ORIGINAL ARTICLE

The rescheduling school bus routing problem under a constraint on the number of students with negative impact

Quy Ta Dinh[a], Thu Huong Dang[b], Minh Hoàng Hà[c], Duc Minh Vu[d], Tien Mai[e]

[a] ORLab, Faculty of Computer Science, Phenikaa University, Hanoi, Vietnam
[b] Department of Management Science, Lancaster University, Lancaster LA1 4YX, UK
[c] CADA, Faculty of Data Science and Artificial Intelligence, College of Technology, National Economics University, Hanoi, Vietnam
[d] SLSCM, Faculty of Data Science and Artificial Intelligence, College of Technology, National Economics University, Hanoi, Vietnam
[e] School of Computing and Information Systems, Singapore Management University, Singapore

ARTICLE HISTORY
Compiled June 24, 2025

ABSTRACT
This study presents a case study of the School Bus Routing Problem (SBRP), focusing on improving the quality of student transportation services. Although numerous studies have addressed the SBRP, this study tackles a new variant driven by a negative impact policy to limit the number of continuing students facing extended travel times. We present this variant and propose a mathematical formulation along with a hybrid metaheuristic approach for it. Furthermore, we conduct comprehensive experiments to evaluate algorithms' performance and various configurations of the negative impact policy on both artificial and real instances. Computational results confirm the effectiveness of our hybrid algorithm.

KEYWORDS
School bus routing problem; negative experience; mixed integer linear programming; ILS; GRASP; hybrid algorithm

1. Introduction

The School Bus Routing Problem (SBRP) involves establishing a reliable, safe, and cost-effective transportation system that transports students to and from schools every day. According to [18], this critical real-world problem impacts millions of families worldwide. Therefore, the SBRP has received ever-increasing attention from academic researchers, industry and the public sector for almost 50 years.

Our case study is motivated by the recent implementation of a new requirement for school bus systems in Vietnamese private schools. In addition to offering exceptional educational and extracurricular experiences, these private schools prioritise high-quality service to maintain their reputation, justify tuition fees, and meet parental expectations. Typically, each of these schools manages its own bus system. Buses follow planned routes

CONTACT Minh Hoàng Hà. Email: hoanghm@neu.edu.vn

with designated stops for picking up students and ensure that students are transported to school.

At the end of each academic semester, the school will create a new bus schedule for transporting students in the next semester. Some students newly register to use the school bus, while others stop using it, resulting in changes to the bus routes compared to the current semester. While adjusting bus schedules to meet operational rules, the school must also address concerns from parents about their children potentially having much longer bus travel times compared to the previous semester. This highlights two key issues. First, if some students now have significantly longer commutes, it could directly affect their physical comfort and emotional well-being. Second, if other students have shorter commutes, it could result in an unfair experience, causing those with longer travel times to feel disadvantaged. To address this, the school has introduced a new requirement to reduce the number of students experiencing these longer bus rides, referred to as "negative impacts".

The objective of this study is to develop minimum-cost routes for a single school bus system that not only satisfy existing constraints but also align with the new requirement. These existing constraints include bus capacity, the maximum travel time (or riding time) of each student on a bus, the maximum route duration, and the maximum number of stops per route.

The SBRP encompasses five distinct sub-problems: bus stop selection, bus route generation, bus route scheduling, school bell time adjustment, and strategic transportation policy [18]. As our case study specifically focuses on generating bus routes, it falls under the category of the Bus Route Generation Sub-problem (BRGP), a specific variant of the Capacitated Vehicle Routing Problem (CVRP).

The contributions of this paper are summarised as follows. Firstly, we introduce a new BRGP variant called the Service-Oriented BRGP (SOBRGP), which distinguishes itself from other BRGPs by placing greater emphasis on service, incorporating multiple measures to prioritise student comfort and safety. This variant also marks a pioneering effort within the literature on the SBGP, using previous semester's solutions to establish constraints for the next semester's problem. Secondly, we devise a mathematical formulation for the SOBRGP and a hybrid metaheuristic to potentially solve large-size instances. The additional constraints, which require the mathematical model to include new variables and constraints tracking the number of students negatively affected, increase the model's size and complexity compared to simpler BRPs. We emphasise that, as we mention later, the newly introduced constraints pose significant challenges for adapting existing heuristic approaches in the CVRP literature. For example, the best metaheuristic algorithms for solving the CVRPs over the past two decades have been based on representing solutions as a giant tour visiting all customers [48, 47, 30]. A splitting procedure, developed using dynamic programming, optimally partitions this giant tour into smaller tours, each corresponding to a vehicle's route. To apply this approach to SOGRGP, an additional dimension must be incorporated into the dynamic programming algorithm of the split procedure to handle constraints related to negative impact. This modification is not trivial and increases the algorithm's complexity. Similarly, the local search phase, a critial component of the state-of-the-art metaheuristics, becomes more computationally intensive due to these new constraints, as verifying each move can no longer be done in $\mathcal{O}(1)$ as in classical CVRPs. Finally, we present extensive computational results on both real and artificial instances. We believe that our artificial instances and solutions can serve as a reference for future research on this variant.

The remainder of the paper is organised as follows. Section 2 contains a literature review. Section 3 describes the SOBRGP, parameters, and notation. Sections 4 and 5 present the Mixed Integer Linear Programming (MILP) formulation and the hybrid metaheuristic, respectively. Section 6 presents the computational results. Finally, Section 7 contains some concluding remarks.

2. Literature review

The literature on the SBRP is vast. While some studies focus on solving a single sub-problem, others simultaneously tackle multiple sub-problems. For a more detailed description and classification of sub-problems and solution approaches, the reader is referred to the surveys [18] and [44]. For the sake of brevity, we will only review papers of direct relevance.

2.1. The bus route generation sub-problem

The BRGP was initially proposed by Newton and Thomas [40], with two constraints: the maximum travel time of students and the bus capacity. A constructive heuristic was developed for the problem using a "route-first, cluster-second" approach [5, 6]. The idea is to construct a single "giant tour" that visits all the bus stops, which is then split into routes that can be travelled by a single bus. While the giant tour is constructed using a heuristic, the splitting is done optimally through a series of shortest-path problems. Hou et al. [29] presented a selection hyper-heuristic for the same problem that uses a reinforcement learning algorithm to select suitable low-level heuristics to iteratively improve the solution. More heuristics were provided in [2, 12, 31, 52] for the same problem but with different objective functions.

