4OR-Q J Oper Res (2017) 15:359–386
DOI 10.1007/s10288-017-0340-8

RESEARCH PAPER

A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads

Fátima M. de Souza Lima[1] · Davi S. D. Pereira[1] · Samuel V. da Conceição[1] · Ricardo S. de Camargo[1]

Received: 18 March 2016 / Revised: 19 October 2016 / Published online: 24 January 2017

Abstract Four multi-objective meta-heuristic algorithms are presented to solve a multi-objective capacitated rural school bus routing problem with a heterogeneous fleet and mixed loads. Three objectives are considered: the total weighted traveling time of the students, the balance of routes among drivers, and the routing costs. The proposed methods were compared with one from the literature, and their performance assessed observing three multi-objective metrics: cardinality, coverage, and hyper-volume. All four devised methods outperformed the one from the literature. The algorithm with a path relinking procedure embedded during the crowding distance selection scheme had the best overall performance.

Keywords Capacitated rural school bus routing problem · Mixed loads · Multi-objective optimization · Multi-objective meta-heuristics · Developing countries · Decision support systems

Mathematics Subject Classification 90B06 · 90B50 · 90-08

✉ Fátima M. de Souza Lima
fatimamsouzalima@gmail.com

Davi S. D. Pereira
ni.doro@gmail.com

Samuel V. da Conceição
sviera@dep.ufmg.br

Ricardo S. de Camargo
rcamargo@dep.ufmg.br

1 Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Antonio Carlos, 6627, Belo Horizonte, MG 31270-901, Brazil

1 Introduction

Brazil has close to fifty million elementary students enrolled in its public educational system, being 13% of them located in rural areas (INEP 2013) and served by multi-grade classes. In a multi-grade class, the same teacher is responsible for pupils of different grades arranged in the same room. Tough debatable, these sort of arrangements are often considered to be unsuitable for today's advanced curricula. They usually have fewer resources than necessary, besides having difficulty to attract and retain qualified, skilled teachers given their remote location, far from larger centers (Vincent 1999).

To offer a more suitable environment, with better school infrastructures, a richer curriculum, and single-grade classrooms for students, the Brazilian federal government has been nucleating rural schools into sites near the towns' downtown, but transferring at the same time the responsibility for the students' transportation to local authorities. Municipalities have now to provide and manage a transportation system which has to pick up the students at their homes, take them to their respective schools, and bring them back to their residence after the school day.

To ease the burden on local authorities, the Brazilian government financially supports the acquisition of new buses and the use and development of decision support systems to plan the buses' routes. The idea is to guarantee a suitable transportation service level for the students, and acceptable drivers' working hours, while respecting labor union policies and holding the costs down. This gives rise to a multi-objective optimization problem with three distinct and conflicting objectives that must be optimized.

The rich literature of rural school bus routing problems (Park and Kim 2010) has some interesting works—e.g. Thangiah and Nygard (1992), Corberán et al. (2002), Schittekat et al. (2006), and Pacheco et al. (2013)—that deal with a single objective, however only a few authors have addressed multiple objectives. Bowerman et al. (1995) devise a framework to solve an urban school bus routing problem with four different objectives: total bus route length, student walking distance, load and length balancing. By assuming a homogeneous fixed size fleet and known weights of importance for each objective beforehand, they first group the students into clusters in each district, and then plan the bus stops and routes by a set covering algorithm followed by a traveling salesman optimization procedure. They report the solution of a test instance extracted from the Wellington County, Ontario, with 138 students transported by three buses with fifty seats.

Recently, Pacheco et al. (2013) propose a bi-objective procedure for minimizing a normalized objective function consisted of the longest route and the total distance traveled for a rural school bus routing problem with a homogeneous fleet. Like Bowerman et al. (1995), they assume the weights of each objective to be known beforehand for their tabu search algorithm enriched with a multi-objective adaptive memory programming procedure. Pacheco et al. solve sixteen instances corresponding to middle schools in the Province of Burgos, Spain, having the largest test set 57 bus stops, 429 students, and fifteen buses. Their results show that their approach outperforms a non-dominated sorting genetic algorithm.

Both aforementioned works adopt the weighted sum form of multi-objective optimization, which is commonly used due to its simplicity. It merges multiple objectives into a single weighted one given that the importance (i.e. the weights' values) of each objective is known beforehand. This requires the prescription of weights which compensates for differences in magnitudes while indicating the importance of each objective (Marler and Arora 2010). An alternative approach, like the one here adopted, is to find multiple trade-off non-dominated solutions with a wide range of objective values and let the decision maker to choose the most suitable solution for their needs.

Adopting a homogeneous dedicate fleet per school may not be the most appropriate for the Brazilian context, given its dimensions and how rural students are usually sparsely scattered (Carvalho et al. 2010). A homogeneous fleet with large capacity vehicles will most likely have many routes with vacant seats. Filling these vacant seats by increasing the trips' length may not be in the students or drivers' best interest, since to ride on long poorly maintained road routes are usually tiresome. On the other hand, using a homogeneous fleet with small or midsize buses requires hiring many drivers leading to large labor costs.

A better alternative is to deploy a heterogeneous fleet which allows decision makers to balance between routes' lengths, fleets and vehicles' sizes, and routing and labor costs. Vehicle capacity utilization can be also further improved by using mixed loads during rides, i.e. students of different schools ride at the same time on the same vehicle, not having thus a dedicated fleet per school. Both features have already been addressed in the literature, but separately.

Golden et al. (1984) are the first authors to consider a heterogeneous fleet for vehicle routing problems, while Corberán et al. (2002) and Pacheco and Martí (2006) introduce it for a rural school bus routing problem with single loads. Whereas mixed loads are first discussed by Bodin and Berman (1979) and Chen and Kallsen (1988). They single out that mixed loads occur frequently in rural areas, and that an excessive number of buses occurs when single loads are adopted for poorly populated areas.

Braca et al. (1997) minimize the number of buses for a school bus routing problem by proposing an insertion procedure capable of handling mixed loads. The method constructs each route by randomly selecting a bus stop and inserting it and its respective associated school into a route at the best cost estimation possible, while respecting the buses' capacities. Park et al. (2012) improve the method of Braca et al. (1997) by devising a post improvement procedure. Starting from a solution with a dedicate fleet per school obtained from a sweep based algorithm (Gillett and Miller 1974), the procedure reallocates one bus stop at a time in a greedy way until routes can be merged or deleted.

Finally, Li and Fu (2002) claim to have devised a multi-objective optimization algorithm for an urban bus school routing problem with heterogeneous fleet. They minimize four different objectives (the total number of buses; the students' waiting time at the bus stops; the total riding time; and the routes' load unbalance) which are solved separately and sequentially by four different heuristics sorted by the relative importance of the objectives. Each procedure receives as input the solution obtained by the immediate predecessor procedure. At the end of the framework, only one solution is returned. They neither use an objective function consisted of a weighted sum of objective functions; nor they supply a pool of non-dominated solutions to select one from at the end.

Motivated by the social relevance of the application, and the scarce literature on the theme, the objective of this study is threefold: (i) to design a framework that can assist public managers in their decision process; (ii) to improve a known meta-heuristic framework (iterated local search) to better handle multi-objective problems by proposing enhancements that can easily be adapted for other multi-objective optimization problems; and, as far the authors' knowledge, (iii) to solve for the first time a capacitated rural school bus routing problem with heterogeneous fleet and mixed loads under a multi-objective perspective.

Five multi-objective heuristics are here devised for the rural school bus routing problem with heterogeneous fleet and mixed loads. The problem seeks to minimize three conflicting objectives: the routing and fixed total costs; the students' total weighted traveling time; and the routes' unbalance riding times. The devised methods return non-dominated solutions in a set named Pareto frontier from which a decision maker can select the most suitable one for his needs.

The first heuristic is a standard multi-objective iterated local search (MOILS) (Assis et al. 2013). The second has an enhanced insertion step into the Pareto frontier that improves the previous method. The last three heuristics embed different strategies to better the quality of the non-dominated solution frontier: a path-relinking (PR) (Glover et al. 2000) like step; an actual PR procedure; and finally a combination of the two PR schemes. The heuristics' performances were assessed by observing three known metrics (cardinality, coverage, hyper-volume).

The next sections are organized as follows: Sect. 2 explains the implementation details of the procedures used in the solution approaches devised on Sect. 3 to solve the problem. Section 4 shows the attained results. Finally in Sect. 5, conclusions and future researches are presented.

2 Implementation details

The problem uses the following definitions: let $G = (N, A)$ be a directed graph where $N = \{0\} \cup P \cup H$, 0 represents the garage node, P is the set of student nodes with $n_p = |P|$, $H = \{n_p + 1, \ldots, n_p + n_s\}$ is the set of schools and n_s is the number of schools. Further, let B be the set of available buses with $n_b = |B|$. Let also $A = \{(i, j): i, j \in N, i \neq j\}$ and $E = \{(i, j) \in A : i < j\}$ be the arc and edge sets, respectively.

Moreover, let $d(i) = s$ be a function which returns the school $s \in S$ associated to given a student node $i \in P$. Here it is assumed that each student node has only one school associated to it. If a student node has more than one school associated to it, this node is then split into other student nodes, one per school. Each node $i \in P$ has q_i students to be picked up and taken to their respective schools. There are also an associated riding time $t_{i,j}$ to arc $(i, j) \in A$, and a cost $c_{ij}^b \geq 0$ for a bus $b \in B$ to ride edge $(i, j) \in E$. The capacities and the fixed costs of the buses are given by Q^b and a_b, respectively.

