A partial evaluation approach for the School Bus Routing Problem

Ana Camila Pérez *, Eduardo Sánchez-Ansola a, Alejandro Rosete b, Omar Rojas b,c, Guillermo Sosa-Gómez b,*

* Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE), Facultad de Informática, 19390, Habana, Cuba
b Universidad Panamericana, Facultad de Ciencias Económicas y Empresariales, Álvaro del Portal 49, Zapopan, 45010, Jalisco, Mexico
c Faculty of Economics and Business, Universitas Airlangga, Surabaya, Indonesia

A R T I C L E   I N F O

Keywords:
Metaheuristics
Optimization
Partial evaluation
SBRP

A B S T R A C T

Several real-life optimization problems, such as the case of several instances of the School Bus Routing Problem (SBRP), are very complex and expensive to solve with exact algorithms. Metaheuristics are a good alternative in these situations because they are capable of generating good quality solutions to these problems in a reasonable time. Metaheuristics iterate thousands of times by introducing changes concerning the previous solutions. Each new solution must be evaluated, and sometimes, the new solutions have elements unchanged that are unnecessarily re-evaluated. However, an approach avoids repeatedly evaluating parts of different solutions known as partial evaluation. This work applies this technique to the SBRP to reduce its execution time. To apply the partial evaluation approach in this problem, each solution contains the information of the change that was made concerning the solution from which it originates. With this information, when evaluating the objective function, it will be only necessary to analyze the routes that changed. In the literature reviewed, no previous work was found in which the partial evaluation approach has been applied in the context of SBRP. In this paper we apply it in order to reduce the computational cost of SBRP solutions based on metaheuristics. The results show that it is possible to decrease the execution time in 80% of the instances, reducing the execution time on average by 73.6%.

1. Introduction

The school bus routing problem (SBRP) has its origin in schools that have a fleet of buses and must transport a group of students to schools [1]. One of the most studied objective of SBRP is to efficiently plan school bus routes to transport students from stops near their homes to schools in a way that the total distances of the routes are minimized. This is the objective of the SBRP model studied in this article. However, it is worth mentioning that there are other objectives for the SBRP, such as minimizing the number of buses used, minimizing the student walking distance, minimizing the total bus travel distance and/or time, and others. These problems usually have constraints that must be guaranteed in the solution such as vehicle capacity, maximum riding time, maximum walking distance, and others [2, 3].

The SBRP is a combinatorial optimization problem that, due to its complexity, is classified as NP-Complete [4]. Metaheuristics are widely used for solving these problems. Some of the most used metaheuristics are Ant Colony, Genetic Algorithm, GRASP, Tabu Search and Stochastic Hill Climbing. They have been successfully used in several cases, finding good solutions in a reasonable time [5, 6, 7]. Metaheuristics, in the search for solutions, must evaluate the objective function many times (hundreds, thousands of times), each one with different configurations of the solution [8]. A large number of evaluations of the objective function in an optimization problem increases the computational cost [9]. A large number of evaluated solutions are created as a result of transformations of previous solutions with the use of mutation operators [10]. This implies that when a solution results from the mutation of another solution, they will have some parts in common, which will be evaluated repeatedly (in each of the solutions that have it in common). There are approaches to obtaining substitute functions for objective function that tend to lower computational costs. One of these approaches is partial evaluation (PE), which focuses on the variations introduced by the operators [8, 9]. Thus, in each evaluation of the objective function, only the modified part of the solution is taken into account, taking advantage of previous evaluations [8].
This work aims to apply the partial evaluation approach to the school bus routing problem (SBRP), seeking to reduce its execution time. The SBRP being studied is based on the model presented in [10], which is explained in detail in the following section. This model is designed for situations where there is a homogeneous fleet, students do not require differentiated treatments among themselves, and all students are destined for a single school. The focus of the paper is on reducing the computational cost of the metaheuristic approach for solving SBRP based on the use of partial evaluation. The advantage of the proposal is experimentally demonstrated by solving several SBRP instances by using Stochastic Hill Climbing metaheuristic implemented in the BiCIAM library [11].

2. School Bus Routing Problem

2.1. Literature review

The School Bus Routing Problem is a classic combinatorial optimization problem [12]. Due to its complexity, it is classified as NP-Complete and it is a specific type of vehicle routing problem (VRP) [4]. This group has the characteristic that, despite being easily described, they are pretty hard to solve. This can be done through exact mathematical methods, but the best solution is not obtained in polynomial time for large instances [6].

