,725	5	68	15,527	0.7	15,418	14.2	19,428	0.000	25.1	26.0	0.9
27			10	66	14,737	0.7	14,573	11.3	18,113	0.000	22.9	24.3	1.4
28			15	65	13,543	0.6	13,545	42.8	16,136	0.000	19.1	19.1	0.0
29			20	63	12,907	0.6	12,899	41.6	15,346	0.016	18.9	19.0	0.1
30			25	62	12,372	0.5	12,184	17.2	13,957	0.000	12.8	14.6	1.7
31	100	2,000	5	72	17,500	1.0	17,496	106.3	22,916	0.015	30.9	31.0	0.0
32			10	75	16,970	1.0	16,960	49.4	21,681	0.000	27.8	27.8	0.1
33			15	77	15,902	1.1	15,761	22.2	19,787	0.000	24.4	25.5	1.1
34			20	75	14,825	0.9	14,825	62.9	17,882	0.016	20.6	20.6	0.0
35			25	77	14,452	0.8	14,278	28.8	16,533	0.016	14.4	15.8	1.4
36		2,300	5	87	20,711	1.2	20,565	21.6	26,490	0.000	27.9	28.8	0.9
37			10	86	18,958	1.2	18,966	87.4	23,707	0.000	25.0	25.0	0.0
38			15	84	17,949	1.1	17,932	99.2	21,716	0.016	21.0	21.1	0.1
39			20	89	17,182	1.1	17,188	77.4	20,206	0.000	17.6	17.6	0.0
40			25	87	16,775	1.0	16,559	36.3	19,056	0.016	13.6	15.1	1.5
41	125	2,500	5	90	22,215	1.6	22,199	111.9	29,300	0.016	31.9	32.0	0.1
42			10	92	20,503	1.5	20,502	116.6	25,664	0.016	25.2	25.2	0.0
43			15	96	19,880	1.6	19,725	44.5	24,792	0.031	24.7	25.7	1.0
44			20	90	18,336	1.3	18,326	211.2	21,543	0.016	17.5	17.6	0.1
45			25	93	17,913	1.2	17,699	46.3	20,399	0.016	13.9	15.2	1.4
46		2,750	5	102	24,498	1.7	24,498	147.0	31,675	0.016	29.3	29.3	0.0
47			10	101	22,897	1.7	22,911	112.8	29,034	0.016	26.8	26.7	−0.1
48			15	105	21,816	1.7	21,643	50.3	26,860	0.016	23.1	24.1	1.0
49			20	105	20,532	1.6	20,544	278.2	24,167	0.016	17.7	17.6	−0.1
50			25	101	19,429	1.3	19,429	193.6	21,930	0.016	12.9	12.9	0.0
51	150	3,000	5	111	26,294	2.4	26,191	80.0	34,583	0.015	31.5	32.0	0.5
52			10	113	25,163	2.3	25,195	163.7	32,394	0.016	28.7	28.6	−0.2
53			15	110	23,516	2.2	23,364	68.4	29,424	0.016	25.1	25.9	0.8
54			20	113	22,064	2.0	22,056	238.5	26,253	0.016	19.0	19.0	0.0
55			25	110	21,301	1.7	21,293	259.1	23,983	0.015	12.6	12.6	0.0
56		3,300	5	122	28,742	2.6	28,763	199.3	37,372	0.016	30.0	29.9	−0.1
57			10	125	27,295	2.5	27,293	158.3	34,151	0.015	25.1	25.1	0.0
58			15	122	25,775	2.4	25,614	79.0	31,479	0.016	22.1	22.9	0.8
59			20	128	24,398	2.2	24,407	255.8	29,032	0.016	19.0	19.0	0.0
60			25	122	23,247	2.0	23,244	574.8	26,415	0.016	13.6	13.6	0.0
61	175	3,500	5	132	30,596	3.2	30,482	87.5	39,528	0.016	29.2	29.7	0.5
62			10	132	28,361	3.3	28,368	266.3	36,430	0.016	28.5	28.4	0.0
63			15	129	26,745	3.0	26,737	377.4	33,133	0.015	23.9	23.9	0.0
64			20	132	25,810	2.8	25,812	403.3	30,729	0.016	19.1	19.0	0.0
65			25	128	24,868	2.5	24,674	133.1	28,391	0.015	14.2	15.1	0.9
66		3,675	5	143	32,201	3.4	32,212	342.5	42,180	0.015	31.0	30.9	0.0
67			10	138	30,205	3.5	30,103	94.0	38,936	0.016	28.9	29.3	0.4
68			15	137	28,316	3.1	28,326	250.5	34,783	0.016	22.8	22.8	0.0
69			20	137	26,880	3.0	26,892	423.3	32,041	0.031	19.2	19.1	−0.1
70			25	140	25,835	2.6	25,840	497.0	29,378	0.031	13.7	13.7	0.0
71	200	4,000	5	149	35,183	4.5	35,201	300.3	47,431	0.031	34.8	34.7	−0.1
72			10	150	33,158	4.4	33,024	180.3	42,818	0.016	29.1	29.7	0.5
73			15	158	31,075	4.1	31,072	390.2	38,298	0.015	23.2	23.3	0.0
74			20	146	29,033	3.8	29,030	578.9	34,773	0.031	19.8	19.8	0.0
75			25	149	28,228	3.0	28,218	942.5	31,981	0.031	13.3	13.3	0.0
76		4,200	5	156	36,333	4.6	36,358	379.7	47,510	0.031	30.8	30.7	−0.1
77			10	158	34,733	4.4	34,607	160.6	44,023	0.031	26.7	27.2	0.46
78			15	156	32,243	4.1	32,220	379.4	40,000	0.016	24.1	24.1	0.09
79			20	158	30,839	3.7	30,831	377.6	36,457	0.031	18.2	18.2	0.03
80			25	155	29,557	3.4	29,538	898.1	33,780	0.031	14.3	14.4	0.07