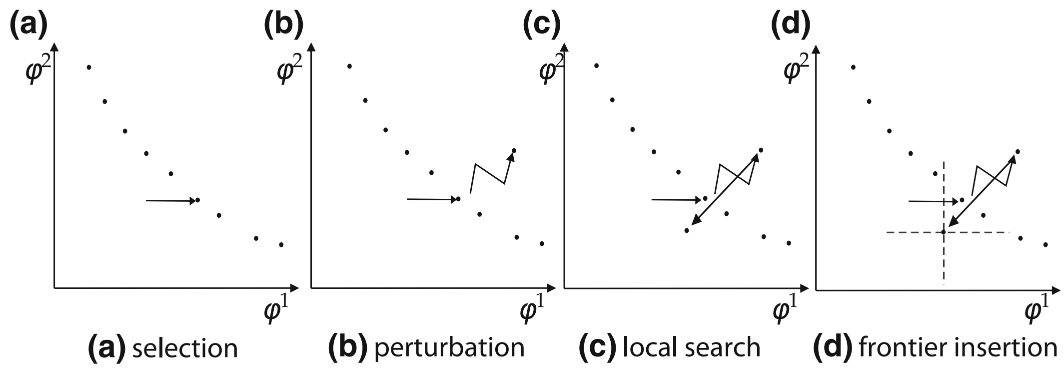

Fig. 1 MOILS general ideal, **a** selection, **b** perturbation, **c** local search, **d** frontier insertion

Five MOILSs are devised to solve the multi-objective mixed load rural bus routing problem with a heterogeneous fleet. The algorithms are thought to have the least number of parameters to be tuned and to be easy to implement and use. Before presenting the heuristics, some important concepts and implementation details are introduced. The MOILS general idea is depicted in Fig. 1 for two objectives. A solution with the largest crowding distance (Deb et al. 2002) is selected from a non-dominated frontier set (Fig. 1a). The selected solution is randomly perturbed to obtain a new solution that is most likely to be dominated by the others in the non-dominated frontier set (Fig. 1b). Then a local search is applied on this new solution (Fig. 1c) to get an improved solution. A dominance checking (Fig. 1d) is performed to verify if the attained solution is dominated or dominates any solution in the non-dominated solution set.

Algorithm 1 depicts the MOILS' steps. The method starts with a set of non-dominated solutions containing only extreme solutions (line 3). How these extreme solutions are generated is explained in Sect. 2.2. For a fixed number of iterations $\mathbb{H}_{max}$, the solution with the largest crowding distance (see Sect. 2.3) is selected from the frontier set $\mathbb{S}$ of non-dominated solutions (line 6). The crowding distance metric allows less explored regions to have higher selection priority during the search. After selecting a solution, the method iterates between a perturbation phase (line 9) followed by a local search (line 10) as long as it is possible to insert non-dominated solutions in the frontier $\mathbb{S}$ (line 11) or until the maximum number of iterations $\mathbb{C}_{max}$ without updating the frontier is reached. The local search attained solution is only inserted into the frontier set if Pareto dominance checking is successful (Sect. 2.3). Assis et al. (2013) suggest the use of a local search consisted of several neighborhood structures for each objective and do the dominance checking only afterward. These suggestions are here altered and improved given rise to four different heuristics.

2.1 Solution representation

Each solution consists of a set of routes represented by a double-linked structure (Li et al. 2007) which stores the predecessor and successor nodes of each bus stop in a route. Moreover a sorted fixed-length neighbor list is assembled for each node containing bus stops that are within 60% of the largest distance among the bus stops of the instance being addressed. This neighbor list is restricted to 20% of the total nodes.

Algorithm 1 MOILS framework

$\mathbb{H}_{max}$: maximum number of iterations
$\mathbb{C}_{max}$: maximum number of iterations without frontier insertion
$\mathbb{S} \leftarrow InitialExtremeSolutions()$
$h \leftarrow 0$
while $h < \mathbb{H}_{max}$ **do**
  $s \leftarrow CrowdingDistanceSelection(\mathbb{S})$
  $c \leftarrow 0$
  while $c < \mathbb{C}_{max}$ **do**
    $s' \leftarrow Perturbation(s)$
    $s'' \leftarrow LocalSearch(s')$
    if $DominanceCheckingInsertion(\mathbb{S}, s'') = True$ **then**
      $s \leftarrow s''$
      $c \leftarrow 0$
    else
      $c \leftarrow c + 1''$
    end if
  end while
  $h \leftarrow h + 1$
end while

2.2 Extreme solutions: initial solutions

The first solutions to be inserted in the frontier of non-dominated solutions are the extreme solutions, i.e. solutions posited as the best ones for each objective. They represent the extremest points of the Pareto set for each objective. When an extreme solution for a particular objective is calculated, the other two objectives are set to zero. To generate each extreme solution, different strategies were adopted:

2.2.1 Total routing and fixed costs

The extreme solution for the total cost is obtained after running a heuristic devised by Lima et al. (2016) for a mixed load capacitated rural school bus routing problem with a heterogeneous fleet. The heuristic combines an iterated local search (ILS) with a Clark and Wright savings procedure (Clark and Wright 1964) to generate the initial solutions, and a random variable neighborhood descent (RVND) (Hansen and Mladenović 2001) search with four different neighborhood structures. The best overall solution is inserted into the frontier of non-dominated solutions.

2.2.2 Total students' weighted traveling time

To generate routes with the least weighted traveling time, one bus is assigned to each student node. This bus has the smallest capacity possible such that it can still serve the student node demand.

2.2.3 Routes' unbalance riding time

To generate the extreme solution for the routes' unbalance riding time objective, one bus is firstly allocated for each student node. Each bus is selected with the smallest possible capacity such that it can still serve the student node demand. Pair of routes are then merged, followed by a *2Opt* procedure (see Sect. 2.4) to avoid artificial balancing. An artificial balancing occurs whenever unnecessary rides are carried out just to artificially increase the time of a route with the purpose of having at the end all routes with about the same length. If the merge improves the overall balance then this merge is executed and the balance is updated. If the merge exceeds the bus capacity, a larger available bus is used. Otherwise the merge is not carried on. The procedure merges pair of routes until no further improvement is possible. The routes' unbalance riding time is measured by subtracting the largest riding time by the shortest one.

2.3 Crowding distance and dominance checking

The crowding distance ω_s of a solution s of a set $\mathbb{S}$ of non-dominated solutions provides an estimation of the density of the solutions surrounding solution s (Deb 2001). It is computed as the summation of each normalized distance ϖ_s^j for each objective $j = 1 \ldots n_o$, where n_o is the number of objectives. ϖ_s^j is calculated as the distance difference of the two adjacent solutions of a solution s normalized by the largest distance of the referred objective j being addressed. Distance is interpreted as the difference in value of the objective function of two different solutions. A larger crowding distance indicates a solution which has fewer solutions around it, being a natural candidate to have its neighborhood searched for other solutions. This way the diversity of the frontier set of non-dominated solutions is most likely to be increased. Figure 2 illustrates the crowding distance concept.

When a new solution is obtained, it has to be verified if it is dominated or not by the other solutions in the frontier set of non-dominated solutions. Hence during the search for non-dominated solutions, a dominance checking is performed every time a candidate solution is found. This dominance checking is based on the Pareto dominance of Definition 1.

Definition 1 Pareto Dominance: a solution $s_1\left(\varphi_1^1, \ldots, \varphi_1^{n_o}\right)$ dominates a solution $s_2\left(\varphi_2^1, \ldots, \varphi_2^{n_o}\right)$, denoted by $s_1 \prec s_2$, if and only if $\varphi_1^j \leq \varphi_2^j$, $\forall\, j \in \{1, \ldots, n_o\}$, and $\exists r \in \{1, \ldots, n_o\}$ such that $\varphi_1^r < \varphi_2^r$ (Pareto 1896, 1897).

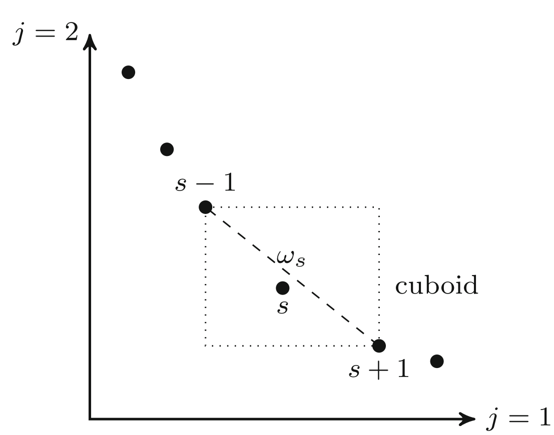

Fig. 2 Crowding distance for two objectives

2.4 Neighborhood structures

Six neighborhood structures inspired on the works of Laporte and Semet (2002) and Gendreau et al. (2002) are used to find non-dominated solutions to populate the frontier of solutions. The structures are divided into two types: inter-routes (Fig. 3) and intra-routes (Fig. 4).

Inter-routes structures search for improving solutions doing movements between pair of routes. Four different structures are proposed: (**a**) **The one-point move** relocates a student node from a route to a different route. The Fig. 3a shows that the student node 5 is moved from a dashed line route to the bold line route. The order of the visited schools needs to be rearranged since the bold line route is not required to visit the gray school anymore. (**b**) **The two-point move** exchanges one student node of a route by another of a different route. In the example on Fig. 3b, the student nodes 5 and 3 are exchanged from their original routes, i.e. node 5 is removed from its original dashed line route and placed at the position of node 3 in the continuous line route. Node 3 is removed from its original continuous line route replacing node 5 in the dashed line route. The order of visited schools has also to be rearranged. (**c**) **cross-exchange move** removes arcs (i, j) and $\left(i', j'\right)$ of two different routes reconnecting the nodes as $\left(i, j'\right)$ and $\left(i', j\right)$. Figure 3c shows that arcs $(6, 4)$ and $(3, school)$ are reconnect as two new arcs $(6, school)$ and $(3, 4)$. (**d**) **merge routes move** choses two random routes to merge but respecting the largest bus capacity. The last node of a route is connected with the first one of another route. If the movement improves the objective function than it is executed; otherwise, another merge is tried. In Fig. 3d dashed and bold line routes are merged through the connection of nodes 3 with node 7.