SBRP provides efficient school bus routes, where buses must pick up students at various stops and transport them to each student’s designated school. A route or a path is a sequence of bus stops that a bus must go through. Each bus is responsible for traveling a route. In the solutions provided to the problem, it is necessary to comply with restrictions such as the capacity of the buses, the maximum time that students can be traveling on the bus, and the periods of the schools, among others [1, 2]. SBRP can be separated into five subproblems [2]: bus stop selection, bus route generation, bus route schedule, school bell time adjustment, and strategic transportation policy.

Based on [2, 3] the SBRP can be categorized according to the following aspects:

- Algorithm: refers to the algorithm solution used or implemented in each research.
- Schools (s): multi school (MD) or single school (SD).
- Fleet: homogeneous (HM) or heterogeneous (HT).
- Objective (Obj.): minimize the number of buses (N), minimize the total bus distance (TBD).
- Special students (SS): yes or no.
- Mixed load (ML): yes or no.
- Problems: bus stop selection (BSS), bus route generation (BRG), bus route scheduling (BRS), school bell adjustment (SBA).

The main literature of SBRP that is more related to the proposal presented in this paper is summarized in Table 1. An extended review of SBRP may be found in [2].

| Ref. | Algorithm | S | Fleet | Obj. | SS | ML | Problems |
|------|-----------|---|-------|------|----|----|----------|
| [13] | MD        | 0 | MD    | N    | No | Yes|          |
| [14] | GRASP     | SD | HM    | TBD  | No | No | BSS,BRG  |
| [10] | Hill Climbing | SD | HM    | TBD  | No | No | BSS,BRG  |
| [12] | ILS       | MD | HT    | N    | TBD| Yes| BSS,BRG, BRS |
| [5]  | Genetic Algorithm | MD | HM    | N, TBD| Yes| Yes| BSS,BRG, BRS |

Metaheuristics are widely used to solve combinatorial optimization problems, such as SBRP [13] as it is shown in Table 1. Studies have demonstrated the ability of these methods to approximately solve complex optimization problems in a reasonable computational time [15]. Metaheuristics, despite being approximate methods that do not guarantee an optimal solution, they allow for good solutions to be reached in a reasonable time [6]. However, metaheuristics tend to have a high computational cost that is directly related to the number of times the objective function of the problem must be evaluated [9].

Surrogate functions are a way to solve the problem of the computational cost of metaheuristics with less expensive alternatives when evaluating the objective function and/or the restrictions [16]. These functions use several techniques and can be an exact or approximate variant of the objective function and/or constraints [9].

One of the exact approaches to surrogate functions is partial evaluation, which is applied in this work to reduce the computational cost of the metaheuristic approach. Partial evaluation is based on the principle of evaluating only the part of the solution that was modified, and it has obtained good results in NP-Complete optimization problems [9].

The effect of partial evaluation is greatly influenced by the operators that are applied for the mutation of solutions [8]. Table 2 summarizes the application a partial evaluation to different optimization problems like Maximal Covering Location Problems (MCLP), Rank Aggregation Problems (Ranking) and Constrained Optimization Problems (COP). Some examples of SBRP applications are included.

It is important to clarify that the concept of Partial Evaluation has been successfully applied in other optimization contexts but this is not the case of SBRP in spite of its possible impact in the reduction of computational cost.

2.2. Mathematical model

This section presents a mathematical formulation of the school bus routing problem presented in [10]. To apply the designed model to the planning of school bus routes, the problem must have the following characteristics: a single school that is the destination of all students; all students are equal and they do not require special treatment; all buses have the same capacity that is fixed, the first and last bus stop is the school.