J. Urban Plann. Dev., 2018, 144(2): 04018018

Table 2. (*Continued*.)

Identifier	Stops	Students	MWD	fs	GA_{average}	GA CTR (s)	MA_{average}	MA CTR (s)	Greedy solution	Greedy CTR (s)	ΔGA (%)	ΔMA (%)	$\Delta MA - \Delta GA$ (%)
81	225	4,500	5	165	39,111	5.9	39,110	602.6	52,625	0.015	34.6	34.6	0.00
82			10	176	36,521	5.9	36,542	548.3	46,785	0.015	28.1	28.0	−0.07
83			15	167	34,756	5.4	34,771	517.0	44,105	0.031	26.9	26.8	−0.06
84			20	168	32,830	4.9	32,735	314.3	39,330	0.047	19.8	20.1	0.35
85			25	171	31,799	4.4	31,810	903.8	36,403	0.047	14.5	14.4	−0.04
86		4,725	5	177	41,110	6.1	41,065	606.3	54,443	0.031	32.4	32.6	0.14
87			10	179	38,899	6.0	38,897	500.4	50,128	0.047	28.9	28.9	0.01
88			15	180	36,546	5.2	36,466	269.0	45,123	0.031	23.5	23.7	0.27
89			20	183	34,364	4.9	34,391	661.1	40,602	0.046	18.2	18.1	−0.09
90			25	174	33,147	4.3	33,148	856.4	37,443	0.062	13.0	13.0	0.00
91	250	5,000	5	188	43,300	7.4	43,289	583.5	57,693	0.031	33.2	33.3	0.04
92			10	191	40,829	7.3	40,833	555.2	52,484	0.046	28.5	28.5	−0.01
93			15	192	38,009	6.7	37,990	635.6	46,871	0.046	23.3	23.4	0.06
94			20	191	36,542	5.9	36,543	702.8	43,388	0.031	18.7	18.7	0.00
95			25	189	34,834	5.2	34,850	1451.0	39,276	0.047	12.8	12.7	−0.05
96		5,250	5	201	46,362	7.5	46,289	254.6	60,714	0.047	31.0	31.2	0.21
97			10	197	42,670	7.4	42,676	597.9	55,129	0.031	29.2	29.2	−0.02
98			15	197	40,095	6.7	40,083	657.5	49,440	0.031	23.3	23.3	0.04
99			20	198	38,162	6.5	38,037	307.6	45,522	0.046	19.3	19.7	0.39
100			25	191	36,873	5.5	36,751	549.6	41,738	0.046	13.2	13.6	0.37
Average											22.1	22.4	0.3