To our knowledge, Dulac et al. [16] were the first to consider a variant of the BRGP with maximum route duration constraints, or BRGPD for short. The authors developed several constructive heuristics based on "cluster-first, route-second" [24, 27]. The idea is to group bus stops into clusters and then determine feasible routes within each cluster. Local search operators were then applied to further improve the solutions. This method was later adopted by Chapleau et al. [9] for a different BRGPD variant with additional constraints on the maximum number of stops per route, and by Bowerman et al. [7] for another multi-objective BRGPD variant.

There have been many studies on the BRGP with capacity constraints as the sole constraint. Li and Fu [35] presented a heuristic algorithm based on the Hungarian algorithm and two shortest route algorithms. Corberán et al. [10] proposed constructive methods and further improved obtained solutions by a scatter search algorithm. Later, Pacheco and Martí [43] developed a different constructive heuristics based on [10] and a Tabu Search (TS) with path relinking. Alabas-Uslu [1] developed a self-tuning local search heuristic that outperforms the algorithms in [10] and [43]. More metaheuristics were proposed later, including Ant Colony Optimisation (ACO) [3, 19], genetic algorithms [15], and TS [41, 42].

A few authors have considered hybrid algorithms for BRGP. Eldrandaly and Abdallah [17] proposed a hybrid algorithm combining the ACO metaheuristic with the iterated Lin–Kernighan local improvement heuristic. Euchi and Mraihi [20] developed a hybrid algorithm based on ACO and variable neighborhood local search. Minocha and Tripathi [39] introduced a hybrid algorithm combining the genetic algorithm with local search methods.

2.2. Other relevant SBGPs

Bus Rescheduling Problem (BRP) is a dynamic version of SBGP [34]. It involves adjusting initial bus routes in response to disruption scenarios, such as delays or breakdowns. The goal is to determine a minimum-cost reschedule for the bus fleet while ensuring service coverage for all customers affected by the disruption. To some extent, the BRP is closely linked to the School Bus Operational Rescheduling Graph Problem (SOBRGP), which uses the planned bus schedule (the previous solution of SBGP) as an input and includes servicing extra passengers from disrupted trips.

Most studies on the Bus Rescheduling Problem (BRP) focus on minimising the deviation from the planned schedules or the overall passenger waiting time [13, 33, 53, 54]. In contrast, our new requirement policy only measures the deviation from the previous semester's schedule for students who used the school bus system and continue to use it in the current semester if the deviation exceeds a threshold. (The last semester's travel times for new students enrolled this semester do not exist.) Moreover, literature on the BRP focuses on public bus transportation rather than the school bus system. As a result, the specific constraints related to students' safety and comfort that are crucial to our study have not been addressed in existing BRP literature.

To some extent, the SBGP can be considered a special case of the CVRP, which is one of the most extensively studied variants of the Vehicle Routing Problem. The CVRP involves designing a set of routes for a fleet of vehicles that start from a depot, visit a number of customers, and then return to the depot, subject to the vehicle capacity. The main difference between the SBGP and the CVRP is that, after reaching the school, the buses do not necessarily have to return to their starting point. Consequently, the bus routes are viewed as paths rather than closed tours in the CVRP. Therefore, the SBGP is classified as the Open CVRP variant. For algorithms addressing the Open CVRPs, we refer the reader to the following articles [8, 11, 14, 19, 25, 26, 32, 36, 45, 49, 55].

2.3. Greedy randomised adaptive search procedure and iterated local search

The Greedy Randomised Adaptive Search Procedure (GRASP) is a multi-start algorithm which was first introduced by Feo and Resende [21]. Each iteration consists of a construction phase, where a feasible solution is generated by a randomised greedy algorithm, followed by an improvement phase, where the solution is further improved using local search operators. This process repeats multiple times, with each iteration being independent, until a stopping condition is met. The final solution is the best solution found across all iterations. Despite its simplicity and ease of implementation, GRASP remains an effective metaheuristic that can generate the best-known solutions for numerous combinatorial optimisation problems. For further details, the reader is referred to the surveys [22, 23, 46, 50].

The Iterated Local Search (ILS) was first proposed by Lourenço et al. [37]. It begins by generating an initial solution, which is then improved by local search operators to converge to a local optimum. To escape local optima and explore a wider search space, ILS applies perturbation procedure to the current solution to create new initial solutions for further local searches. This perturbation and local search procedures are repeated iteratively until a stopping condition is met. Similar to GRASP, ILS has been widely applied to various combinatorial optimisation problems. For further information on ILS, the readers are referred to the survey by [38].

3. Problem description and notation

In this section, we provide a detailed description of the SOBRGP. We begin by presenting the problem in Subsection 3.1, followed by an introduction to the parameters and notation in Subsection 3.2.

3.1. Problem narrative

Every day, a fleet of heterogeneous school buses picks up students from specific bus stops and transports them to school. Each bus type is associated with a daily fixed cost, such as the driver's salary and bus rental cost. Buses with larger capacities can transport more students but have higher fixed costs, such as higher wages for more experienced drivers and higher rental prices. The salary of a monitor, who is responsible for supervising students during the ride, is also included in the fixed cost of each bus used.

Each semester, school bus routes need to be redesigned to accommodate newly enrolled students who have recently registered to use the buses, as well as continuing students who used the service in the previous semester and plan to continue using it in the current semester. Whenever possible, continuing students are assigned to the same bus stops they used in the previous semester. However, a few new bus stops may be introduced to accommodate newly enrolled students who are far away from existing bus stops. We note that this paper does not discuss the allocation of students to bus stops or the selection of locations for new bus stops. Instead, it focuses on designing the bus routes for picking up students and dropping them off at school. Therefore, all the following data such as the set of bus stops to be visited, the number of students at each stop, and the travel times among stops and between stops and the school are already known.

Moreover, the travel time from the bus's departure point to the first stop is considered a fixed factor and excluded from the route's duration and cost, as it is typically accounted for separately for administrative convenience. The focus of route design is primarily on the time spent from the first stop onward.

Students are expected to walk from their homes to their assigned bus stops before the bus arrives and then board the bus at these stops to travel to school. Each bus stop must be visited only once by a single bus to ensure that every student is picked up and dropped off at school. Partial pick-ups are not allowed.