The proposed model is presented below. The input variables are:

- $c$: capacity of buses
- $b$: number of buses
- $d$: maximum distance students can walk to reach a stop
- $P$: set of possible stops, being the index identifying the school.
- $E$: student set
- $C^s$: set of vectors with coordinate pairs of possible stops
- $C^s = \{ (\hat{x}^s_1, \hat{y}^s_1), \ldots, (\hat{x}^s_{|C^s|}, \hat{y}^s_{|C^s|}) \}$ where $\hat{e}^s = \langle \hat{x}^s, \hat{y}^s \rangle$
- $\hat{x}^s_j$: $x$ coordinate of the stop $p$
- $\hat{y}^s_j$: $y$ coordinate of the stop $p$
- $C^c$: vector set with coordinate pairs of each student’s house
- $C^c = \{ (\hat{x}^c_1, \hat{y}^c_1), \ldots, (\hat{x}^c_{|C^c|}, \hat{y}^c_{|C^c|}) \}$ where $\hat{e}^c = \langle \hat{x}^c, \hat{y}^c \rangle$
- $\hat{x}^c_e$: $x$ coordinate of the student $e$
- $\hat{y}^c_e$: $y$ coordinate of the student $e$
- $W_{ij}$: cost matrix between each pair of stops $(i,j)$
\( W_{ij} = \begin{cases} D(c_i^e, c_j^e), & i \neq j \\ 0, & i = j \end{cases} \quad \forall i, j \in P \)

\( D \): defines the distance function between any two nodes. In this article, the Euclidean distance was used as \( D \).

\( S_{ij}^e \): binary matrix with 1 if the student \( e \) can reach the stop \( p_i \), and 0 otherwise.

\( S_{ij}^p \)\( \leq 1, D(c_i^e, c_j^e) \leq d \quad e \in E, p \in P \) - \( \{ p_0 \} \)

Decision variables:

\( R_k^{m} \): stop that is visited by the route \( k \) in order \( m \)

With \( R_k^{m} = R_k^{m+} = 0 \) where \( P_k \): size of the route \( k \), including origin and destination, which is the school.

\( Z_e \): stop where each student \( e \) is picked up.

Objective function

\[
\text{Minimize} \sum_{k=1}^{b} \sum_{m=1}^{k} W[R_{km} | R_{km+}] \tag{1}
\]

Constraints

\[
\begin{align*}
|R_{km} | & \leq 1 \quad \forall p \in P - \{ p_0 \}, \forall m \in [1, ... |P|], \forall k \in [1, ... b] \\
(c, p) | Z_e = p & \subseteq (c, p) | S_{ij}^e = 1 \quad e \in E, p \in P \tag{2} \\
\{E | m | R_{km} = Z_e \} & \subseteq m \in [1, ... |P|], \forall k \in [1, ... b], \forall e \in E \tag{3} \\
\{R_{km} | Z_e = R_{km} \} & = 1 \quad \forall e \in E \tag{4}
\end{align*}
\]

The objective function (1) minimizes the total distance traveled by the bus fleet. Equations (2), (3), (4), and (5) are constraints that must be met for the solution to be feasible. Constraint (2) ensures that each stop is visited at most once, except for the stop representing the school, the final destination of all buses. Constraint (3) ensures that each student can reach the stop to which he was assigned to take the bus. Constraint (4) takes into account that the capacity of each bus is not exceeded. Constraint (5) ensures that each stop assigned to at least one student is visited by a bus.

3. Metaheuristic approach for SBRP with partial evaluation

3.1. Solution representation example

In Table 3, a possible solution of the previously described model is represented. In this solution, there are five routes. For route 1(R1), four stops must be visited in the same order as shown in the solution. Each route starts and ends at the school (Route 1: School -> 17 -> 3 -> 12 -> 14 -> School). The other component of the representation of the solution appears in the Table 4. This structure represents the assignment of the students to the stops. Every student must be assigned to one stop.

3.2. Components of the metaheuristic approach

As in previous papers [9], here we illustrate the partial evaluation approach for SBRP by using Stochastic Hill Climbing as the metaheuristic method. The solution is implemented using BiCIAM [11], a library of metaheuristic algorithms that were previously used in [10].

The initial construction combines two heuristic algorithms, one for assigning each student to a bus stop and the other to create an initial collection of routes. The assignment algorithm, named StudentToFirst-Stop, is based on the following idea. For each student a set of stops is selected. These are the stops to which the student can be assigned in compliance with the maximum distance constraint between the home of each student and the stop. Stops are sorted arbitrarily based on the order of the stops in the description of the instance. Then, the first stops in the set are selected and it is assigned to the student. If the selected stop has been already assigned to a number of students equal to the capacity of the vehicles, a new attempt is made in order to insert the student to one of the other stops in the set. This is done by selecting the next stop in the list, and always ensuring that the bus capacity is not exceeded by the number of assigned students.