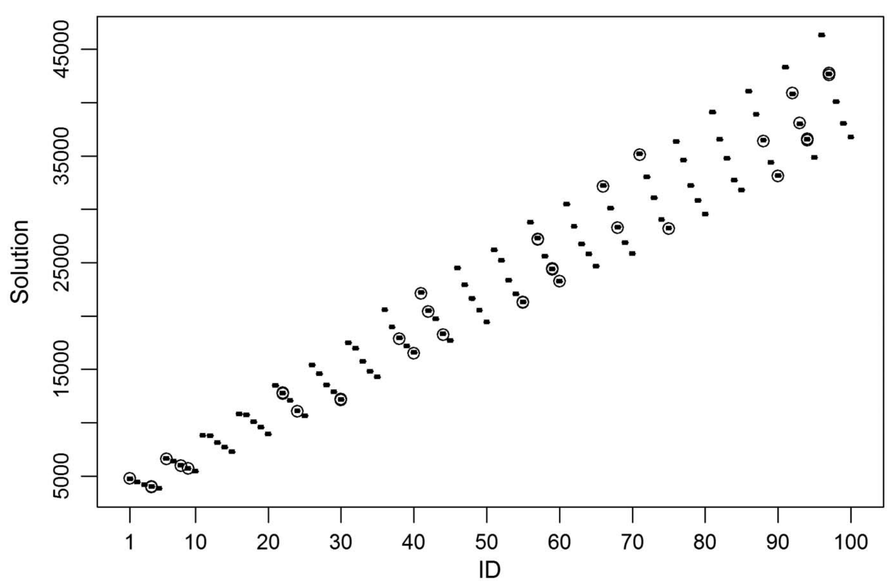

Fig. 5. MA boxplots generated for every instance. Circles = outliers.

metaheuristics is of paramount importance in applying the algorithms to real-life problems.

To study the behavior of the MA, boxplots were generated for the solutions obtained in each instance. Each boxplot is based on the solution samples obtained in the 10 executions of the algorithm for each instance. Fig. 5 shows the boxplots for all 100 instances.

Based on the results, we concluded that the MA exhibits high consistency in its solutions, with a low standard deviation and a small number of outliers (only 34 in the 1,000 tests carried out). Even on the bigger instances, where we might have concluded that the variability in MA results would be greater, the number of outliers did not show any meaningful difference.

Regarding the quality of the MA results, we can emphasize the following observations. Table 3 shows a pattern between the gaps obtained for the memetic and genetic algorithm compared with the greedy algorithm for different maximum walking distances. With the increase in maximum walking distance from 5 to 25 distance units, the gaps between the memetic and greedy algorithms decrease from an average of 29.8% to an average of 13.9%.

Table 3. Average GA and MA gaps compared with greedy algorithm gaps for MWD categories

MWD	GA (%)	MA (%)
5	29.5	29.8
10	26.4	26.8
15	22.9	23.2
20	18.3	18.3
25	13.3	13.9

J. Urban Plann. Dev., 2018, 144(2): 04018018

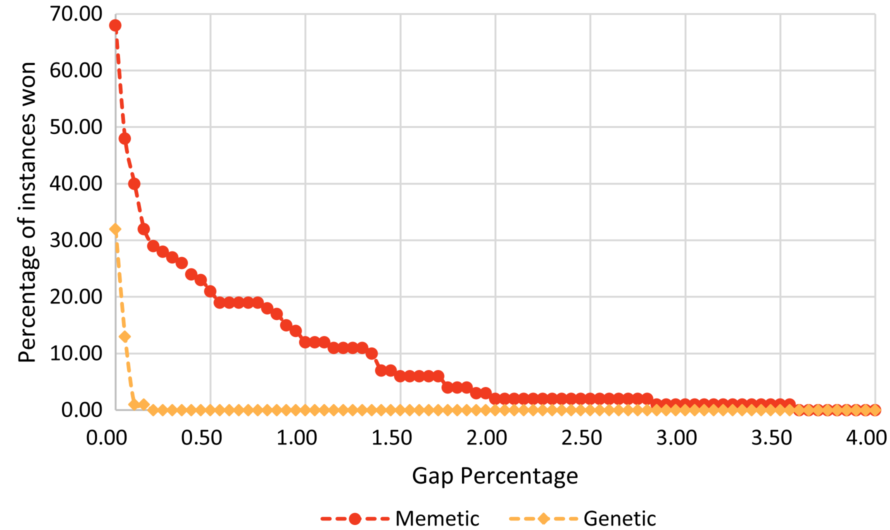

Fig. 6. Percentage of instances versus percentage of gaps won by memetic/genetic algorithm.