Recently, many parents of continuing students have expressed concerns about increased travel times to school for their children compared to last semester. While parents and students are willing to accept the increase that do not exceed γ times the previous duration, such increases in travel time are sometimes unavoidable. In response, the school has introduced a negative impact policy to ensure that the proportion of students experiencing travel time increases exceeding γ times the duration of the previous semester remains below the threshold β. A lower β value indicates that fewer students experience a negative impact, meaning most will have travel times similar to or shorter than their previous duration (not exceeding γ times their past travel time). This leads to greater satisfaction among students and parents but results in higher expenditure for the school. Negotiations between the school and parents determine the value of β to achieve a mutual agreement.

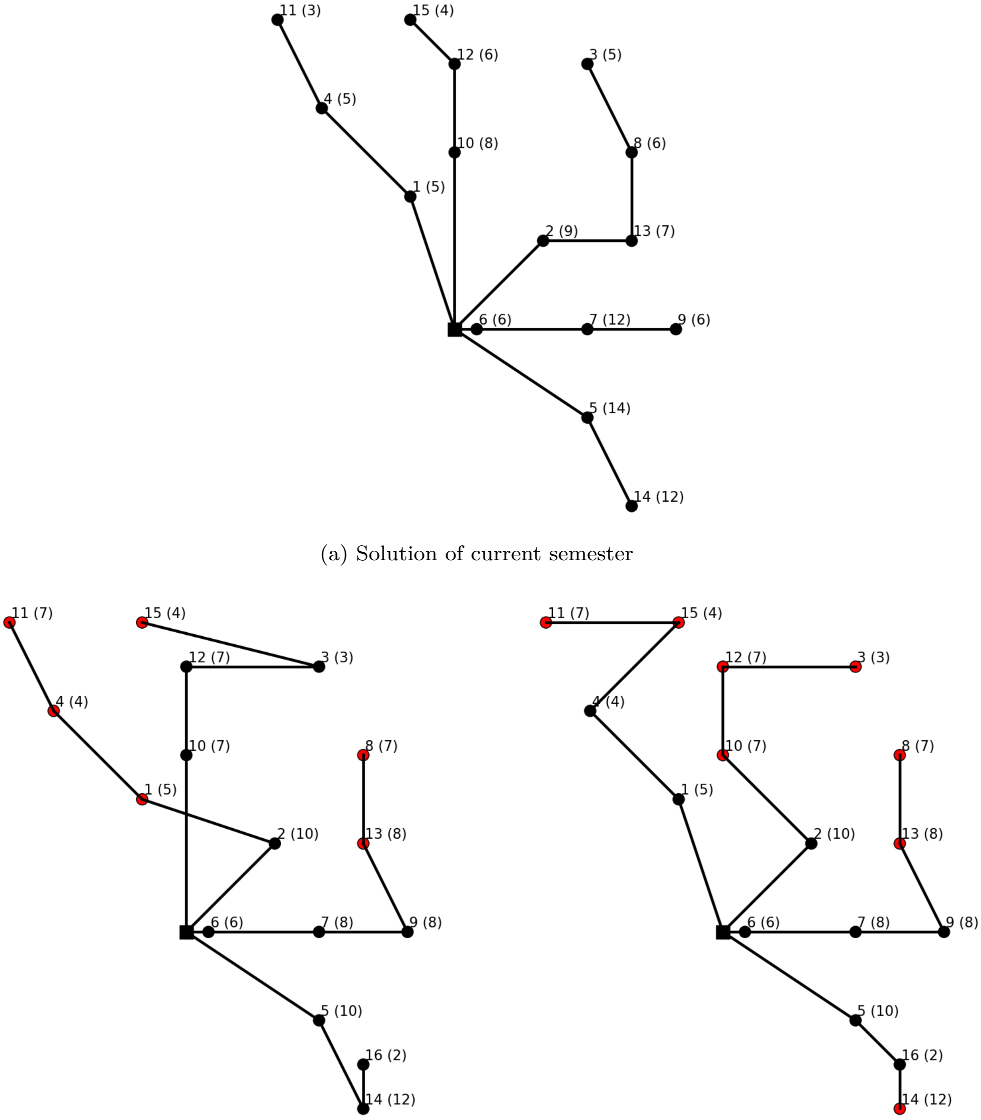
Students facing such travel time increases are considered negatively impacted. We note that newly enrolled students assigned to new stops have their previous semester's riding time set to infinity, indicating that the negative impact policy does not apply to them. However, to address concerns that excessively long riding times might discourage parents from choosing school buses, the constraint that the riding time for each student, including both new and continuing students, must be less than α times the direct travel time from their assigned bus stop to the school is imposed.

Our study places greater emphasis on student services compared to BRGPs discussed in the literature. It not only addresses the negative impact constraint, the maximum riding time for students compared to the direct travel time, but also incorporates additional constraints to enhance students' safety, comfort, and punctuality in arriving at school, as follows: (1) Each bus must spend a sufficient amount of time at each stop to allow students to board and be seated safely before departing; (2) The number of stops that each bus can visit is limited, and (3) each bus has a maximum allowable driving time. We remark that both the second and third rules are necessary. Constraint (2) helps to keep bus routes manageable and prevents excessive stops that could make rides uncomfortable for students. Constraint (3) ensures that students arrive on time and that no student spends more than a limited duration on the bus. These private school policies set this limit at 1 hour. Moreover, the rule simplifies fixed cost calculations since drivers are paid in hourly increments.

The task is to determine the composition of the school bus fleet and the daily routes for each bus, beginning at one bus stop and ending at the school, while satisfying the above-mentioned constraints, with the objective of minimising the total cost. This total cost includes both the daily fixed costs of operating the buses and the travel expenses.

The introduction of negative impact constraints has not been studied in the literature and is worth considering for three reasons. First, significantly longer commutes than in the previous semester may harm students' well-being, affecting their focus and engagement in school. Second, if some students have shorter commutes while others face longer ones, it may create a sense of unfairness. Third, while the proportion of students experiencing negative impacts is capped, the specific students affected are only revealed after a solution is generated. This makes applying existing CVRP heuristics challenging or likely to produce suboptimal solutions.