The algorithm to route the stops, named CompletingFirstRoute, is based on the following idea. For each route a set of stops is selected, containing all those stops where at least one student was assigned and containing a number of students that can still be inserted to the current route without exceeding the capacity of the vehicles. The stops are randomly selected to complete the route, and the set is updated according to the availability of the route's capacity. Once the set is empty a new route must be created, which is filled following the same procedure. If, when satisficing the capacity of the last route, there are still stops to be located, they are incorporated into the last route and this solution is considered unfeasible.

After the construction of the initial solution following the previous heuristics, the local search goes on by trying to improve the quality of the solutions by using mutations. For the selection of the mutation to be applied, a probabilistic adaptive mechanism is applied. It is adaptive because the selection of the mutation to be applied is based on the previous performance of the mutations, i.e. each mutation operator is selected in a probabilistic way according to the result of its previous use. Each operator has a weight, and the probability of selection it in each iteration is directly proportional to its weight. Then, in each iteration the selected operator is applied to the current solution, and depending on the result obtained, two operations are carried out: the operator selected is penalized for its use, and it is rewarded if it obtained a better solution than the previous one. With these operations, the weights are modified, and therefore the selection probabilities. Thus, in each iteration where an operator produced and improved solution, then it is more likely to be selected. In general, an operator that achieves improvements is probably more selected rather than another that produces less improvements.

The mutation operators that compete to be selected by the selection mechanism are TWO_SWAP, SECTION_SWAP, and TWO_OPT.

- **TWO_SWAP** operator [17]: swap two bus stops. Two points can be selected from two different routes or the same route. To implement that, two routes \( R_k \) and \( R_l \) are randomly selected. From each of them, a point is randomly chosen. Finally, both points \( p_k \) and \( p_l \) are swapped. The next two figures show new solutions generated by **TWO_SWAP** operator. Fig. 1 represents the application of this operator in one route, and Fig. 2 represents the application of two different routes.

- **SECTION_SWAP** operator [17]: The objective of applying this operator is to exchange two substrings of nodes. Both substrings must be equal in size. For the implementation of the operator, initially, two routes are randomly selected, \( R_k \) and \( R_l \), guaranteeing that \( R_k \neq R_l \). The length of the substrings is then randomly selected. Subsequently, it is defined in which position the substring will be

| Table 3. Representation of the routes. |
|---------------------------------------|
| R1 | 17 | 3 | 12 | 14 |
| R2 | 5  | 13 | 2  |
| R3 | 16 | 18 | 9  | 8 |
| R4 | 6  | 11 | 4  | 1 |
| R5 | 7  | 15 | 10 |
| Original Solution | Transformed Solution |

Fig. 1. Example of TWO_SWAP operator to route a) Original solution, b) Transformed solution.
inserted in each route. Finally, each of the elements is exchanged between both substrings. Fig. 3 show mutations generated by SECTION_SWAP operator.

- **TWO_OPT operator** [17]: The result of applying this operator is the exchange of a substring in a route. Initially, a route R is randomly chosen. Then two nodes of the selected route are randomly chosen, \( p_{R_1} \) and \( p_{R_2} \), in such a way that \( 1 \leq p_{R_1} < p_{R_2} \leq |R| \). Finally, the string \( p_{R_1}, ..., p_{R_2} \) belonging to R is updated. The Fig. 4 show mutations generated by TWO_OPT operator.

The previously described initial construction heuristics guarantee compliance with the constraints of the problem. Likewise, the mutation operators take these restrictions into account and guarantee to introduce changes that do not compromise the feasibility of the solution in each iteration. It is worth noting that the implemented operators affect at most two routes. In general, the search is conducted following the Hill Climbing strategy, i.e. every new mutated solution is compared to the current solution and it is accepted as the future current solution if the new one is better or equal than the previous one.

### 3.3. Partial evaluation

The partial evaluation can be applied to the evaluation of objective functions and to the evaluation of constraints. In this work, the partial evaluation is applied only to the objective function of the optimization model, because the constraint satisfaction is guaranteed by the operators. Algorithm 1 shows the pseudo-code of the objective function of the original model. In its operation, it is iterated through the variables of the model that represents the planning of the school routes. When the partial evaluation is not applied, the method iterates through every planning route. Each stop in each route must also be iterated, and the total cost of the solution increases according to the solution represented.