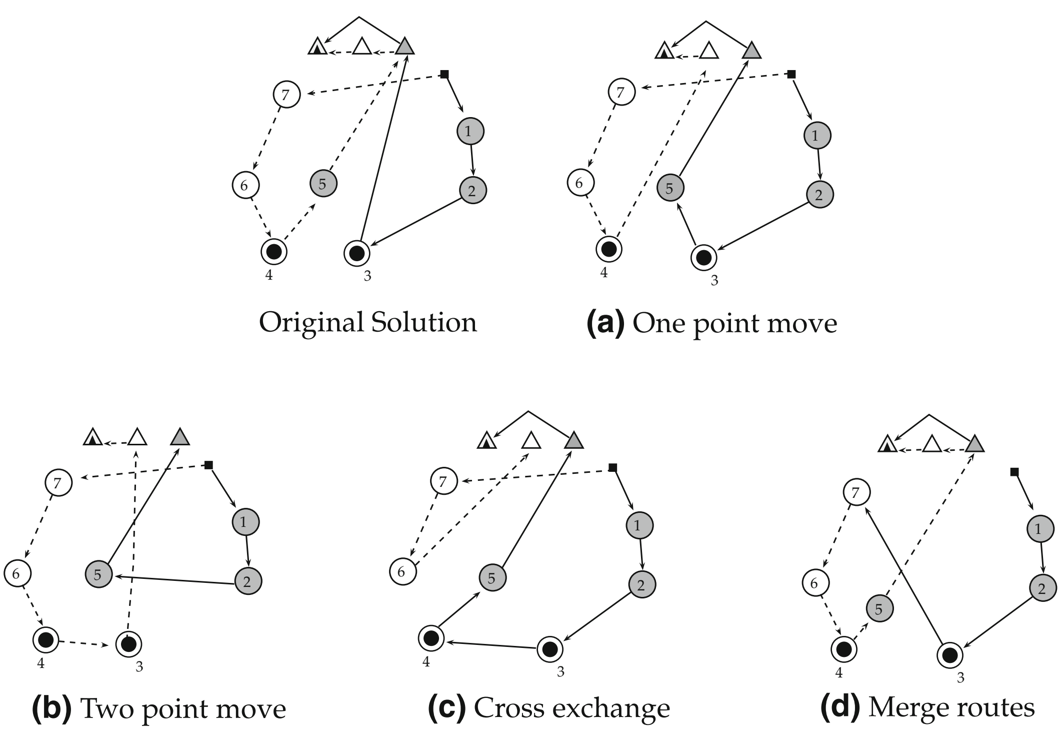

Fig. 3 Inter-routes local search operators, **a** one point move, **b** two point move, **c** cross exchange, **d** merge routes

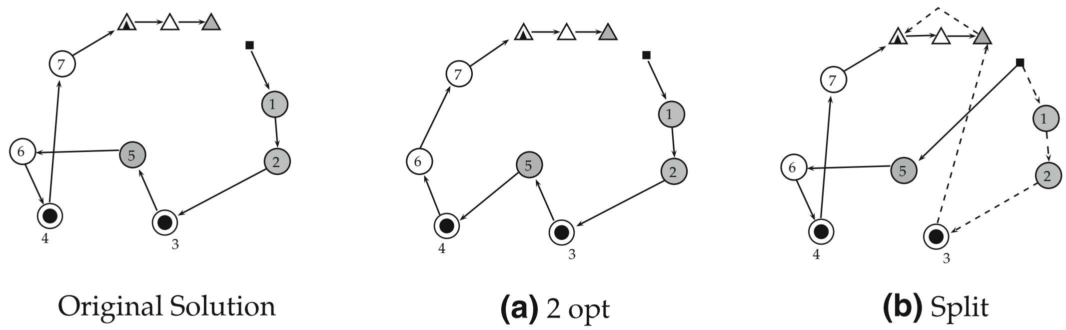

Fig. 4 Intra-routes local search operators, **a** 2 opt, **b** split

Intra-routes apply a movement within a route. Only two structures are devised: (**a**) **2opt move** removes two non-consecutive arcs, e.g. (i, j) and $\left(i', j'\right)$, of a route and reconnects them by linking the heads and the tails together, or $\left(i, i'\right)$ and $\left(j, j'\right)$. In Fig. 4a, the original solution is modified by removing arcs (5, 6) and (4, 7), replacing them for new ones (5, 4) and (6, 7). (**b**) **split move** randomly selects a route and divides it into two new routes on the arc next to the middle of the route. In Fig. 4b the original solution has its route split into two new ones by removing the arc (3, 5), the arc closest to the middle of the route.

Is important to remark that the split move is applied for all of the objectives but the total cost one, because this movement does not improve the total cost function, since one has to pay for an additional bus. Further, the first improving movement strategy is used for all the neighborhood structures.

When applied on the pickup section, these movements may require modifications in the delivery section. Despite all neighborhood structures are organized in a random variable neighborhood descent (RVND) (Hansen and Mladenović 2001) search, there are only three cases in which the delivery section of a route will need to be changed: (i) if schools are no longer needed to be visited in a route; (ii) if new schools are required to be visited in a route; and (iii) if the last visited student node is changed. Checking for these three cases is very time consuming.

Given a current or initial solution and a list of available neighborhood structures, a RVND search randomly selects a neighborhood structure among the list of available structures. Then a local search is performed in this neighborhood with the hope of improving the current solution. If successful, the current solution is updated. The list of available movement structures is also augmented with any removed neighborhoods. In case of not getting a better solution, the current movement structure is removed from the list and the process restarted. These steps go on while the list of available neighborhood structures is not empty.

2.5 *Walk* procedure

Generally speaking, a path relinking (PR) (Glover et al. 2000) strategy generates new solutions from a given solution when trying to reach a target solution randomly selected from an elite set. An elite set usually has a restricted number of the overall best solutions that are structurally different, i.e. that have some degree of dissimilarity.

The idea is to “walk” from a guided solution over a path of solutions towards a guiding solution in the elite set by performing small structural changes in each step. At each step, a local search is carried out with the hope of obtaining better solutions. In the present case, the dissimilarity of a pair of solutions ($\mathbb{D}$) is measured by the summation of the total number of different arcs used in the routes.

For the devised heuristics, two different schemes based on the PR method are implemented. One is an actual PR phase embedded into MOILS based heuristics. This PR phase maintains an elite set for each objective which is constantly updated if necessary. The other scheme resembles a PR method, but instead of keeping an elite set for each objective, the adjacent solutions of the solution selected by the crowding distance procedure is used as guiding and guided solutions. This scheme greatly increases the diversity of the frontier set of non-dominated solutions.

Both PR based schemes rely on a *Walk* procedure. This procedure executes the best possible movement among splitting or merging routes, and inter and intra node moves. These movement structures are done in separated phases. In the beginning, the *Walk* procedure effort is toward having the guided solution with the same number of routes of the guiding solution. This is accomplished by applying splitting or merging routes routines. After that, routes in the guided solution are relabeled so that each route has a correspondent route in the guiding solution.

Inter route node movements are then performed so that at the end routes of the guided solution will have the same nodes of the guiding solution. A node is inserted in a route after observing the best improvement possible. Finally, intra route node movements are done so that the routes will have the same order of visitation. After each phase, a RVND search is realized with the hope of obtaining better solutions and improving the quality of the frontier set of non-dominated solutions.

As the PR based schemes are computationally time consuming, a prespecified number of iterations is actually set to limit this effort. So at the end the implemented PR based procedures may not have an exact match, but a guided solution that resembles the guiding solution. At each non-dominated solution found during the *Walk* procedure, the frontier of non-dominated solutions is updated accordingly.

3 New MOILS heuristics

The Algorithm 1 is enhanced with minor modifications and procedures to improve the overall quality of the attained solutions. Assis et al. suggest the adoption of many distinct specialized neighborhoods for each addressed objective. They propose the use of all of these movement structures during the local search phase which can possibly result in a zigzag phenomenon.

Depending on how the neighborhood structures are organized or selected, a neighborhood may improve one objective, while the next one to be used in sequence will most likely go in a different direction, most probably worsening the previous gain obtained by the former movement structure.

Further, as the achieved solution is only verified to be inserted into the frontier of non-dominated solutions at the end of the local search phase, a failure in this test results in a waste of computational effort and time. To circumvent the implementation of many different neighborhood structures, the zigzag phenomenon, and the possibility of failure after the local search phase, a modified MOILS algorithm is proposed.

3.1 A modified multi-objective iterated local search (MOILS-M)

To increase the success rate of generating good non-dominated solutions, the neighborhood structures of Sect. 2.4 were slightly modified. The policy of adopting the first improving move was changed to the first improving move which results in a non-dominated solution, i.e. the frontier set is passed as an input to the local search phase so that the dominance checking (Sect. 2.3) can be done along the search inside the neighborhood structure.

Further, instead of having different specialized neighborhood structures for each objective, each neighborhood is set to work with any objective. To accomplish that, the objectives which are being optimized has to be informed in the beginning of the local search. For instance, the one point move in Fig. 3a modifies a route in its cost, in the total weighted traveling time, and the unbalance of the routes. Hence it can be used to obtain good solutions with respect to any of the objectives, given that the appropriate direction is adopted. So when trying to improve the unbalance of the routes, the search direction of the moves carried out by the one point move has to be in this direction, and not, for example, toward decreasing the costs. To prevent the zigzag phenomenon, the local search is performed separately for each objective one at a time.

The aforementioned gives rise to the modified MOILS presented in Algorithm 2. This pseudo-algorithm has the same steps of Algorithm 1, but with the difference depicted by lines 11–14, and variables $\mathbb{I}$ and Υ. The local search is done in lines 11–14. Note that it receives as input not only the perturbed solution but the frontier set of non-dominated solutions, the current objective being addressed, and variable $\mathbb{I}$ and parameter $\mathbb{I}_{max}$.

As the added enhancements to the MOILS greatly increases the cardinality of the frontier (see section for the results), parameter $\mathbb{I}$ counts the number of insertions made for the current objective. An insertion limit $\mathbb{I}_{max}$ is established to prevent too many insertions in the neighborhood of a single solution. If any solution is added to the frontier set $\mathbb{S}$, then variable Υ receives the value true. Otherwise Υ is set to false. If the maximum number of insertions are made, then Υ is also set to false. Whenever the number of unsuccessful insertions reach $\mathbb{C}_{max}$, a new solution is selected by the crowding distance selection procedure of line 7. The algorithm iterates for $\mathbb{H}_{max}$ iterations when it returns the frontier set to the decision maker for analysis.