Figure 1 illustrates the solutions for a SOGRGP instance, where circular nodes represent student pickup points (bus stops), and square nodes represent schools. The numbers outside the parentheses indicate the ID of the pickup points, while the numbers inside the parentheses represent the number of students to be picked up at that location. Node 16 is a newly introduced bus stop in the next period due to newly registered students who will be picked up at this location. The red nodes in the solutions for the next academic period indicate that students picked up at these points will experience negative effects, meaning their travel time on the bus exceeds the permissible threshold compared to the current period. It is evident that the routes in the three scenarios differ significantly. Notably, the optimal solution for the next period in both cases, where $\beta = 30\%$ and $\beta = 100\%$, results in a reduction of one bus compared to the current period's solution. The solution in the case of $\beta = 100\%$ (where the constraint on negative impact is relaxed) still differs fundamentally from the current period's solution due to changes in the number of students at pickup points. As β increases from 30% to 100%, the routes tend to shift towards prioritizing lower operational costs. For instance, when $\beta = 30\%$, bus stop 16 is visited before bus stop 14 to ensure that students at bus stop 14 are not negatively affected. However, when $\beta = 100\%$, the bus follows a shorter route by visiting bus stop 14 before bus stop 16, but as a result, students at bus stop 14 experience negative effects.



(a) Solution of current semester

(b) SOGRGP with $\beta = 30\%$

(c) SOGRGP with $\beta = 100\%$

Figure 1.: Solutions for a SOGRGP instance in three scenarios: current semester's routes (a); Optimal SOGRGP routes for next semester with $\beta = 30\%$ (b) and 100% (c)

3.2. Parameters and notation

We are given a complete directed graph with node set $V = \{0, 1, \ldots, n\}$ and arc set A. Node 0 represents the school. We define V_S as $V \setminus \{0\}$, the set of bus stops. For $v \in V_S$, we exclude $(0, v)$ from A since the bus starts from a bus stop and performs a single route, ending at the school. We assume that the travel costs are asymmetric, with the time of traversing arc (u, v) denoted as τ_{uv}. It is important to note that our approach can be easily adapted in the case of symmetric costs. Let τ^{max} represent the longest traveling time between any two bus stops.

The set $\mathcal{M}$ represents the types of vehicles within a school bus fleet. Each type of bus $m \in \mathcal{M}$ has unlimited availability, a capacity of Q_m, a fixed cost of C^f_m, and a travel cost per distance unit of C^d_m. Let ψ denote the average speed of the bus, irrespective of the type of bus. We define the maximum capacity of a bus in the fleet as $Q^{max} = \max_{m \in \mathcal{M}}\{Q_m\}$. Furthermore, each bus can visit up to L stops, each incurring a loading time w. Let T represent the maximum duration of any route.

The school has a total of N students to be picked up. Each bus stop $v \in V_S$ has a known (positive integer) number of students d_v awaiting boarding, d'_v and t'_v represents the number and the travel time of students assigned to stop v last semester, including loading time at that stop.

If a bus of type m is used, its route is denoted as the sequence $R^m = \{0, v_1, v_2, \ldots, v_l\}$, where the bus departs from stop v_l to pick up d_{v_l} students, and then travels to stop v_{l-1} to pick up $d_{v_{l-1}}$ students and so forth, until it has picked up d_{v_1} students at stop v_1. Finally, it drops off all students at the school at node 0. For $i \in \{1.2, \ldots, l\}$, the travel time and travel distance of students waiting at stop v_i on the route R^m is $t_{v_i} = \tau_{v_i v_{i-1}} + \ldots + \tau_{v_1 0} + w \times i$ and $e_{v_i} = \psi(\tau_{v_i v_{i-1}} + \ldots + \tau_{v_1 0})$. The travel cost of route R^m is computed by multiplying the route length with the travel cost per distance unit of the bus type m (i.e., $C^d_m e_{v_l}$). The total cost of route R^m is computed by summing its travel cost and the fixed cost of the bus type m.

4. The mathematical programming formulation

Our formulation for the SOBRGP uses the following variables:

- For each arc $(u, v) \in A$, the binary variable x_{uv} indicates whether node v is immediately followed by node u on the route.
- For each bus type $m \in \mathcal{M}$ and each node $v \in V_S$, the binary variable y_{mv} indicates whether there is a bus of type m starting from stop v.
- For each node $v \in V_S$, t_v represents the travel time of students waiting at stop v.
- For each node $v \in V_S$, e_v denotes the distance traversed by students at stop v to school.
- For each node $v \in V_S$, the binary variable r_v indicates whether the students at stop v incur a negative impact.
- For each node $v \in V_S$, l_v represents the total number of stops remaining for a bus to pass through on its designated route from stop v to the school, including stop v.
- For each node $v \in V_S$, s_v represents the total number of students remaining for a bus to pick up on its designated route from stop v to the school, including those at stop v.

The formulation is then as follows:

$$\text{Minimise} \sum_{m\in\mathcal{M}} \sum_{v\in V_S} (C_m^d e_v + C_m^f) y_{mv} \tag{1}$$

$$\text{s.t.} \sum_{u\in V_S\setminus\{v\}} x_{uv} + \sum_{m\in\mathcal{M}} y_{mv} = 1 \qquad (v \in V_S) \tag{2}$$

$$\sum_{v\in V\setminus\{u\}} x_{uv} = 1 \qquad (u \in V_S) \tag{3}$$

$$l_v - l_u \leq L(1 - x_{uv}) - 1 \qquad ((u, v) \in A) \tag{4}$$

$$1 \leq l_v \leq L \qquad (v \in V_S) \tag{5}$$

$$s_v - s_u \leq Q^{max}(1 - x_{uv}) - d_u \qquad ((u, v) \in A) \tag{6}$$

$$d_v \leq s_v \leq Q^{max} \qquad (v \in V_S) \tag{7}$$

$$s_v \leq Q^{max} - y_{mv}(Q^{max} - Q_m) \qquad (v \in V_S, m \in \mathcal{M}) \tag{8}$$

$$t_v - t_u + \tau_{uv} + w \leq (T + \tau^{max})(1 - x_{uv}) \qquad ((u, v) \in A) \tag{9}$$

$$w + \tau_{v0} \leq t_v \leq \max\{T, \alpha\tau_{v0}\} \qquad (v \in V_S) \tag{10}$$

$$e_v - e_u + \psi\tau_{uv} \leq \psi(T + \tau^{max})(1 - x_{uv}) \qquad ((u, v) \in A) \tag{11}$$

$$t_v - t'_v\gamma \leq Tr_v \qquad (v \in V_S) \tag{12}$$

$$\sum_{v\in V_S} d'_v r_v \leq N\beta \tag{13}$$

$$t_0 = l_0 = s_0 = e_0 = 0 \tag{14}$$

$$x_{uv} \in \{0, 1\} \qquad ((u, v) \in A) \tag{15}$$

$$y_{mv} \in \{0, 1\} \qquad (v \in V,\ m \in \mathcal{M}) \tag{16}$$

$$r_v \in \{0, 1\} \qquad (v \in V_S) \tag{17}$$

$$e_v \in \mathbb{R}^+ \qquad (v \in V_S) \tag{18}$$

The objective function (1) minimises the total cost, which includes the fixed cost for all buses in use and the travel cost. Constraints (2) and (3) force each bus stop to be visited exactly once. Constraints (4), (6), (9) and (11) are known as Miller-Tucker-Zemlin constraints and ensure that the routes are connected. Constraints (4-5) impose the maximum number of stops per route. Constraints (6-7) impose the vehicle capacity restrictions and constraints (8) ensure that each route is assigned the correct type of bus based on its physical capacity. Constraints (9) and (10) impose the maximum riding time for each student and, at the same time, the maximum travelling time for each route. Constraints (11) define travel distance of students waiting at stop v.