This downward trend also occurs with the GA. It happens because the maximum walking distance increases the inherent complexity of all subproblems by increasing the possibilities for student allocation, routing, and bus allocation, but decreases solution quality. We observe in Table 3 that the MA compares favorably with the GA and the genetic and greedy algorithms because it obtains greater or equal average gaps in every maximum walking distance (MWD) category shown.

An analysis of the gaps obtained by the GA and the MA is shown in Fig. 6. On the *y*-axis, the percentage of instances in which the genetic/memetic algorithm won over the memetic/genetic algorithm presents a gap greater than the value of the *x*-axis. For example, the MA won 12.00% of the instances over the GA, with a gap greater than 1.00%. The GA does not show a gap greater than 0.15% in relation to the MA for any tested instances. Besides, the GA curve plummets near the origin point, indicating that the instances that GA won were with a very thin gap. It was expected that the MA would produce better solutions than the GA because it incorporates the GA plus the lambda interchange local search procedure. As seen in Fig. 6, this neighborhood search incurs in an increase of the solution quality at the expense of an increase of the computational time required.

In Fig. 7, the average computational times required (on logarithmic scale) are shown on the *y*-axis. Identification of the instance is shown on the *x*-axis. As expected, computational time increases with increased instance size, which is reflected in the growing number of stops and students. After Instance 81 (225 stops and 4,500 students), there is a decrease in the influence of instance size versus computational time required.

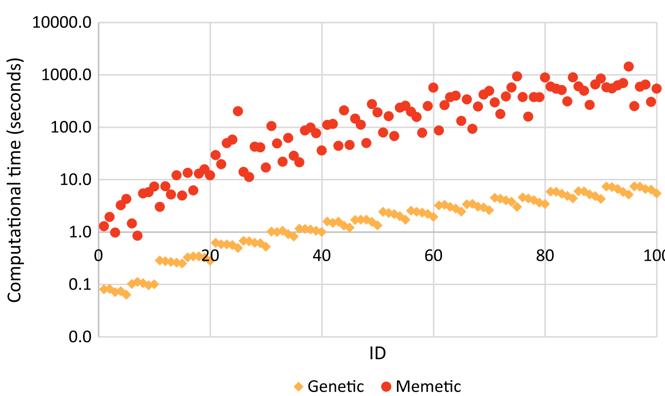

Fig. 7. Mean computational time for all instances executed.

In a clear way, maximum walking distance influences the computational time required by the GA and the MA, because, with the increase in MWD, the algorithms tend to use fewer stops, lowering computational effort when routing. In the MA, we do not observe as well-defined a pattern of computational time required as that in the GA because the MA can explore the neighborhood deeply (or not). The lambda interchange procedure, after the parent's crossover, has 50 tries for exploring the neighborhood and finding a better solution than the current one; it stops exploring when it finds one.

Based on the results, it is possible to determine that the GA tends to be quicker than the MA. There is a trade-off between solution quality and computational time required, as is typically observed in experiments with similar algorithms. Although the GA's computational time was lower in all instances, the MA was capable of obtaining solutions for all instances in acceptable computational times.

Discussion and Conclusions

An optimized management of school transport systems can produce appreciable gains in terms of cost reduction along with improved quality of the service provided. In this paper we presented different algorithms for solving the integrated subproblems of bus stop selection, student allocation, and bus routing, considering a heterogeneous fleet.

In experiments conducted with small instances (up to 10 stops), the exact model solved to optimality with the *CPLEX* solver in acceptable computational time. However, in larger-sized instances (more than 25 stops), it was not possible to find optimal solutions to the exact model and lower bounds originated from linear relaxation. A memetic algorithm was proposed for the HFSBRP, which was validated in experiments with small-sized instances (up to 10 stops), returning the optimal solution for every instance.