3.2 Multi-objective iterated local search with a standard path relinking (MOILS-PR)

This variant embeds a standard path relinking (PR) procedure (Martí and Campos 2011) into the method. The PR integrates intensification and diversification in the search process, exploring paths that connect previously found solutions (guided solutions) with high quality solutions (guiding solutions) stored in an elite set (Ξ). The idea is similar to the MOILS-PRA (Algorithm 4). However in the standard PR, the

Algorithm 2 Modified MOILS (MOILS-M)

```
ℍmax: maximum number of iterations
ℂmax: maximum number of iterations without frontier insertion
𝕀max: maximum number of insertions in the frontier
$ ← InitialExtremeSolutions()
h ← 0
while h < ℍmax do
  s ← CrowdingDistanceSelection($)
  c ← 0
  while c < ℂmax do
    s' ← Perturbation(s)
    for j = 1 to n_o do
      𝕀 ← 0
      Υ ← LocalSearch(s', φ^j, $, 𝕀, 𝕀max)
    end for
    if Υ = True then
      c ← 0
    else
      c ← c + 1
    end if
  end while
  h ← h + 1
end while
```

guiding solution is chosen from an elite set (see lines 13 of Algorithm 3) instead of the adjacent solutions of the solution with the largest crowding distance.

At every $\mathbb{W}$ iterations, an elite set $\varXi$ is assembled for every objective (line 12). Then an elite solution s_e is randomly selected from the elite set $\varXi$ (line 13). Then the method proceeds in the same way as Algorithm 4. The elite set $\varXi$ is assembled with the best overall solutions of the frontier $\mathbb{S}$ with respect with every objective, but with at least 10% of dissimilarity. Note that the degree of dissimilarity $\mathbb{D}$ (lines 15 and 16) between both solutions is used as a stopping criterion for the *Walk* procedure. If in $\mathbb{D}$ moves the solutions do not match, the procedure is stopped.

3.3 A multi-objective iterated local search with a path relinking procedure for the crowding distance adjacency (MOILS-PRA)

An enhanced feature is added to Algorithm 2 given rise to Algorithm 4. A *Walk* procedure (Sect. 2.5) is executed every $\mathbb{W}$ iterations. The idea is to increase the diversity of the frontier $\mathbb{S}$ on its most sparse regions. Further, by doing a RVND along the *Walk* procedure, not only diversification but intensification as well is also sought to further improve the quality of the frontier set.

In Algorithm 4, the immediate predecessor (s_p) and successor (s_p) solutions (i.e. adjacent solutions) of the solution (s) with the current largest crowding distance, see lines 9–20, are selected to be used as the guiding and guided solutions. The degree of dissimilarity $\mathbb{D}$ (lines 15 and 16) between both solutions is also used as a stopping criterion for the *Walk* procedure as previously mentioned. If in $\mathbb{D}$ moves the solutions do not match, the procedure is stopped.

Algorithm 3 MOILS with standard path relinking procedure (MOILS-PR)

$\mathbb{H}_{\max}$: maximum number of iterations
$\mathbb{C}_{\max}$: maximum number of iterations without frontier insertion
$\mathbb{I}_{\max}$: maximum number of insertions in the frontier
$\mathbb{S} \leftarrow InitialExtremeSolutions()$
$h \leftarrow 0$
$w \leftarrow 0$
while $h < \mathbb{H}_{\max}$ **do**
  $s \leftarrow CrowdingDistanceSelection(\mathbb{S})$
  if $w = \mathbb{W}$ **then**
    $w \leftarrow 0$
    for $j = 1$ to n_o **do**
      $\Xi \leftarrow EliteSet(\mathbb{S}, \varphi^j)$
      $s_e \leftarrow RandomSelection(\Xi)$
      $\mathbb{D} \leftarrow DissimilarityDegree(s, s_e)$
      $Walk(s, s_e, \varphi^j, \mathbb{S}, \mathbb{I}_{\max}, \mathbb{D})$
      $Walk(s_e, s, \varphi^j, \mathbb{S}, \mathbb{I}_{\max}, \mathbb{D})$
    end for
  else
    $w \leftarrow w + 1$
  end if
  $c \leftarrow 0$
  while $c < \mathbb{C}_{\max}$ **do**
    $s' \leftarrow Perturbation(s)$
    for $j = 1$ to n_o **do**
      $\mathbb{I} \leftarrow 0$
      $\Upsilon \leftarrow LocalSearch(s', \varphi^j, \mathbb{S}, \mathbb{I}, \mathbb{I}_{\max})$
    end for
    if $\Upsilon = True$ **then**
      $c \leftarrow 0$
    else
      $c \leftarrow c + 1$
    end if
  end while
  $h \leftarrow h + 1$
end while

The walk along the "path" of solutions from s_p to s_s is done in both directions for every objective. Recall from Sect. 2.5 that on every change of movement phase of the *Walk* procedure, a RVND search is executed. The number of allowed insertions into the frontier of non-dominated solutions $\mathbb{S}$ during the RVND calls is given by $\mathbb{I}_{\max}$. The remainder of the algorithm is the same of Algorithm 2.

3.4 Multi-objective iterated local search combining MOILS-PR with MOILS-PRA (MOILS-PR-PRA)

The last devised algorithm is a combination of both Algorithms 4 and 3. For sake of presentation, it is not shown here. Basically, the algorithm executes both path relinking strategies at every $\mathbb{W}$ iterations.

Algorithm 4 MOILS with a path relinking procedure for the criterion of the crowding distance adjacency (MOILS-PRA)

```
$\mathbb{H}_{\max}$: maximum number of iterations
$\mathbb{C}_{\max}$: maximum number of iterations without frontier insertion
$\mathbb{I}_{\max}$: maximum number of insertions in the frontier
$\mathbb{S} \leftarrow InitialExtremeSolutions()$
$h \leftarrow 0$
$w \leftarrow 0$
while $h < \mathbb{H}_{\max}$ do
  $s \leftarrow CrowdingDistanceSelection(\mathbb{S})$
  if $w = \mathbb{W}$ then
    $w \leftarrow 0$
    $s_p \leftarrow pred(s)$
    $s_s \leftarrow succ(s)$
    $\mathbb{D} \leftarrow DissimilarityDegree(s_p, s_s)$
    for $j = 1$ to $n_o$ do
      $Walk(s_p, s_s, \varphi^j, \mathbb{S}, \mathbb{I}_{\max}, \mathbb{D})$
      $Walk(s_s, s_p, \varphi^j, \mathbb{S}, \mathbb{I}_{\max}, \mathbb{D})$
    end for
  else
    $w \leftarrow w + 1$
  end if
  $c \leftarrow 0$
  while $c < \mathbb{C}_{\max}$ do
    $s' \leftarrow Perturbation(s)$
    for $j = 1$ to $n_o$ do
      $\mathbb{I} \leftarrow 0$
      $\Upsilon \leftarrow LocalSearch(s', \varphi^j, \mathbb{S}, \mathbb{I}, \mathbb{I}_{\max})$
    end for
    if $\Upsilon = True$ then
      $c \leftarrow 0$
    else
      $c \leftarrow c + 1$
    end if
  end while
  $h \leftarrow h + 1$
end while
```

4 Computational experiments

This section describes the computational experiments performed to test the efficiency of the aforementioned heuristics. All algorithms were coded in C++, compiled with GCC 4.8.1, and tested on an Intel Xeon 2.53 GHz with 24 GB RAM running Linux Mint 16.

For the experiments, three bus types with capacities of 20, 30 and 40 seats were made available to transport the students. The fixed costs (daily fixed depreciation costs) and the routing costs for each bus type were set to \$100, \$150, and \$200, and \$1.00, \$1.20 and \$1.40 per unit of traveled distance, respectively. The daily depreciation costs (fixed costs) were estimated by assuming a bus lifespan of 10 years and the distances between nodes were considered to be Euclidean.

To test the devised heuristics, three sets of experiments were performed. The first two use 15 random instances generated to depict Brazilian counties with different sizes.

They assess how the methods perform when limiting the number of iterations, and the total computational running time, respectively. The last set uses the real instances of Park et al. (2012), but assuming a dedicated fleet per school, i.e. single loads.

The random instances were generated by using the following features: The bus stops were scattered within the set {50, 75, 100, 150} in an area of 155 mi^2 (12.5 mi × 12.5 mi). The number of schools were varied according to the set {5, 10, 20}. Moreover 20% of the bus stops as well as all schools were located inside an imaginary downtown area with a radius of 1.24 miles. The remaining nodes were located outside of this radius. For each node, a school and its respective demand were uniformly selected. The demands $q_l, l \in P$, were generated within the range of 1 to 3 students. A total of 15 random instances were devised, being 12 instances to assess the devised heuristics, and three for the calibration phase. Calibration instances had the number of stops and schools set to (50, 5), (75, 5) and (100, 10). All test problems are available from the corresponding author upon request.

In multi-objective optimization, one wishes to find a frontier set of non-dominated solutions, which keeps the best compromise solutions among all the objectives, since finding a Pareto optimal set is usually not possible or computationally prohibitive (Batista et al. 2014). Hence it is important to analyze the quality of the attained frontiers since non-dominated solutions are typically worse than then ones of a Pareto optimal frontier. One can use a metric such as diversity or convergence to measure the frontier set quality, but since they can be conflicting sometimes, it is advised to adopt more than one (Deb 2001). For that purpose three metrics are considered:

1. **Cardinality (car):** number of solutions in the Pareto frontier. Its purpose is to verify the efficiency of the developed algorithm in finding non-dominated points. It is assumed that the decision maker prefer rather more options than fewer efficient solutions (Martí and Campos 2011).
2. **Coverage of many sets (CS):** this metric was proposed by Batista et al. (2014) and is a generalization of the Coverage of two sets (Zitzler and Thiele 1999). This metric indicates the percentage of a set $\mathbb{U}$ that is dominated by a set $\mathbb{S}$. It quantifies the domination of a frontier over the union of the remaining ones. The function which gives this value is stated as:

$$CS\left(\mathbb{S}_i, \mathbb{U}_i\right) = \frac{|\{s_1 \in \mathbb{U}_i : \exists s_2 \in \mathbb{S}_i \wedge s_2 \preceq s_1\}|}{|\mathbb{U}_i|}$$

where $\mathbb{S}_i$ represents the Pareto frontier of algorithm i, for all $i = 1, \ldots, n_a$, and n_a is the number of available algorithms, and $\mathbb{U}_i$ is the union of all Pareto frontier of all algorithms except algorithm i. This super set is defined as:

$$\mathbb{U}_i = \bigcup_{\substack{j=1 \\ j \neq i}}^{n_a} A_j \quad \forall i = 1, \ldots, n_a$$

The expression $s_2 \preceq s_1$ means that the solution s_2 is no worse than s_1 in all objectives and the solution s_2 is strictly better than s_1 in at least one objective. This concept defines weakly dominance. The function CS maps the pair ($\mathbb{S}_i$,$\mathbb{U}_i$)

within the interval [0, 1]. The value $CS(\mathbb{S}_i, \mathbb{U}_i) = 1$ means that all points in $\mathbb{U}_i$ are dominated by or equal to the points of $\mathbb{S}_i$. Likewise, when $CS(\mathbb{S}_i, \mathbb{U}_i) = 0$ represents that the frontier $\mathbb{S}_i$ of the algorithm i is dominated by the others.