Constraints (12) imply that if the riding time of students waiting at stop v is more than γ times longer than before, these students will experience a negative impact. Constraints (13) limit the total number of students who experience a negative impact. Finally, constraints (14) to (18) simply describe the range of variables.

The objective function is non-linear because of the terms $(C_m^d e_v + C_m^f)y_{mv}$, which are the product of a continuous and a binary variables. By defining f_{mv} as the total cost incurred by a bus of type m starting from stop v, for each type $m \in \mathcal{M}$ and each node $v \in V_S$ (i.e., $f_{mv} = (C_m^d e_v + C_m^f)y_{mv}$), and by adding the following constraints:

$$f_{mv} \geq C_m^d e_v + C_m^f y_{mv} + \psi T C_m^d (y_{mv} - 1) \qquad (m \in \mathcal{M}, v \in V_S)$$

$$f_{mv} \geq 0 \qquad (m \in \mathcal{M}, v \in V_S)$$

to the previous programming formulation, the objective function can be linearised as

$$\text{Minimise} \sum_{m\in\mathcal{M}} \sum_{v\in V_S} f_{mv}$$

The resulting formulation is called the MILP. We will conduct computational experiments with this formulation, and the results are shown in Section 6.

5. Metaheuristic

Given that the instances which can be optimally solved using the previous MILP are very small-scale, we propose a hybrid metaheuristic in this section capable of producing good solutions for larger-scale instances. In particular, we have designed a Multi-Start Iterated Local Search (MS-ILS) that combines two different metaheuristics: the Greedy Randomised Adaptive Search Procedure (GRASP) and Iterated Local Search (ILS). The MS-ILS is outlined in Subsection 5.1, and its procedures are described in the following subsections.

5.1. Multi-start iterated local search

Algorithm 1 describes our hybrid algorithm for the SOBRGP. S^* is the best solution found over ϕ iterations. Each of these iterations provides the inner loop (Lines 3-16) with a different initial solution using the Construction procedure (Line 3), which is then improved by Local Search operators (Line 4). Within the inner loop, each initial solution (or current solution) S is transformed into a giant tour S_G by concatenating all routes and eliminating the copies of node 0 (Line 8). The obtained giant tour then undergoes the Perturbation procedure before being transformed into a new feasible solution S' for the SOBRGP via the Split procedure (Lines 9-10). The solution is further improved using Local Search operators (Line 11). The inner loop terminates after φ consecutive iterations without improvement. In both the outer and inner loops, the new solution is accepted if an improvement is found. Note that we define $f(S)$ as the total cost of solution S.

It can be seen that when φ is equal to 1, our metaheuristic becomes GRASP, while when ϕ is equal to 1, our metaheuristic becomes ILS. Therefore, based on studies [48] and [28], our hybrid algorithm is also called GRASP $\times$ ILS.

5.2. Construction procedure

To kick off the inner loop (or the ILS phase), we compute initial solutions using the well-known *Cheapest Insertion Heuristic* (CIH). This heuristic was initially developed for the Traveling Salesman Problem [51], but we have specially tailored it for our particular problem. Additionally, to generate different initial solutions, we integrate randomisation into the node selection process.

The CIH is outlined in Algorithm 2. First, it initialises a set of unvisited bus stops $V^{\text{unvisited}}$ containing all bus stops, and a solution S consisting of a route containing only node 0 (Line 1-2). Such a route is called “empty” as it contains no bus stops and the node 0 is positioned first. The algorithm then computes insertion costs for each unvisited stop at all feasible positions and stores them in the *insertPattern* list (Line 4-10). The

Algorithm 1: The framework of the MS-ILS

Set $f(S^*)$ to ∞;
for $i = 1$ **to** ϕ **do**
    Let S be the solution obtained by the *Construction* procedure;
    Execute the *LocalSearch* procedure on S;
    Set j to 0 ;
    while $j < \varphi$ **do**
        Set j to $j + 1$;
        Merge all routes in S, removing duplicates of node 0, to form S_G;
        Execute the *Perturbation* procedure on S_G;
        Let S' be the solution obtained by executing the *Split* procedure on S_G;
        Execute the *LocalSearch* procedure on S';
        if $f(S') < f(S)$ **then**
            Set S to S';
            Set j to 0;
    if $f(S') < f(S^*)$ **then**
        Set S^* to S';
return S^*

insertion position is considered feasible if it satisfies all constraints of the problem, with the vehicle capacity constraint replaced by ensuring that the total demand does not exceed Q^{max}. When calculating the insertion cost, the bus type chosen is the smallest one that can still meet the total demand of the route. Subsequently, the *insertPattern* list is sorted in increasing order of insertion cost (Line 11). The algorithm selects a tuple randomly from the sorted list based on a parameter λ (Line 12-14), inserts the chosen unvisited node into the selected position (Line 15) and updates the set $V^{\text{unvisited}}$ accordingly (Line 16). If S does not include an empty route, one is created (Lines 17-18). This ensures there's always at least one route into which unvisited nodes can be inserted in the subsequent step. This process continues until the set $V^{\text{unvisited}}$ is empty.

5.3. Split procedure

The aim of the split procedure is to divide the giant tour into segments that are small enough to be traversed by a single vehicle. This procedure is originally proposed by Beasley [5] as part of the "route-first cluster-second" heuristic. Beasley showed that the splitting is done optimally by solving a series of shortest-path problems in an auxiliary directed acyclic graph. Prins [47] later presented a dynamic programming approach for this splitting version without explicitly generating the auxiliary graph. We adapt Prins's version for our Split procedure.