Nevertheless, these algorithms do not deal with other subproblems of the HFSBRP, such as time windows, roadway capacities during peak periods, toll road use, or the requirement to balance riding times for both drivers and students. These subproblems are time-dependent and can deeply affect the solutions obtained.

The memetic algorithm exhibited consistent behavior, outperforming, in terms of quality, the genetic algorithm and a greedy algorithm developed for the HFSBRP. However, the computational time required for the MA is multiple times larger than that required for the genetic and greedy algorithms. We suggest the study of other local search techniques that may give good solutions with less computational effort. Also, because these algorithms were tested only in a randomly generated urban scenario, there is no assurance of relevant solutions in other scenarios, such as widely spaced rural bus stops and limitations in the road network.

As a natural development of the research reported in this article, the authors are working on the integration the proposed variant with the student nucleation problem (Rocha Kloeckner 2015), the incorporation of time windows constraints, and the solution of real-world HFSBRP instances.

Acknowledgments

The support of the Ceará State Foundation for the Support of Scientific and Technological Development (FUNCAP) is acknowledged and appreciated. The funding source had no involvement with the research.

References

Barr, R. S., B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart Jr. 1995. "Designing and reporting on computational experiments with heuristic methods." *J. Heuristics* 1 (1): 9–32. https://doi.org/10.1007/BF02430363.

Beasley, J. E. 1993. *Modern heuristic techniques for combinatorial problems*. Edited by C. R. Reeves. Coventry, UK: Wiley.

Beasley, J. E., and P. C. Chu. 1996. "A genetic algorithm for the set covering problem." *Eur. J. Oper. Res.* 94 (2): 392–404. https://doi.org/10.1016/0377-2217(95)00159-X.

Bennett, B. T., and D. C. Gazis. 1972. "School bus routing by computer." *Transp. Res.* 6 (4): 317–325. https://doi.org/10.1016/0041-1647(72)90072-X.

Bowerman, R., B. Hall, and P. Calamai. 1995. "A multi-objective optimization approach to urban school bus routing: Formulation and solution method." *Transp. Res. Part A* 29 (2): 107–123. https://doi.org/10.1016/0965-8564(94)E0006-U.

Caceres, H., R. Batta, and Q. He. 2017. "School bus routing with stochastic demand and duration constraints." *Transp. Sci.* 51 (4): 1349–1364. https://doi.org/10.1287/trsc.2016.0721.

Cattaruzza, D., N. Absi, D. Feillet, and T. Vidal. 2014. "A memetic algorithm for the multi trip vehicle routing problem." *Eur. J. Oper. Res.* 236 (3): 833–848. https://doi.org/10.1016/j.ejor.2013.06.012.

Chapleau, L., J. A. Ferland, and J. M. Rousseau. 1985. "Clustering for routing in densely populated areas." *Eur. J. Oper. Res.* 20 (1): 48–57. https://doi.org/10.1016/0377-2217(85)90283-8.

Clarke, G., and J. W. Wright. 1964. "Scheduling of vehicles from a central depot to a number of delivery points." *Oper. Res.* 12 (4): 568–581. https://doi.org/10.1287/opre.12.4.568.

Dulac, G., J. A. Ferland, and P. A. Forgues. 1980. "School bus routes generator in urban surroundings." *Comput. Oper. Res.* 7 (3): 199–213. https://doi.org/10.1016/0305-0548(80)90006-4.

Fallahi, A. E., C. Prins, and R. Wolfler Calvo. 2008. "A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem." *Comput. Oper. Res.* 35 (5): 1725–1741. https://doi.org/10.1016/j.cor.2006.10.006.

Feng, S., B. Hu, C. Nie, X. Shen, and Y. Ci. 2016. "Game-based competition models between bus routes." *J. Urban Plan. Dev.* 142 (3): 04015022. https://doi.org/10.1061/(ASCE)UP.1943-5444.0000313.

Fügenschuh, A. 2009. "Solving a school bus scheduling problem with integer programming." *Eur. J. Oper. Res.* 193 (3): 867–884. https://doi.org/10.1016/j.ejor.2007.10.055.