3. **Hyper-volume:** introduced by Zitzler and Thiele (1999), this indicator measures the volume of the region dominated by the Pareto front with respect to a reference point which can be defined by a vector of the worst values for the objective function. In this work the results of hyper-volume metrics were normalized by dividing the difference between the current value and the minimum value of the objective function by the difference of the maximum and minimum value of the objective function. The reference point was set to $\mathscr{H}_0(1, 1, 1)$.
   Mathematically, for each solution $s \in \mathbb{S}$, a hyper-cube $\mathcal{V}_s$ is created accordingly to the reference point $\mathscr{H}_0$. The final result is the sum of all obtained hyper-cubes. Assuming minimization objectives, a higher value for the hyper-volume evinces a higher spread among solutions of the Pareto front as well as a higher convergence. A more detailed explanation about how to compute hyper-volume please refer to Beume et al. (2009) and Bradstreet (2011).

4.1 Calibration phase

1. **Maximum number of iterations** ($\mathbb{H}_{max}$): to define the parameter $\mathbb{H}_{max}$, required by all algorithms, 15 executions for each one of the three calibration instances were executed. The results showed that $\mathbb{H}_{max} = 100$ presented better results than $\mathbb{H}_{max} = 50$ for cardinality in MOILS-PR (28.9), MOILS-M (12) and MOILS (42.5%). For the hyper-volume, $\mathbb{H}_{max} = 100$ was also better in the algorithms: MOILS-PR (6.5), MOILS-M (6.2) and MOILS (2.9%). For coverage the $\mathbb{H}_{max} = 100$ had also the better results on MOILS-PR (32) and MOILS (30%) except for MOILS-M whose average was 44% better. The value of 150 for $\mathbb{H}_{max}$ returned best results only for coverage on MOILS-M (11%) and hyper-volume on MOILS (0.03%) better values for cardinality is expected because a higher number of iterations allows more solutions to get into the Pareto set increasing the result for this metric. Thus, based on the results the $\mathbb{H}_{max}$ parameter was settled as 100.
2. **Maximum number of iterations without insertion** ($\mathbb{C}_{max}$): the parameter $\mathbb{C}_{max}$ (maximum number of iterations without any insertion in the frontier, Algorithm 1) was tested at the same calibration instances with the values $\{1, 2\}$. Five runs for each algorithm on each instance were performed. The best values for cardinality, coverage and hyper-volume were obtained when $\mathbb{C}_{max} = 1$ mainly because as many solutions are inserted in the frontier $\mathbb{S}$, the counter c takes a long time to reach the value $\mathbb{C}_{max} = 2$. In this case, many solutions are also inserted, however they are close to each other which does not improve the hyper-volume or the spread, besides preventing further neighborhoods to be visited. Thus, this metric was set to $\mathbb{C}_{max} = 1$.
3. **Maximum insertion** ($\mathbb{I}_{max}$): The value of $\mathbb{I}_{max}$ was set to the number of stops of the instance. As the instance size increases, the number of values to be inserted also increases.

4. **Iterations between** *Walk* **procedure calls** ($\mathbb{W}$): the *Walk* procedure is very time consuming not being able to be applied at each iteration. The number of iterations should not be set as a fixed number, instead it should vary with the instance size. Three different values were tried $\lceil(n/2)\rceil$, $\lceil(n/4)\rceil$ and $\lceil(n/8)\rceil$. The value $\lceil(n/8)\rceil$ presented the best overall results.

4.2 Statistical analysis

To evaluate the methods' performances, a statistical analysis was made to identify their differences and magnitudes, if they exist. For each metric a pairwise comparison was applied independently through a T test, summing ten null hypothesis of absence of difference between the algorithms.

As many tests are performed, a correction on the alpha level is necessary to prevent **Type I** error, i.e to reject a null hypothesis when it is true caused the alpha level is inflated. This problem usually happens when many tests are executed.

When many tests are performed on a data set, the more likely it is to reject the null hypothesis as a consequence of the logic of the hypothesis test. A null hypothesis is rejected if a rare event is witnessed. However the larger the number of tests, the easier it is to find a rare event, and to think that there is an effect, when there is none actually. Therefore, by making the alpha level smaller, less error will be accepted, but hardening the detection of real effects as well (Salkind and Rasmussen 2013).

To correct the alpha inflation, the final alpha significance level was divided by the number of tests as suggested by the Bonferroni correction (Hochberg 1988). As the final significance level desired is 5 and 10% hypothesis had to be tested, the significance level of 0.005 was applied for each T test. The statistical analysis were applied in the average of the results of eight replications for the five heuristics on solving the twelve instances in aleatory mode.

4.3 First experiment results: limiting the total number of iterations

Tables 1, 2, 3 and 4 report the results for the first set of experiments. They show the average for the executions of each instance by each algorithm and the standard deviation, in parenthesis, for the metrics of cardinality, coverage and hyper-volume. The statistical analysis are summarized in Table 5.

For each instance, the results of cardinality (see Table 1) and time (refer to Table 4) were divided by the smallest value obtained among the 8 executions of the five methods (depicted at the last column of the referred tables) to make easier the comparison among them. Observing the statistical analyze in Table 5, one can conclude that all proposed methods overcame the literature framework MOILS in all metrics.

For the cardinality metric, the algorithm MOILS-M was outperformed by the variants MOILS-PR-PRA, MOILS-PRA and MOILS-PR. Among these three methods, the MOILS-PRA was outperformed by the versions MOILS-PR-PRA and MOILS-PR, which was able to generate 7.12 and 5.61 times more solutions than MOILS-PRA, respectively. However, these last heuristics MOILS-PRA, MOILS-PR-PRA and MOILS-PR are statistically equal. Therefore is possible to conclude that the MOILS-

Table 1 Average (standard deviation) of cardinality

Instance	Cardinality					
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS	Smallest
A50	12.74 (1.08)	11.70 (1.70)	11.53 (1.25)	8.29 (1.59)	1.50 (0.32)	67
B50	11.59 (1.34)	11.79 (0.87)	10.31 (1.47)	8.19 (0.88)	1.33 (0.28)	79
C50	12.76 (1.56)	11.68 (1.22)	11.85 (1.71)	9.66 (1.72)	1.63 (0.47)	67
A75	20.45 (1.59)	17.13 (2.67)	17.51 (3.12)	13.72 (2.05)	1.45 (0.37)	73
B75	22.66 (3.03)	23.01 (3.73)	18.81 (2.34)	16.77 (2.38)	1.74 (0.37)	62
C75	22.05 (1.48)	20.57 (3.01)	18.29 (1.83)	15.77 (1.02)	1.25 (0.23)	59
A100	17.62 (2.46)	18.39 (3.29)	17.35 (2.21)	13.81 (2.12)	1.20 (0.16)	89
B100	16.98 (2.15)	17.36 (2.75)	16.42 (3.50)	12.08 (1.94)	1.37 (0.26)	101
C100	16.93 (2.62)	14.70 (2.54)	15.25 (1.19)	12.03 (1.54)	1.21 (0.19)	105
A150	54.52 (9.30)	48.38 (11.92)	41.38 (6.31)	34.74 (3.83)	1.51 (0.38)	55
B150	29.08 (5.08)	28.04 (5.58)	24.68 (3.39)	20.72 (2.96)	1.44 (0.30)	95
C150	31.97 (3.62)	31.94 (4.55)	27.52 (4.86)	23.96 (3.01)	1.52 (0.49)	81
Average	22.45 (2.94)	21.22 (3.65)	19.24 (2.77)	15.81 (2.09)	1.43 (0.32)	77.75

Table 2 Average (standard deviation) of coverage

Instance	Coverage				
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS
A50	0.60 (0.18)	0.46 (0.14)	0.40 (0.15)	0.19 (0.05)	0.06 (0.05)
B50	0.61 (0.13)	0.44 (0.14)	0.39 (0.20)	0.27 (0.15)	0.03 (0.01)
C50	0.57 (0.15)	0.42 (0.17)	0.41 (0.13)	0.26 (0.08)	0.09 (0.04)
A75	0.51 (0.10)	0.37 (0.17)	0.52 (0.20)	0.24 (0.12)	0.04 (0.03)
B75	0.55 (0.16)	0.40 (0.14)	0.45 (0.16)	0.24 (0.05)	0.06 (0.03)
C75	0.53 (0.07)	0.37 (0.09)	0.39 (0.14)	0.33 (0.04)	0.04 (0.03)
A100	0.40 (0.11)	0.37 (0.14)	0.50 (0.12)	0.29 (0.16)	0.06 (0.03)
B100	0.47 (0.14)	0.31 (0.12)	0.48 (0.17)	0.28 (0.11)	0.10 (0.06)
C100	0.42 (0.22)	0.34 (0.18)	0.51 (0.09)	0.32 (0.11)	0.05 (0.03)
A150	0.46 (0.16)	0.33 (0.14)	0.38 (0.08)	0.32 (0.14)	0.03 (0.02)
B150	0.39 (0.16)	0.29 (0.15)	0.45 (0.11)	0.38 (0.08)	0.06 (0.04)
C150	0.41 (0.13)	0.28 (0.08)	0.46 (0.13)	0.36 (0.10)	0.07 (0.05)
Average	0.51 (0.14)	0.38 (0.14)	0.44 (0.14)	0.27 (0.10)	0.06 (0.03)

PR heuristic embedded in the MOILS-M allows the proposed method to return more non-dominated solutions. Between MOILS-PR-PRA and MOILS-PR, the MOILS-PR is more interesting once it requires smaller computational time to find a better frontier, please refer to Table 4.