The giant tour can be defined as a sequence of bus stops $(v_1, v_2, \cdots, v_n)$ travelled in the order. Each subsequence $R_{ij} = (v_i, \cdots, v_j)$, where $i \leq j$, corresponds to a route of a single bus $R_{ij}^m = \{0, v_i, \cdots, v_j\}$, in which the bus departs from stop v_j, then go to stop v_{j-1}, and so on, until it reaches stop v_i, and finally visits the school at node 0. The smallest bus m that fits the total number of students on the route is chosen. The set of subsequences corresponding to feasible routes is denoted by $\mathcal{R}$.

For each subsequence R_{ij}, we define a set of associated metrics for the corresponding route:

- c_{ij}: the total cost.
- t_{ij}: the riding time of the students waiting at bus stop v_i.

Algorithm 2: Construction procedure

```
Set V^unvisited to V_S;
Let S include a empty route;
while V^unvisited is not empty do
    Create an empty list insertPatterns;
    for v ∈ V^unvisited do
        for R ∈ S do
            for p = 2, ..., len(R) + 1 do
                if insertion of v at position p in route R is feasible then
                    Let c be the increase in cost by inserting v at position p in R;
                    Add (c, v, p, R) to insertPatterns;
    Sort insertPatterns in increasing order of cost;
    for (c, v, p, R) ∈ insertPatterns do
        Let r be a random number in interval [0, 1];
        if r < λ then
            Insert node v at position p in route R ;
            Remove v from V^unvisited ;
            if S does not include an empty route then
                Create an empty route and add it to S;
            break;
Remove empty routes from S;
return S
```

- l_{ij}: the number of bus stops.
- s_{ij}: the number of students picked up.
- $\hat{s}_{ij}$: the number of students having the negative impacts.

Algorithm 3 shows how to efficiently generate the set $\mathcal{R}$ and evaluate the metrics for each subsequence in $\mathcal{R}$. One can verify that the computation of metrics for subsequence R_{ij} is derived in $O(1)$ time from the metrics of $R_{i+1,j}$, rather than scanning the entire subsequence.

We remark that we only consider subsequences corresponding to feasible routes. As a result, if the subsequence R_{ij} leads to an infeasible route, we no longer consider the subsequences $R_{i',j}$ where $i' < i$. A route is considered feasible if it satisfies the following constraints: (1) the maximum number of stops per route ($l_{ij} \leq L$); (2) the maximum duration of each route ($t_{ij} \leq T$); (3) the maximum riding time of students at stop i ($t_{ij} \leq \alpha\tau_{i0}$), and (4) the capacity constraint ($s_{ij} \leq Q^{max}$). Route feasibility can be checked in constant time ($O(1)$). Consequently, each subsequence R_{ij} is processed in $O(|\mathcal{M}|)$, resulting in an overall worst-case complexity of $O(nL|\mathcal{M}|)$ for Algorithm 3.

Algorithm 3: Computing associated metrics for all subsequences

for $j = 1, \ldots, n$ **do**
    Set i to j;
    Set l_{ij} to 1;
    Set t_{ij} to $\tau_{v_j 0} + w$;
    Set s_{ij} to d_{v_j};
    Set $\hat{s}_{ij}$ to 0;
    while *R_{ij} corresponds to an feasible route* **do**
        Let m be a bus such that $Q_m = \min\{C^f_m : Q_{m'} \geq s_{ij}, m' \in \mathcal{M}\}$;
        Set c_{ij} to $C^f_m + C^d_m(t_{ij} - l_{ij} * w)\psi$;
        Add a subsequence R_{ij} to $\mathcal{R}$;
        if $i > 1$ **then**
            Set i to $i - 1$;
            Set l_{ij} to $l_{i+1,j} + 1$;
            Set t_{ij} to $t_{i+1,j} + \tau_{v_{i+1}v_i} + w - \tau_{v_{i+1}0} + \tau_{v_i 0}$;
            Set s_{ij} to $s_{i+1,j} + d_{v_i}$;
            if $t_{ij} > \gamma t'_i$ **then**
                Set $\hat{s}_{ij}$ to $\hat{s}_{i+1,j} + d_{v_i}$;
            else
                Set $\hat{s}_{ij}$ to $\hat{s}_{i+1,j}$;

return $\mathcal{R}$

Algorithm 4 demonstrates how to compute solutions for the SOBRGP. We define $f(j, \hat{s})$ as the minimum cost when partitioning the first j bus stops into feasible routes with exactly $\hat{s}$ negatively impacted students, where $0 \leq \hat{s} \leq N\beta$ and $0 \leq j \leq n$. The $f(j, \hat{s})$ values are computed recursively (Line 1-13). The cost of the optimal splitting is the minimum value of $f(n, \hat{s})$, where $0 \leq \hat{s} \leq N\beta$ (Line 14-18). The predecessors $P_{j\hat{s}}$ and $Q_{j\hat{s}}$ enable us to reconstruct the set of feasible routes back to bus stop v_1 (Line 19-23). One can check that the algorithm runs in $O(nLN\beta)$ time.

Algorithm 4: Split procedure

```
for j = 0 to n do
    for ŝ = 0 to Nβ do
        Set f(j, ŝ) to +∞;
Set f(0.0) to 0;
for j = 1 to n do
    for ŝ = 0 to Nβ do
        Set i to j;
        while (R_{ij} ∈ \mathcal{R}) and (ŝ_{ij} ≤ ŝ) and (i ≥ 1) do
            if f(j, ŝ) > f(i − 1, ŝ − ŝ_{ij}) + c_{ij} then
                Set P_{jŝ} to i − 1;
                Set Q_{jŝ} to ŝ − ŝ_{ij};
                f(j, ŝ) = f(i − 1, ŝ − ŝ_{ij}) + c_{ij};
            Set i to i − 1;
Set f* to +∞ and ŝ* to 0;
for ŝ = 0 to Nβ do
    if f(n, ŝ) < f* then
        Set f* to f(n, ŝ);
        Set ŝ* to ŝ;
Set j to n, ŝ to ŝ*, and S to ∅;
while j > 0 do
    Set i to P_{jŝ} + 1;
    Add the corresponding route of the subsequence R_{ij} to S;
    Set j to i − 1 and ŝ to Q_{jŝ};
return S and its cost f*.
```