#### Algorithm 1 Total Evaluation in SBRP.

```plaintext
1: cost = 0
2: currentSolution
3: for each route k of currentSolution do
4: for each stop p of currentSolution(k) do
5: cost ← add travel cost from currentSolution(k)[p] to currentSolution(k)[p + 1]
6: end for
7: end for
```

It can be seen that the applied mutation operators affect at most two routes. Due to this, in a solution like the one represented in Table 3, each mutation affects at most two routes. However, in Algorithm 1 it is necessary to reevaluate the five routes.

The partial evaluation solves this problem because each new solution generated only evaluates the affected routes concerning the original solution. Algorithm 2 shows how the new objective function works with partial evaluation. For the implementation of the partial evaluation, each solution must keep some extra information such as the modified routes and the evaluation of the previous solution. In this way, the cost of the modified routes before mutation is subtracted from the previous evaluation. Then the cost of the modified routes in the current solution is added to the previous evaluation. When these two operations are executed, the cost of the current solution is obtained. The solutions that were not the result of a mutation operator are completely evaluated, as it is the case of the initial solution in each execution of the metaheuristic.

#### Algorithm 2 Partial Evaluation is SBRP.

```plaintext
1: previousSolution
2: currentSolution
3: modifiedRoutes ← routes that changed between the previous solution and the current one
4: if modifiedRoutes is not empty then
5: cost ← cost of previousSolution
6: for each route k of modifiedRoutes do
7: for each stop p of previousSolution(k) do
8: cost ← subtract travel cost from previousSolution(k)[p] to previousSolution(k)[p + 1]
9: end for
10: end for
11: for each route k of modifiedRoutes do
12: for each stop p of currentSolution(k) do
13: cost ← add travel cost from currentSolution(k)[p] to currentSolution(k)[p + 1]
14: end for
15: end for
16: else
17: cost ← 0
18: for each route k of currentSolution do
19: for each stop p of currentSolution(k) do
20: cost ← add travel cost from currentSolution(k)[p] to currentSolution(k)[p + 1]
21: end for
22: end for
23: end if
```

The changes introduced for the implementation of the partial evaluation are related only to the evaluation of the solutions. The initial solution construction algorithms, as well as the mutation operators,
were not modified. This ensures that the results obtained by the problem are not affected.

It is worth noting that, as can be seen in Algorithm 2, the partial evaluation must do a double evaluation of each of the routes that were modified by the operator in order to subtract the cost of the modified routes in the previous solutions and to add the cost of the modified routes in the new solution. However the total evaluation, Algorithm 1, for the modified routes, each route are only evaluated once. Thus the advantage of the partial evaluation is more noticeable when there are many routes. In large instances with more routes, it can expect a cost gain from partial evaluation. But in small instances with two routes or less, the partial evaluation will be generating an extra cost with respect to total evaluation.

4. Experiments

4.1. Instances description

The experiments presented here are carried out on a set of 112 instances that were generated in [14] and used in [10] for experimentation. Instances vary from the simplest one with 5 stops and 25 students to the most complex instances where 800 students must be located at 80 stops. The instances have a homogeneous fleet composed by buses with a capacity of 25 or 50. The maximum distance that each student can travel to reach the assigned stop is also a parameter defined in each instance. Table 5 summarizes the characteristics of the instances of the test set.

The experiments presented in this article were executed on a laptop with the following characteristics: Intel Core i5-5200U processor, 8 GB of RAM and Windows 10 operating system. The solution was developed in Java and it was executed and compiled in IntelliJ IDEA.

4.2. Analysis of the results

Table 6 presents the results of the experiments. In this table, we include the execution time (columns T) and the quality (columns Q) of the solution. The parameter Q is the cost of the objective function and it is expressed in distance units to this model. In the Table 6 are included the results of the Algorithm 2 with partial evaluation (PE) and Algorithm 1 with total evaluation (TE), results from [10]). In order to allow comparison, we also included the results (T and Q) of the GRASP algorithm presented in [14] in columns with T(G) and Q(G). In addition, the quality of the optimal solutions obtained by an exact method [14] is presented in column Q(O) when it is known. The first column (I) indicates the identifier of each instance.

The reduction of the execution time due to the partial evaluation is graphically presented in Fig. 5. The X-axis of Fig. 5 indicates the identifiers of each instance (I) in ascending order of its complexity. The Y-axis represents the percent (%) that represents the values in column T(PE) with respect to values in column T(TE) of the Table 6.