To examine the results for coverage (see Table 2) one can follow the same reasoning above. The MOILS-M was outperformed by the variants MOILS-PR-PRA, MOILS-

Table 3 Average (standard deviation) of hyper-volume

Instance	Hyper-volume				
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS
A50	0.93 (0.01)	0.92 (0.01)	0.92 (0.01)	0.90 (0.01)	0.87 (0.04)
B50	0.92 (0.01)	0.91 (0.01)	0.91 (0.01)	0.90 (0.01)	0.85 (0.03)
C50	0.92 (0.01)	0.92 (0.01)	0.92 (0.01)	0.92 (0.01)	0.89 (0.01)
A75	0.94 (0.01)	0.93 (0.01)	0.94 (0.01)	0.93 (0.01)	0.88 (0.02)
B75	0.93 (0.02)	0.93 (0.02)	0.92 (0.02)	0.92 (0.02)	0.86 (0.04)
C75	0.91 (0.01)	0.91 (0.01)	0.91 (0.01)	0.91 (0.01)	0.84 (0.03)
A100	0.90 (0.01)	0.89 (0.02)	0.90 (0.02)	0.90 (0.01)	0.85 (0.03)
B100	0.91 (0.01)	0.91 (0.01)	0.91 (0.02)	0.90 (0.02)	0.87 (0.03)
C100	0.90 (0.01)	0.90 (0.02)	0.91 (0.01)	0.91 (0.02)	0.86 (0.03)
A150	0.90 (0.02)	0.89 (0.02)	0.90 (0.01)	0.90 (0.02)	0.82 (0.04)
B150	0.91 (0.02)	0.90 (0.01)	0.91 (0.01)	0.91 (0.01)	0.86 (0.02)
C150	0.91 (0.01)	0.90 (0.01)	0.92 (0.01)	0.92 (0.01)	0.87 (0.03)
Average	0.92 (0.01)	0.91 (0.01)	0.92 (0.01)	0.91 (0.01)	0.86 (0.03)

Table 4 Average (standard deviation) of time

Instance	Time					Smallest
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS	
A50	20.19 (3.34)	17.87 (2.27)	7.95 (3.80)	3.38 (1.41)	1.46 (0.43)	19.12
B50	11.44 (2.35)	11.56 (2.04)	5.74 (1.88)	3.00 (1.24)	1.56 (0.71)	33.35
C50	15.04 (2.04)	12.44 (2.02)	7.49 (1.33)	4.05 (1.39)	1.89 (0.98)	23.09
A75	14.06 (2.93)	13.02 (3.01)	8.22 (1.44)	4.23 (1.29)	1.37 (0.48)	60.16
B75	18.61 (4.10)	15.81 (4.26)	6.23 (1.84)	4.20 (1.10)	1.80 (0.71)	75.97
C75	10.33 (1.38)	8.63 (2.27)	4.80 (1.19)	4.15 (1.53)	1.18 (0.21)	83.55
A100	11.61 (3.05)	13.17 (4.22)	7.69 (2.59)	3.75 (0.83)	1.80 (0.67)	143.05
B100	16.66 (6.89)	12.88 (4.06)	6.17 (2.62)	3.14 (1.32)	2.19 (0.81)	203.61
C100	13.81 (4.70)	10.76 (3.14)	6.66 (1.46)	3.33 (1.08)	2.02 (1.69)	161.10
A150	9.55 (1.64)	9.06 (1.31)	4.56 (1.49)	3.76 (1.56)	1.65 (0.63)	706.02
B150	8.60 (1.81)	7.00 (1.35)	4.39 (2.33)	2.78 (0.88)	1.68 (0.55)	1033.51
C150	9.86 (3.39)	9.56 (1.68)	4.41 (0.89)	3.10 (0.45)	1.56 (0.45)	743.29
Average	13.31 (3.13)	11.81 (2.64)	6.19 (1.90)	3.57 (1.17)	1.68 (0.69)	273.82

PR and MOILS-PRA. The frontier of these later methods were cover 20, 8 and 16% more solutions, respectively, than of the version MOILS-M. MOILS-PR was outperformed by MOILS-PRA and MOILS-PR-PRA, which are not different statistically at 5% of significance. Thus, the MOILS-PRA has better performance for the coverage metric. Once MOILS-PR does not get to improve this metric, and that the MOILS-

Table 5 Estimated difference in average between the row and column algorithms for the performance metrics

MOILS	Cardinality				Coverage				Hyper-volume			
	PR	PRA	M	MOILS	PR	PRA	M	MOILS	PR	PRA	M	MOILS
PRA-PR	n.s.	7.12	9.74	11.63	0.13	n.s.	0.20	0.44	0.004	n.s.	n.s.	0.054
PR		5.61	8.24	10.13		−0.08	0.08	0.31		−0.004	n.s.	0.049
PRA			2.62	4.50			0.16	0.39			0.005	0.054
M				1.89				0.23				0.049

Positive values indicate higher average value for the algorithm in the row
n.s. not statistically significant, *Obs.* only results significant at 95% of confidence level adjusted for multiple hypothesis testing using Bonferroni correction (Hochberg 1988) are shown

PRA took half of the time to return a good frontier, the MOILS-PRA is considered the best method for this metric.

The last metric to be evaluated is the hyper-volume (see Table 3). The statistical analyze showed that the MOILS-PR-PRA and MOILS-PR have no significant difference from MOILS-M. MOILS-PRA had a better performance than MOILS-M and MOILS-PR, with 0.5 and 0.4% bigger hyper-volume, respectively, and no significant difference from MOILS-PR-PRA, which in turn outperformed the MOILS-PR with 0.4% larger hyper-volume. The MOILS-PRA heuristic showed to be once again more interesting. The values obtained for this metrics confirm the above conclusion about the advantage got by MOILS-PRA to the coverage metric. With those results one can also state that the MOILS-PRA has also a better performance for the hyper-volume, which estimates both convergence and spread of the solutions.

The MOILS-PRA is considered the overall best proposed meta-heuristic. Even better than MOILS-PR-PRA because if one consider the three multi-objective metrics, which are coverage, spread and time, the former presented better performance for all of them. The better performance of MOILS-PRA over MOILS-PR can be explained by the fact that MOILS-PRA explores new empty spaces guided by the crowding distance procedure while MOILS-PR might be exploring a space of good solutions, close to the frontier, but already populated by others, not improving thus the hyper-volume or the coverage. Moreover, once the solution of the elite set for the MOILS-PR has to have 10% of different edges and the space of the extreme solution can be over crowded the solutions of the elite set might be not that good. Because if it space is over crowded, a solution with 10% of different edges can be far from the extreme one, so there is a likelihood of the search has been exploring poor paths.

4.4 Second experiment results: limiting the total running time

One might reason that the previous computational experiment was biased, because the methods with the "Walk" procedure took more time to perform the $\mathbb{H}_{max}$ iterations than the ones without the procedure. One might contemplate then how would the devised heuristics perform if they have the same amount of time to solve the instances, instead of limiting the number of iterations as it is actually done in an ILS framework. This second computational experiment limits thus the total running time (see Table 6) that each method can have to solve each instance and assess how the methods perform.

Table 6 Imposed time limits

Instance	Time limit [s]	Instance	Time limit [s]
A50	386	A100	1660
B50	385	B100	3392
C50	347	C100	2225
A75	846	A150	6745
B75	1414	B150	8892
C75	863	C150	7332

Table 7 Average (standard deviation) of cardinality

Instance	Cardinality					
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS	Smallest
A50	9.97 (1.10)	9.22 (1.33)	9.31 (0.80)	7.26 (1.35)	1.38 (0.24)	88
B50	12.99 (1.01)	12.05 (0.51)	11.80 (1.47)	11.98 (1.17)	2.11 (0.71)	71
C50	10.23 (0.75)	9.48 (1.42)	10.61 (1.46)	10.63 (1.75)	1.46 (0.38)	77
A75	13.00 (1.52)	12.61 (1.26)	14.80 (0.88)	14.36 (2.02)	1.44 (0.43)	97
B75	9.91 (0.90)	9.91 (1.02)	11.04 (1.49)	10.31 (1.86)	1.51 (0.46)	138
C75	20.34 (1.43)	19.77 (2.26)	22.83 (3.05)	20.70 (1.68)	2.42 (0.94)	60
A100	13.34 (2.62)	14.86 (2.76)	17.74 (2.31)	17.09 (2.27)	1.64 (0.52)	110
B100	12.49 (1.35)	13.25 (1.43)	14.36 (1.35)	13.89 (1.93)	1.33 (0.32)	153
C100	17.17 (2.86)	16.46 (1.84)	19.39 (1.78)	17.93 (2.37)	2.01 (0.69)	102
A150	45.95 (4.25)	39.28 (6.11)	45.65 (7.29)	43.31 (3.57)	1.58 (0.46)	70
B150	14.64 (2.50)	13.05 (1.52)	19.82 (1.59)	16.84 (1.83)	1.37 (0.43)	172
C150	25.14 (3.75)	22.60 (2.46)	25.21 (4.16)	23.90 (3.32)	1.38 (0.32)	118
Average	17.10 (2.00)	16.04 (1.99)	18.54 (2.30)	17.35 (2.09)	1.64 (0.49)	104.67

The times were chosen as the average running times for the MOILS-PR-PRA to solve the instances.

This experiment uses the same instances which are solved eight times by each method. Tables 7, 8 and 9 reports the attained results for the cardinality, coverage and hyper-volume metrics, respectively, while Table 10 summarizes the statistical analysis. Once again, the four enhanced MOILS heuristics (MOILS-M, MOILS-PR-PRA, MOILS-PR and MOILS-PRA) outperformed the standard MOILS from the literature. Observing the three metrics (cardinality, coverage, and hyper-volume), the MOILS-PRA variant had a slightly better performance than the other three versions.

These results confirm the results of the previous section: the "Walk" procedure can boost the insertion of non-dominated solutions in the frontier set when adjacent solutions of a solution with the largest crowding distance is used. Nevertheless, a remark is in order: the methods had a similar efficiency when only the hyper-volume metric is observed. As all methods, but the MOILS-PR-PRA variant, had more time to do more iterations, the number of non-dominated solutions was greatly augmented for all versions, which implies that the marginal contribution of each new non-dominated solution to the hyper-volume is very small (Tables 11, 12, 13, 14, 15).