5.4. Local search

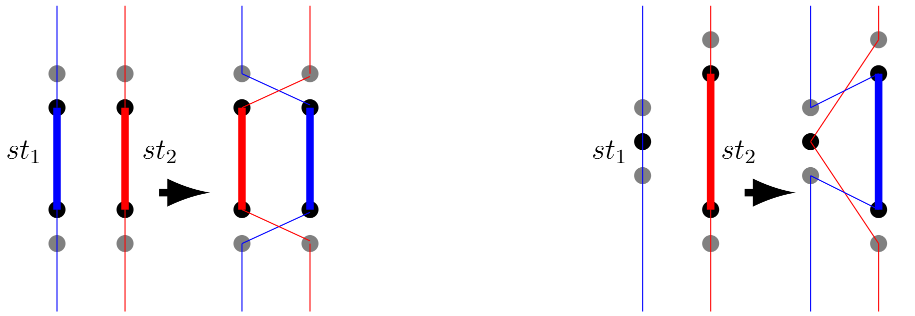
Our local search uses classical neighborhood moves known as "Cross String", which involves swapping two strings between the same or different routes, as shown in Figure 2. (In the online version, the blue lines show the string segments chosen from one route for swapping. The red lines show the other segments involved in the swap, which may come from the same or a different route. The black dots mark the end nodes of each string.) The length of a string is limited to $\lfloor \frac{L}{2} \rfloor$, although the swapped strings may have different lengths. Note that the Cross String neighborhood includes the "Relocation" move as a special case, illustrated in Figure 2b, where the relocation of one string is caused by another string having a length of 0. (A string of zero length indicates that its endpoints coincide.) We do not consider scenarios where both strings have lengths of 0, as the routes remain unchanged. We use the first improvement strategy, where upon finding an improved solution, it replaces the current solution. The pseudocode is described in Algorithm 5.

After creating a new solution, we need to check its feasibility and determine whether there is a reduction in total cost. While the latter is straightforward, the former requires a more efficient approach due to the numerous constraints involved. To manage this, we categorise these constraints into groups that can be evaluated simultaneously and examine them sequentially based on their computational complexity. If one constraint is found to be violated, the checking process is terminated. The first group contains a single constraint on the maximum number of stops per route, which can be verified in $\mathcal{O}(1)$ time. The second group includes constraints on the maximum route duration and vehicle capacity. This evaluation is conducted alongside the checking of cost savings. The third group consists of constraints on the maximum riding time of students and the maximum number of negatively impacted students. The checks for the second and third groups can be completed in $\mathcal{O}(L)$ time.

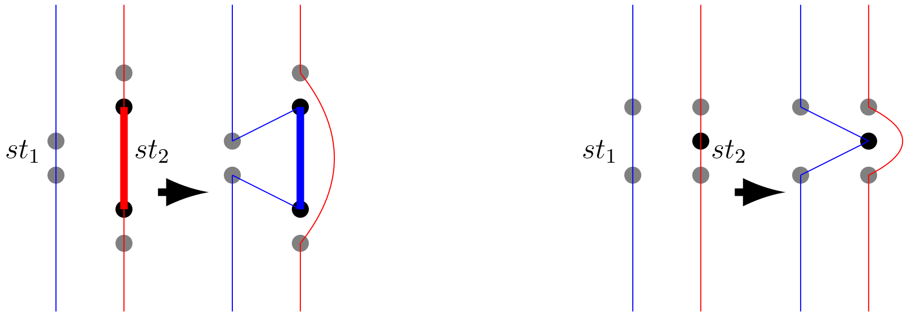
Algorithm 5: Local search

Input : Solution S

Set L_{ls} to $\lfloor \frac{L}{2} \rfloor$;
Set *stop* to $False$;
while *stop* $== False$ **do**
  Set *stop* to $True$;
  for *route* R^{m_1} *in* S **do**
    for *route* R^{m_2} *in* S **do**
      for $i_1 = 1$ *to* $len(R^{m_1})$ **do**
        for $len_1 = 0$ *to* $\min\{L_{ls}, len(R^{m_1}) - i_1\}$ **do**
          for $i_2 = 1$ *to* $len(R^{m_2})$ **do**
            for $len_2 = 0$ *to* $\min\{L_{ls}, len(R^{m_1}) - i_2\}$ **do**
              Let st_1 be a string of length len_1 on R^{m_1};
              Let st_2 be a string of length len_2 on R^{m_1};
              if *swapping* st_1 *and* st_2 *is feasible and cost-saving is positive* **then**
                Swap st_1 and st_2;
                Set *stop* to $False$;
return Solution S



a, General cases



b, Special cases

Figure 2.: Cross string moves

5.5. Perturbation procedure

The appropriate level of perturbation is critical for the performance of ILS phase [48]. Insufficient perturbation may not effectively escape local optima, while excessive perturbation can cause major changes in the solution, potentially slowing down the convergence to the global optimum. Moreover, to explore different regions in the solution space, perturbation moves should not be easily achieved by local search moves. After conducting preliminary experiments, we opted to randomly swap between 1 to 3 pairs of stops in the giant tour for perturbation procedure. Note that no constraints require checking in this procedure.

6. Experimental results

In this section, we describe our experiments and present the findings. The real instances and their parameters are introduced in Section 6.1. For comprehensive testing of the algorithms' performance, we generated artificial instances and discussed parameter settings in subsection 6.2. Extensive computational results on a set of generated instances and real instances are presented in Subsection 6.3 and Subsection 6.4, respectively, along with a sensitivity analysis on some parameters from Subsections 6.5-6.7.

All experiments are coded in C++ and tested on a single thread of an AMD Ryzen Threadripper PRO 5975WX Processor at 3.60 GHz with 32GB of RAM. To solve the MILP, we use ILOG CPLEX 22.1.1 Solver with its default setting. A time limit of one hour per instance was imposed. Based on our preliminary experiments, we set ϕ to 5 and φ to 12000 for the MS-ILS framework, and λ to 0.9 for the construction procedure.

For each instance, we run the MS-ILS algorithm 10 times with difference random seeds and record the best and average solutions.

6.1. Real instances

We are given two real instances. The first instance, **V41-N185-real**, involves 40 stops and 185 students, while the second instance, **V69-N324-real**, involves 68 stops and 324 students. The number of students at each stop and the travel time of continuing students from current semester are provided and used to generate the schedule for the next semester. However, due to confidentiality reasons, we cannot disclose the specifics of this information. Other parameters are as follows:

- The maximum number of stops per route L is 7.
- The maximum route duration T is fixed at 60 minutes.
- The loading time at each stop w is (estimated to be) 2 minutes.
- The value of α is 2.0, meaning a student's riding time is capped at twice the direct travel time from their bus stop to school. The value of γ is 1.05, meaning students are considered to experience a negative impact if their riding time increases by more than 1.05 times their travel duration from the previous semester.
- Three types of buses are available, with capacities of $16, 29$, and 45 seats. They can accommodate up to $15, 28$, and 44 students, respectively.
- The distance between two stops is calculated using the OpenStreetMap API, and the average bus speed is set to 20 km/h to account for urban settings.