Fig. 5 shows that there are a small number of instances in which the application of the partial evaluation worsens the execution time with respect to the total evaluation. These instances represent 20% of the set of instances analyzed. This percentage corresponds to the less complex instances. However, in 80% of the instances PE is faster than TE.

As the operators affect at most two paths and, as it was detailed in the description of the partial evaluation in Algorithm 2, at most 4
It is worth noting that as routes as considered as a unit with respect to partial evaluation, there are paths inside each route that are unnecessarily reevaluated. This is an opportunity to take advantage of the use of partial evaluation even with a reduced number of routes. From this analysis, the first recommendation for future work is derived and it is to implement partial evaluation in such a way that evaluate only specific paths of the routes that have changed. In this way, only the paths with altered stops should be evaluated and not the altered routes.

In 20% of the instances, the cost of implementing PE is greater than the cost of implementing TE. For these instances, on average, the partial evaluation is 60.66% times greater than the total evaluation. On the other hand, in 80% of the instances, shorter times are obtained with the partial evaluation, which corresponds to the larger instances in which time-saving is more important. For these cases, the partial evaluation on average is carried out in 26.4% of the time of the total evaluation. Notably, in one instance it is only used 4% of the time of the total evaluation. In other words, with the implementation of the partial evaluation, an average of all 112 instances of 73.6% of the time consumed with the total evaluation is saved.

Table 6, contains in column T(G) the execution time obtained by the GRASP algorithm presented in [14]. Figs. 6 and 7 presents a comparison of PE and TE with respect to this GRASP approach. In both figures, the y-coordinate of each orange dot represents the ratio Q(TE)/Q(G) in one instance while the x-coordinate of this orange dot represents the ratio T(TE)/T(G) in this instance. On the other hand, the y-coordinate of each blue dot represents the ratio Q(PE)/Q(G) in one instance while the x-coordinate of this blue dot represents the ratio T(PE)/T(G) in this instance. For example, the last point (right) in Fig. 6 represents an instance where TE is around 4 times slower than G with a similar quality. It is worth noting. As the quality of TE and PE are the same, there is a pair of orange-blue points with the same quality (y-axis) with different execution time (corresponding to the results of PE and TE in the same instance).

Doing a comparison between the GRASP algorithm execution time versus PE algorithm execution time the following results are obtained:
1. In 61.6% of the 112 instances (69 instances) a better or equal time execution was reached with PE with respect to GRASP. For these instances, PE algorithm in average, represents the 15% of time consumption with respect to GRASP algorithm (6.7 times faster). Fig. 6 represents these 69 instances and compares the relation quality/time for the result PE and TE with respect to GRASP. The x-coordinates of most blue points are less than 0.3, indicating that (in general) the solutions with PE are 3 times faster, even with similar quality in several cases.
2. In 38.4% of the 112 instances (43 instances) a better execution time is reached by GRASP with respect to PE. For these instances, PE algorithm in average, is 9 times slower than GRASP. Fig. 7 represents these 43 instances and compares the relation quality/time for the result PE and TE respect GRASP.

paths must be evaluated. However in the total evaluation 1, all routes must be evaluated, so this number increases in the largest instances. Consequently, in cases where the complete solution has 4 routes or less, the total evaluation may have the same cost or even less than the partial evaluation. This explains the results presented in Fig. 5.
In the instances represented in Fig. 7, the PE solution is not competitive with respect to GRASP algorithm. These instances are less complex and in 17 of these instances (39.5%) both PE and GRASP obtain the optimal result.

Based on the experimental results, it can be seen that the application of the concept of partial evaluation in SBRP allow saving a considerable computational effort, this advantage is remarkable in the largest instances.

5. Conclusions

In this work, the objective of implementing the partial evaluation for the SBRP and reducing the execution time has been accomplished. After the completion of the work, it may be concluded that:

- The partial evaluation approach implemented in this work is applied for the evaluation of the objective function in the SBRP model described in the article.
- As a result of the work, the execution time of the SBRP model solution algorithm was reduced with the implementation of the partial evaluation in the objective function in instances with more than two routes as solution.
- The implementation of the partial evaluation does not introduce changes in the results of the problem reached by the algorithm.
- The partial evaluation implementation, for instances with 1 or 2 routes, does not improve the execution time to find the solutions.
- The execution time with the partial evaluation, in instances with more than 2 routes, represents an average of 26.4% of the times consumed by the total evaluation thus reducing the execution time on average by 73.6%.