Gavish, B., and E. Shlifer. 1979. "An approach for solving a class of transportation scheduling problems." *Eur. J. Oper. Res.* 3 (2): 122–134. https://doi.org/10.1016/0377-2217(79)90098-5.

Gillett, B. E., and L. R. Miller. 1974. "A heuristic algorithm for the vehicle-dispatch problem." *Oper. Res.* 22 (2): 340–349. https://doi.org/10.1287/opre.22.2.340.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley Professional.

Karaoglan, I., and F. Altiparmak. 2015. "A memetic algorithm for the capacitated location-routing problem with mixed backhauls." *Comput. Oper. Res.* 55: 200–216. https://doi.org/10.1016/j.cor.2014.06.009.

Ke, L., and Z. Feng. 2013. "A two-phase metaheuristic for the cumulative capacitated vehicle routing problem." *Comput. Oper. Res.* 40 (2): 633–638. https://doi.org/10.1016/j.cor.2012.08.020.

Khakbaz, A., A. S. Nookabadi, and N. S. Bushehri. 2016. "Two-phase approach for designing bus-based park-and-ride system: Case study of Isfahan, Iran." *J. Urban Plann. Dev.* 143 (1): 04016027. https://doi.org/10.1061/(ASCE)UP.1943-5444.0000359.

Li, L. Y. O., and Z. Fu. 2002. "The school bus routing problem: A case study." *J. Oper. Res. Soc.* 53 (5): 552–558. https://doi.org/10.1057/palgrave.jors.2601341.

Lima, C. M. R. R., M. C. Goldbarg, and E. F. G. Goldbarg. 2004. "A memetic algorithm for the heterogeneous fleet vehicle routing problem." *Electron. Notes Discret. Math.* 18: 171–176. https://doi.org/10.1016/j.endm.2004.06.027.

Matei, O., P. C. Pop, J. L. Sas, and C. Chira. 2015. "An improved immigration memetic algorithm for solving the heterogeneous fixed fleet vehicle routing problem." *Neurocomputing* 150: 58–66. https://doi.org/10.1016/j.neucom.2014.02.074.

Mendoza, J. E., B. Castanier,C. Guéret, A. L. Medaglia, and N. Velasco. 2010. "A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands." *Comput. Oper. Res.* 37 (11): 1886–1898. https://doi.org/10.1016/j.cor.2009.06.015.

Miller, C. E., R. A. Zemlin, and A. W. Tucker. 1960. "Integer programming formulation of traveling salesman problems." *J. ACM* 7 (4): 326–329. https://doi.org/10.1145/321043.321046.

Moscato, P. 1989. *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*. Caltech Concurrent Computation Program 158-79. Pasadena, CA: California Institute of Technology.

Moscato, P., and C. Cotta. 2003. *Handbook of metaheuristics*. Edited by F. Glover and G. A. Kochenberger. New York: Kluwer Academic Publishers.

Moscato, P., and C. Cotta. 2010. *Handbook of metaheuristics*. Edited by M. Gendreau and J. Y. Potvin. New York: Springer.

Nagata, Y., O. Bräysy, and W. Dullaert. 2010. "A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows." *Comput. Oper. Res.* 37 (4): 724–737. https://doi.org/10.1016/j.cor.2009.06.022.

Newton, R. M., and W. H. Thomas. 1969. "Design of school bus routes by computer." *Socioecon. Plann. Sci.* 3 (1): 75–85. https://doi.org/10.1016/0038-0121(69)90051-2.

Newton, R. M., and W. H. Thomas. 1974. "Bus routing in a multi-school system." *Comput. Oper. Res.* 1 (2): 213–222. https://doi.org/10.1016/0305-0548(74)90047-1.

Ngueveu, S. U., C. Prins, and R. Wolfler Calvo. 2010. "An effective memetic algorithm for the cumulative capacitated vehicle routing problem." *Comput. Oper. Res.* 37 (11): 1877–1885. https://doi.org/10.1016/j.cor.2009.06.014.

Ochi, L. S., D. S. Vianna, L. M. D. A. Drummond, and A. O. Victor. 1998. "A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet." *Future Gener. Comput. Syst.* 14 (5–6): 285–292. https://doi.org/10.1016/S0167-739X(98)00034-X.