4.5 Third experiment results: single loads

The third computational experiment uses twelve test problems extracted from Park et al. (2012). These instances have students varying within the set {250, 500} clustered around schools varying within the set {6, 12, 20}. The heuristics' performance were assessed under the perspective of having a dedicated fleet per school, i.e. the adoption of single loads on the buses. Though the heuristics were devised to work with the

Table 8 Average (standard deviation) of coverage

Instance	Coverage				
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS
A50	0.53 (0.12)	0.35 (0.16)	0.58 (0.20)	0.21 (0.09)	0.08 (0.02)
B50	0.54 (0.18)	0.53 (0.10)	0.38 (0.22)	0.31 (0.13)	0.08 (0.08)
C50	0.43 (0.12)	0.40 (0.16)	0.44 (0.20)	0.41 (0.09)	0.09 (0.02)
A75	0.37 (0.14)	0.33 (0.13)	0.57 (0.10)	0.43 (0.13)	0.05 (0.03)
B75	0.44 (0.07)	0.33 (0.16)	0.55 (0.16)	0.36 (0.09)	0.10 (0.07)
C75	0.42 (0.12)	0.36 (0.08)	0.66 (0.06)	0.34 (0.12)	0.05 (0.04)
A100	0.27 (0.16)	0.28 (0.10)	0.55 (0.08)	0.45 (0.11)	0.08 (0.04)
B100	0.31 (0.14)	0.27 (0.10)	0.63 (0.15)	0.46 (0.16)	0.06 (0.02)
C100	0.31 (0.13)	0.35 (0.19)	0.52 (0.12)	0.45 (0.10)	0.04 (0.03)
A150	0.39 (0.12)	0.22 (0.08)	0.55 (0.15)	0.42 (0.16)	0.02 (0.01)
B150	0.32 (0.09)	0.15 (0.03)	0.57 (0.09)	0.55 (0.14)	0.05 (0.02)
C150	0.32 (0.16)	0.25 (0.11)	0.51 (0.17)	0.52 (0.08)	0.05 (0.03)
Average	0.39 (0.13)	0.32 (0.12)	0.54 (0.14)	0.41 (0.12)	0.06 (0.03)

Table 9 Average (standard deviation) of hyper-volume

Instance	Hyper-volume				
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS
A50	0.93 (0.01)	0.92 (0.01)	0.93 (0.01)	0.91 (0.01)	0.89 (0.02)
B50	0.92 (0.01)	0.92 (0.01)	0.91 (0.01)	0.91 (0.01)	0.87 (0.02)
C50	0.92 (0.01)	0.92 (0.01)	0.92 (0.01)	0.92 (0.01)	0.89 (0.02)
A75	0.93 (0.01)	0.93 (0.01)	0.94 (0.01)	0.93 (0.01)	0.88 (0.03)
B75	0.92 (0.01)	0.92 (0.01)	0.93 (0.01)	0.92 (0.01)	0.88 (0.03)
C75	0.92 (0.01)	0.91 (0.01)	0.92 (0.01)	0.92 (0.01)	0.86 (0.02)
A100	0.90 (0.01)	0.90 (0.02)	0.91 (0.02)	0.91 (0.02)	0.88 (0.03)
B100	0.91 (0.01)	0.91 (0.01)	0.92 (0.01)	0.92 (0.01)	0.88 (0.01)
C100	0.78 (0.01)	0.79 (0.01)	0.80 (0.02)	0.79 (0.01)	0.76 (0.02)
A150	0.90 (0.02)	0.90 (0.02)	0.91 (0.01)	0.91 (0.02)	0.84 (0.04)
B150	0.91 (0.01)	0.90 (0.01)	0.92 (0.01)	0.92 (0.01)	0.88 (0.01)
C150	0.88 (0.02)	0.88 (0.01)	0.89 (0.02)	0.90 (0.01)	0.83 (0.04)
Average	0.90 (0.01)	0.90 (0.01)	0.91 (0.01)	0.90 (0.01)	0.86 (0.03)

concept of mixed loads, they had a computational performance similar to the first set of experiments (Sect. 4.3). The four enhanced variants once again outperformed the literature MOILS heuristic, but having the MOILS-PRA a slightly better overall performance than the others, when all three metrics were considered.

Table 10 Estimated difference in average between the row and column algorithms for the performance metrics

MOILS	Cardinality				Coverage				Hyper-volume			
	PR	PRA	M	MOILS	PR	PRA	M	MOILS	PR	PRA	M	MOILS
PRA-PR	n.s.	n.s.	n.s.	15.462	n.s.	−0.155	n.s.	0.323	n.s.	n.s.	n.s.	0.042
PR		−2.502	n.s.	14.409		−0.223	n.s.	0.255		−0.009	n.s.	0.039
PRA			1.197	16.911			0.133	0.478			n.s	0.048
M				15.714				0.345				0.045

Positive values indicate higher average value for the algorithm in the row
n.s. not statistically significant, *Obs.* only results significant at 95% of confidence level adjusted for multiple hypothesis testing using Bonferroni correction (Hochberg 1988) are shown

Table 11 Average (standard deviation) of cardinality

Instance	Cardinality					
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS	Smallest
CSCB01	6.90 (0.73)	7.18 (0.98)	5.22 (0.82)	4.14 (0.60)	1.30 (0.14)	62
CSCB02	11.84 (1.33)	12.85 (1.99)	9.82 (1.73)	7.26 (0.97)	1.32 (0.17)	42
CSCB09	15.35 (0.86)	13.99 (0.66)	11.12 (0.76)	4.91 (0.26)	1.65 (0.18)	27
CSCB10	24.51 (2.17)	25.44 (2.08)	20.11 (1.35)	18.44 (1.35)	1.96 (0.39)	49
CSCB17	15.47 (1.49)	15.49 (3.86)	9.71 (2.03)	7.96 (1.58)	1.12 (0.12)	73
CSCB19	21.88 (1.57)	21.48 (1.79)	17.99 (1.79)	15.83 (1.24)	1.70 (0.36)	48
CSCB03	17.52 (1.94)	14.47 (2.57)	10.22 (1.48)	10.21 (1.30)	1.35 (0.24)	51
CSCB04	12.21 (1.94)	11.02 (1.49)	9.27 (2.32)	8.46 (1.74)	1.31 (0.28)	83
CSCB11	12.07 (1.79)	9.97 (2.01)	7.04 (1.77)	7.03 (1.24)	0.93 (0.16)	74
CSCB12	18.73 (3.35)	20.62 (1.67)	14.94 (2.18)	12.19 (3.70)	1.20 (0.20)	92
CSCB19	37.93 (5.64)	32.55 (5.35)	25.97 (2.61)	20.63 (3.69)	1.58 (0.41)	54
CSCB20	24.85 (5.95)	21.96 (0.58)	15.16 (4.33)	13.08 (0.61)	1.16 (0.19)	85
Average	18.27 (2.40)	17.25 (2.09)	13.05 (1.93)	10.84 (1.52)	1.38 (0.24)	61.67

5 Final remarks

A multi-objective approach for the capacitated rural school bus routing problem with heterogeneous fleet and mixed loads was proposed. The problem considers not only costs, but the average weighted riding times, and distance balance among drivers. This is an important problem that has been neglected by the government and the literature. It is very common to have very short and very long routes in the solution set of paths, giving rise to labor complains by the drivers, and most likely exposing students to possible accidents due to drivers' weariness. Routing costs and travel times also play an important role and can not be disregarded. So a multi-objective optimization approach helps decision makers to solve these issues.

Table 12 Average (standard deviation) of coverage

Instance	Coverage				
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS
CSCB01	0.54 (0.15)	0.52 (0.11)	0.27 (0.08)	0.14 (0.11)	0.15 (0.05)
CSCB02	0.50 (0.21)	0.47 (0.08)	0.38 (0.19)	0.13 (0.09)	0.03 (0.03)
CSCB09	0.55 (0.15)	0.45 (0.11)	0.19 (0.06)	0.01 (0.01)	0.10 (0.06)
CSCB10	0.41 (0.12)	0.42 (0.11)	0.35 (0.07)	0.27 (0.07)	0.05 (0.03)
CSCB17	0.51 (0.13)	0.43 (0.14)	0.25 (0.14)	0.21 (0.09)	0.07 (0.02)
CSCB19	0.45 (0.10)	0.43 (0.14)	0.41 (0.18)	0.27 (0.10)	0.05 (0.02)
CSCB03	0.56 (0.17)	0.48 (0.13)	0.30 (0.09)	0.15 (0.08)	0.10 (0.03)
CSCB04	0.55 (0.12)	0.47 (0.05)	0.28 (0.09)	0.20 (0.07)	0.06 (0.02)
CSCB11	0.56 (0.07)	0.48 (0.08)	0.30 (0.07)	0.15 (0.05)	0.10 (0.02)
CSCB12	0.45 (0.16)	0.40 (0.17)	0.33 (0.17)	0.28 (0.16)	0.06 (0.02)
CSCB19	0.45 (0.09)	0.32 (0.10)	0.34 (0.09)	0.16 (0.12)	0.05 (0.03)
CSCB20	0.47 (0.12)	0.47 (0.22)	0.35 (0.09)	0.16 (0.03)	0.07 (0.03)
Average	0.50 (0.13)	0.45 (0.12)	0.31 (0.11)	0.18 (0.08)	0.07 (0.03)