As recommendations for future work, the following aspects may be considered:

- To implement a more detailed version of the partial evaluation to evaluate only the altered paths in the mutated solutions.
- To implement the partial evaluation also for the evaluation of constraints of the problem.

Declarations

Author contribution statement

Ana Camila Pérez; Eduardo Sánchez-Ansola: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.
Alejandro Rosete: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Omar Rojas; Guillermo Sosa-Gómez: Analyzed and interpreted the data; Wrote the paper.

**Funding statement**

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Data availability statement**

Data included in article/supplementary material/referred in article.

**Declaration of interests statement**

The authors declare no conflict of interest.

**Additional information**

No additional information is available for this paper.

**References**

[1] R.M. Newton, W.H. Thomas, Design of school bus routes by computer, Socio-Econ. Plan. Sci. 3 (1) (1969) 75–85, https://www.sciencedirect.com/science/article/pii/0038012169900512.

[2] W.A. Ellegood, S. Solomon, J. North, J.F. Campbell, School bus routing problem: contemporary trends and research directions, Omega (United Kingdom) 95 (2020) 102056, https://www.sciencedirect.com/science/article/pii/S0305048318305127.

[3] D.M. Miranda, R.S. de Camargo, S.V. Conceição, M.F. Porto, N.T. Nunes, A meta-heuristic for the rural school bus routing problem with bell adjustment, Expert Syst. Appl. 180 (2021) 115086.

[4] A. Bock, E. Grant, J. Könenmann, L. Sanità, The school bus problem on trees, Algorithmica 67 (2013).

[5] A. Rashidi Komijana, P. Ghasemi, K. Khalili-Damghani, F. Hashemi Yazdi, A new school bus routing problem considering gender separation, special students and mix loading: a genetic algorithm approach, J. Optim. Ind. Eng. 14 (2) (2021) 23–39.

[6] E.G. Talbi, Metaheuristics: From Design to Implementation, vol. 74, John Wiley & Sons, 2009.

[7] M. Neroni, Ant colony optimization with warm-up, Algorithms 14 (10) (2021) 205.

[8] J.A. Aledo, J.A. Gámez, A. Rosete, Partial evaluation in rank aggregation problems, Comput. Oper. Res. 78 (2017) 299–304, https://www.sciencedirect.com/science/article/pii/S0305054816302313.

[9] C. Porras, J. Fajardo, A. Rosete, A.D. Masegosa, Partial evaluation and efficient discarding for the maximal covering location problem, IEEE Access 9 (2021) 20542–20556.

[10] A.C. Pérez, E. Sánchez-Ansola, A. Rosete Others, Una solución metaheurística al problema de planificación de rutas de auto-buses escolares con flota homogénea y selección de paradas, Ingenier. 26 (2) (2021).

[11] J. Fajardo, Soft Computing en problemas de optimización dinámicos, Ph.D. thesis, Universidad de Granada, 2015.

[12] D.M. Miranda, R.S. de Camargo, S.V. Conceição, M.F. Porto, N.T. Nunes, A multi-loading school bus routing problem, Expert Syst. Appl. 101 (2018) 228–242, https://www.sciencedirect.com/science/article/pii/S0957417417300927.

[13] Y.E. Hou, L. Dang, W. Dong, Y. Kong, A metaheuristic algorithm for routing school buses with mixed load, IEEE Access 8 (2020) 158293–158305.

[14] P. Schittekat, J. Kinable, K. Sörensen, M. Sevaux, F. Spieksma, J. Springael, A metaheuristic for the school bus routing problem with bus stop selection, Eur. J. Oper. Res. 229 (2) (2013) 518–528.

[15] K.P. Doerner, M. Gendreau, P. Greistorfer, W.J. Gutjahr, R.F. Hartl, M. Reimann, Metaheuristics: Progress in Complex Systems Optimization, vol. 39, Springer Science & Business Media, 2007.

[16] K.H. Rahi, H.K. Singh, T. Ray, Partial evaluation strategies for expensive evolutionary constrained optimization, IEEE Trans. Evol. Comput. (2021).

[17] R. Lewis, K. Smith-Miles, K. Phillips, The school bus routing problem: an analysis and algorithm, in: Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 10765, 2018, pp. 287–298.