Pan, S., J. Yu, X. Yang, Y. Liu, and N. Zou. 2015. "Designing a flexible feeder transit system serving irregularly shaped and gated communities: Determining service area and feeder route planning." *J. Urban Plann.*

Dev. 141 (3): 04014028. https://doi.org/10.1061/(ASCE)UP.1943-5444.0000224.

Park, J., and B.-I. Kim. 2010. "The school bus routing problem: A review." *Eur. J. Oper. Res.* 202 (2): 311–319. https://doi.org/10.1016/j.ejor.2009.05.017.

Park, J., H. Tae, and B.-I. Kim. 2012. "A post-improvement procedure for the mixed load school bus routing problem." *Eur. J. Oper. Res.* 217 (1): 204–213. https://doi.org/10.1016/j.ejor.2011.08.022.

Prins, C. 2009. "Two memetic algorithms for heterogeneous fleet vehicle routing problems." *Eng. Appl. Artif. Intell.* 22 (6): 916–928. https://doi.org/10.1016/j.engappai.2008.10.006.

Prins, C., C. Prodhon, and R. W. Calvo. 2006. *Evolutionary computation in combinatorial optimization*. *Lecture notes in computer science*. Edited by J. Gottlieb and G. R. Raidl. Berlin: Springer.

Qi, Y., Z. Hou, H. Li, J. Huang, and X. Li. 2015. "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows." *Comput. Oper. Res.* 62: 61–77. https://doi.org/10.1016/j.cor.2015.04.009.

Riera-Ledesma, J., and J. J. Salazar-González. 2013. "A column generation approach for a school bus routing problem with resource constraints." *Comput. Oper. Res.* 40 (2): 566–583. https://doi.org/10.1016/j.cor.2012.08.011.

Riera-Ledesma, J., and J.-J. Salazar-González. 2012. "Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach." *Comput. Oper. Res.* 39 (2): 391–404. https://doi.org/10.1016/j.cor.2011.04.015.

Rocha Kloeckner, N. V. R. 2015. *The student nucleation problem*. [In Portuguese.] Fortaleza, Brazil: Federal Univ. of Ceará.

Rosenkrantz, D., R. Stearns, and P. Lewis. 1974. "Approximate algorithms for the travelling salesperson problem." In *Proc., 15th Annual IEEE Symp. Switching and Automata Theory*, 33–42. Piscataway, NJ: IEEE.

Sales, L. P. A., C. S. Melo, T. O. Bonates, and B. A. Prata. 2015. "Heterogeneus fleet school bus routing problem instances." Accessed September 26, 2015. https://www.researchgate.net/publication/282184239_Heterogeneus_Fleet_School_Bus_Routing_Problem_Instances.

Schittekat, P., J. Kinable, K. Sorensen, M. Sevaux, F. Spieksma, and J. Springael. 2013. "A metaheuristic for the school bus routing problem with bus stop selection." *Eur. J. Oper. Res.* 229 (2): 518–528. https://doi.org/10.1016/j.ejor.2013.02.025.

Souza Lima, F. M., D. S. Pereira, S. V. Conceição, and N. T. Ramos Nunes. 2016. "A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context." *Expert Syst. Appl.* 56: 320–334. https://doi.org/10.1016/j.eswa.2016.03.005.

Verderber, W. J. 1974. "Automated pupil transportation." *Comput. Oper. Res.* 1 (2): 235–245. https://doi.org/10.1016/0305-0548(74)90049-5.

Wang, Z., H. Jin, and M. Tian. 2015. "Rank-based memetic algorithm for capacitated arc routing problems." *Appl. Soft Comput. J.* 37: 572–584. https://doi.org/10.1016/j.asoc.2015.08.003.

Wassan, N. A., and I. H. Osman. 2002. "Tabu search variants for the mix fleet vehicle routing problem." *J. Oper. Res. Soc.* 53 (7): 768–782. https://doi.org/10.1057/palgrave.jors.2601344.

Whitley, D., T. Starkweather, and D. Shaner. 1991. *Handbook of genetic algorithms*. Edited by L. Davis. New York: Van Nostrand Reinhold.

Yellow, P. C. 1970. "A computational modification to the savings method of vehicle scheduling." *Oper. Res. Q.* 21 (2): 281–283. https://doi.org/10.1057/jors.1970.52.