Table 13 Average (standard deviation) of hyper-volume

Instance	Hyper-volume				
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS
CSCB01	0.87 (0.01)	0.87 (0.01)	0.85 (0.03)	0.84 (0.03)	0.84 (0.03)
CSCB02	0.86 (0.01)	0.85 (0.02)	0.85 (0.03)	0.86 (0.01)	0.80 (0.01)
CSCB09	0.89 (0.01)	0.89 (0.00)	0.88 (0.01)	0.85 (0.02)	0.85 (0.01)
CSCB10	0.91 (0.01)	0.90 (0.01)	0.90 (0.01)	0.90 (0.01)	0.87 (0.01)
CSCB17	0.88 (0.02)	0.88 (0.01)	0.87 (0.02)	0.87 (0.02)	0.82 (0.02)
CSCB18	0.90 (0.01)	0.90 (0.01)	0.90 (0.01)	0.90 (0.01)	0.86 (0.02)
CSCB03	0.86 (0.01)	0.86 (0.01)	0.85 (0.01)	0.84 (0.01)	0.81 (0.02)
CSCB04	0.82 (0.02)	0.81 (0.01)	0.81 (0.02)	0.80 (0.01)	0.77 (0.01)
CSCB11	0.86 (0.00)	0.86 (0.01)	0.85 (0.01)	0.84 (0.01)	0.81 (0.01)
CSCB12	0.88 (0.01)	0.88 (0.01)	0.88 (0.01)	0.87 (0.01)	0.82 (0.01)
CSCB19	0.88 (0.02)	0.88 (0.03)	0.88 (0.01)	0.86 (0.01)	0.81 (0.03)
CSCB20	0.89 (0.01)	0.89 (0.01)	0.88 (0.01)	0.87 (0.01)	0.84 (0.01)
Average	0.88 (0.01)	0.87 (0.01)	0.87 (0.01)	0.86 (0.01)	0.82 (0.02)

Four multi-objective iterated local search heuristics (MOILS-M, MOILS-PR, MOILS-PRA, MOILS-PR-PRA) were devised and used to solve three different computational experiments. All of the proposed heuristics outperformed the literature framework (Assis et al. 2013). The heuristic MOILS-PRA presented the overall best results in the studied metrics followed closely by the MOILS-PR-PRA variant. Further, the proposed heuristics have enhancements that can easily be extended to other

Table 14 Average (standard deviation) of time

Instance	Time					Smallest
	MOILS-PR-PRA	MOILS-PR	MOILS-PRA	MOILS-M	MOILS	
CSCB01	11.99 (1.41)	13.16 (1.43)	3.24 (0.93)	1.83 (0.63)	1.21 (0.12)	5.93
CSCB02	18.57 (2.71)	18.07 (1.74)	6.97 (2.82)	2.34 (0.70)	1.14 (0.11)	3.97
CSCB09	19.69 (1.58)	18.18 (0.75)	4.20 (1.28)	1.10 (0.07)	1.14 (0.12)	3.61
CSCB10	16.85 (2.11)	17.19 (1.51)	6.28 (1.11)	4.95 (1.11)	1.07 (0.09)	37.62
CSCB17	17.92 (2.82)	17.11 (3.53)	4.83 (1.81)	2.83 (0.98)	1.11 (0.10)	31.95
CSCB18	19.39 (4.17)	19.68 (2.96)	8.97 (2.75)	6.57 (1.87)	1.30 (0.28)	27.47
CSCB03	19.34 (3.07)	19.00 (3.25)	5.93 (1.87)	3.17 (0.76)	1.12 (0.09)	40.68
CSCB04	16.77 (3.33)	14.88 (1.57)	5.93 (2.34)	3.81 (0.84)	1.12 (0.12)	47.25
CSCB11	14.96 (1.74)	14.69 (1.57)	4.59 (1.60)	2.45 (0.49)	0.86 (0.07)	52.60
CSCB12	14.17 (1.06)	14.06 (0.52)	4.67 (0.71)	2.73 (0.65)	1.14 (0.15)	386.11
CSCB19	12.39 (2.37)	11.16 (2.01)	4.56 (0.79)	2.32 (0.68)	1.08 (0.10)	405.94
CSCB20	13.76 (0.76)	13.79 (0.48)	4.98 (1.33)	2.75 (0.13)	1.19 (0.14)	459.13
Average	16.32 (2.26)	15.91 (1.78)	5.43 (1.61)	3.07 (0.74)	1.12 (0.12)	125.19

Table 15 Estimated difference in average performance between the row and column algorithms for the performance metrics

MOILS	Cardinality				Coverage				Hyper-volume			
	PR	PRA	M	MOILS	PR	PRA	M	MOILS	PR	PRA	M	MOILS
PRA-PR	n.s.	5.224	7.427	16.890	0.128	n.s	0.203	0.436	0.004	n.s.	n.s.	0.054
PR		4.204	6.407	15.870		n.s.	n.s.	0.308		n.s.	n.s.	0.049
PRA			2.203	11.666			0.156	0.389			0.005	0.054
M				9.463				0.233				0.049

Positive values indicate higher average value for the algorithm in the row
n.s. not statistically significant, *Obs.* only results significant at 95% of confidence level adjusted for multiple hypothesis testing using Bonferroni correction (Hochberg 1988) are shown

multi-objective problems, bringing great improvements to the frontier quality. Several issues still remain open for future research. Lower bounds can be generated for each objective, and used to guide in the construction of the frontier set, besides providing an assessment of solutions' quality. The incorporation of school location decisions combined with the bus routing problem may lead to great saving costs for rural areas. Another especial issue for those areas is the possibility of creating transshipment points. Transshipment allows smaller vehicles to reach areas that larger buses can not. Driving the students from home to these points would allow higher capacity buses to do last part of the route until the school.

Acknowledgements This work was supported by the state government of Minas Gerais, by the National Fund for Education Development (FNDE), and by Foundation of Coordination for the Improvement of Higher Education Personnel (CAPES) of Brazil, and by the Research Foundation of Minas Gerais

(FAPEMIG)—Grant # APQ-02508-13—and the Brazilian National Council for Scientific and Technological Development (CNPq).

References

Assis LP, Maravilha AL, Vivas A, Campelo F, Ramírez JA (2013) Multiobjective vehicle routing problem with fixed delivery and optional collections. Optim Lett 7(7):1419–1431

Batista LS, Campelo F, Guimarães FG, Ramírez JA (2011) Pareto cone ε-dominance: improving convergence and diversity in multiobjective evolutionary algorithms. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, pp 76–90

Beume N, Fonseca CM, López-Ibáñez M, Paquete L, Vahrenhold J (2009) On the complexity of computing the hypervolume indicator. IEEE Trans Evol Comput 13(5):1075–1082

Bodin LD, Berman L (1979) Routing and scheduling of school buses by computer. Transp Sci 2(13):113–129

Bowerman R, Hall B, Calamai P (1995) A multi-objective optimization approach to urban school bus routing: formulation and solution method. Transp Res Part A 29(2):107–123

Braca J, Bramel J, Posner B, Simchi-Levi D (1997) A computerized approach to the New York City school bus routing problem. IIE Trans 29:693–702

Bradstreet L (2011) The hypervolume indicator for multi-objective optimisation: calculation and use. University of Western Australia

Carvalho WL, Cruz ROM, Câmara MT, Aragão JJG (2010) Rural school transportation in emerging countries: the Brazilian case. Res Transp Econ 29:401–409

Chen DS, Kallsen HA (1988) School bus routing and scheduling: as expert system approach. Comput Ind Eng 15:179–183

Clark G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 12(4):568–581

Corberán A, Fernández E, Laguna M, Martí R (2002) Heuristic solutions to the problem of routing school buses with multiple objectives. J Oper Res Soc 53:427–435

Deb K (2001) Multi-objective Optimization Using Evolutionary Algorithms. Wiley, London (ISBN 9780471873396)

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 6(2):182–197

Gendreau M, Laporte G, Potvin J-Y (2001) Metaheuristics for the capacitated VRP. In: Toth P, Vigo D (eds) The vehicle routing problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA 129–154

Gillett BE, Miller LR (1974) A heuristic algorithm for the vehicle-dispatch problem. Oper Res 22(2):340–349

Glover F, Laguna M, Martí R (2000) Fundamentals of scatter search and path relinking. Control Cybern 39:653–684

Golden B, Assad A, Levy L, Gheysens F (1984) The fleet size and mix vehicle routing problem. Comput Oper Res 11(1):49–66

Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130(3):449–467

Hochberg Y (1988) A sharper bonferroni procedure for multiple tests of significance. Biometrika 75(4):800–802

INEP (2013) School census—2013. http://portal.inep.gov.br/basica-censo (**in Portuguese**)

Laporte G, Semet F (2002) 5. Classical heuristics for the capacitated VRP, Chapter 5, pp 109–128

Li L, Fu Z (2002) The school bus routing problem: a case study. J Oper Res Soc 53:552–558

Li F, Golden B, Wasil S (2007) The open vehicle routing problem: algorithms, large-scale test problems, and computational results. Comput Oper Res 34:2918–2930

Lima FMS, Pereira DS, Conceio SV, Nunes NTR (2016) A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context. Expert Syst Appl 56:320–334

Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. Struct Multidiscip Optim 41(6):853–862

Martí R, Campos V, Resende MG, Duarte A (2011) Multi-objective grasp with path-relinking. AT&T Labs Research Technical Report

Pacheco J, Martí R (2006) Tabu search for a multi-objective routing problem. J Oper Res Soc 57:29–37

Pacheco J, Caballero R, Laguna M, Molina J (2013) Bi-objective bus routing: an application to school buses in rural areas. Transp Sci 47(3):397–411

Pareto V (1896) Cours d'économie politique professé a l'université de Lausanne, volume I. Librairie de l'Universit

Pareto V (1897) Cours d'économie politique professé a l'université de Lausanne, volume II. Librairie de l'Universit

Park J, Kim BI (2010) The school bus routing problem: a review. Eur J Oper Res 202:311–319

Park J, Tae H, Kim BI (2012) A post-improvement procedure for the mixed load school bus routing problem. Eur J Oper Res 217:204–213

Salkind Neil J (2006) Encyclopedia of measurement and statistics. Sage Publications

Schittekat P, Sevaux M, Sorensen K (2006) A mathematical formulation for a school bus routing problem. In: 2006 international conference on service systems and service management, vol 2. IEEE, pp 1552–1557

Thangiah SR, Nygard KE (1992) School bus routing using genetic algorithm. In: Proceedings of the SPIE conference on applications of artificial intelligence, X: Knowledge Based Systems, Orlando, Florida, pp 387–398

Vincent S (Ed.) (1999) The multigrade classroom: a resource handbook for small, rural schools, volume 1. Northwest Regional Education Laboratory, Book 1: Review of the Research on Multigrade Instruction

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evolut Comput 3(4):257–271