According the operations manager of the school, the current schedule was manually made. And due to human computational limitations, they could not ensure 100% compliance with all constraints, including those related to negative impacts. Recently, numerous complaints from parents have compelled them to seek a more scientific approach to solving the problem. We were also informed that, in the next semester, approximately 20% of students would discontinue using the bus stop system and be replaced by roughly the same number of newly enrolled students. There is also a possibility that up to 5% of new bus stops could be introduced.

The fixed cost per day C_m^f and travel cost per kilometer C_m^d of a bus type are provided in Table 1, with the costs in thousands of Vietnamese dong.

Bus capacity (Q_m)	Fixed cost per day (C_m^f)	Travel cost per km (C_m^d)
15	950	8
28	1200	10
44	1700	15

Table 1: Fixed and travel costs for each of bus type

6.2. Artificially generated instances

Since the SOBRGP has not been addressed in the literature, we generated test instances to evaluate the performance of our algorithms. The main idea is to generate instances that imitate real scenarios. In particular, we used benchmark instances for the CVRP created by Augerat [4], with the number of nodes varying between 16 and 101. Nodes represent the bus stops, the depot represents the school, and the demand at each node represents the number of students waiting at the corresponding bus stop. The coordinates of nodes are provided, enabling us to calculate the travel time between two stops by dividing the Euclidean distance by the average bus speed. We use minutes as the unit for travel time. It is important to mention that we adjust the coordinates of the graph nodes to fit within a 20x20 km grid, which represents the operational area of a bus system in an actual urban environment.

In previous semesters, the school manually generated bus routes based on the scheduler's experience, using a process similar to a greedy heuristic with nearest insertion to create the routes. We also adapted a heuristic approach to generate the last semester's schedule for continuing students in our test instances. In particular, we used our construction procedure without the randomised factor to obtain an initial solution with $t'_v = +\infty \; \forall v \in V$ (i.e., no student experiences negative impact), and then applied our proposed local search procedure to improve it.

To generate data for newly enrolled students, we began by randomly removing 20% of students from their assigned bus stops, rounding down to the nearest integer number. Then, we randomly introduced up to 5% new bus stops within the existing graph. The same number of students removed were then re-added as newly enrolled students, with their assignments to bus stops made randomly. All removal and insertion operations were applied uniformly across all stops. Table 2 shows the number of bus stops (V_S) and the average number of students ($\overline{d}$) waiting at bus stops both before and after the generation of newly enrolled student data for each instance. Our instances are labeled Va-Nb, where a represents the number of bus stops and b represents the number of students.

We assume the same three types of buses for all generated instances are used as in the real instances. Other parameters T, L, w, γ, and α are also unchanged. For each instance, we test different values of $\beta = \{0\%, 10\%, 30\%, 50\%, 100\%\}$ to assess the impact of the negative impact threshold on our algorithms. Note that a β value of 100% indicates no restrictions on the number of students experiencing a negative impact, meaning the negative impact policy is not implemented. We also remark that, due to instance creation, setting $\beta = 100\%$ effectively means β is around 80%, as roughly 20% of the students are classified as new. These students are not subject to this constraint, and their proportion varies slightly across instances due to rounding. Full details of all instances are made available at ORLab Data Repository[1].

6.3. Computational results for artificial instances

Table 3 presents the results obtained with the MILP formulation. The first column shows the instance names, while the second presents the value of β, the maximum proportion of students experiencing negative impacts (i.e., those who travel time increases exceed γ times their travel duration from the previous semester). The next three columns display the total cost (Column "Obj"), the fixed cost (Column "C^f"), and the travel cost (Column "C^d") of the best found solution, respectively. We note that the total cost in this study refers to the operating cost, which includes daily fixed costs such as driver wages, bus rentals, and other expenses, in addition to the travel cost. The sixth column reports the percentage gap between the upper and lower bounds obtained with the default branch-and-cut algorithm of CPLEX. The seventh column provides the running time of the algorithm in seconds. The last column shows the proportion of negatively impacted students in the current best solution.

[1] http://orlab.com.vn/home/download

Instance	Current semester		Next semester	
	V_S	$\overline{d}$	V_S	$\overline{d}$
V17-N108	15	7.20	16	6.75
V19-N114	19	6.00	18	6.33
V21-N108	21	5.14	20	5.40
V22-N139	20	6.95	21	6.62
V23-N126	20	6.30	22	5.73
V24-N154	22	7.00	23	6.70
V24-N96	23	4.17	23	4.17
V40-N269	40	6.73	39	6.90
V46-N192	46	4.17	45	4.27
V50-N270	50	5.40	49	5.51
V51-N234	49	4.78	50	4.68
V51-N283	50	5.66	50	5.66
V54-N321	52	6.17	53	6.06
V55-N318	57	5.58	54	5.89
V57-N336	55	6.11	56	6.00
V57-N341	55	6.20	56	6.09
V57-N347	56	6.20	56	6.20
V60-N281	63	4.46	59	4.76
V61-N337	60	5.62	60	5.62
V66-N321	65	4.94	65	4.94
V72-N381	72	5.29	71	5.37
V77-N409	76	5.38	76	5.38
V78-N391	74	5.28	77	5.08
V106-N495	101	4.90	105	4.71

Table 2: Summary of test instances

Data	$\beta(\%)$	Obj	C^f	C^d	Gap	T(s)	$\hat{s}$
V17-N108	0	6410	6000	410	0.0	0	0.0
	10	6146	5750	396	0.0	2	3.4
	30	5800	5300	500	0.0	72	29.9
	50	5760	5300	460	0.0	94	44.8
	100	5760	5300	460	0.0	62	48.3
V19-N114	0	6266	5750	516	0.0	4	0.0
	10	5850	5300	550	0.0	77	7.6
	30	5840	5300	540	0.0	437	19.6
	50	5840	5300	540	0.0	306	19.6
	100	5600	5050	550	0.0	253	51.1
V21-N108	0	6290	5750	540	0.0	6	0.0
	10	6020	5500	520	0.0	90	8.0
	30	5845	5300	545	0.0	1338	23.0
	50	5845	5300	545	0.0	652	17.2
	100	5625	5050	575	0.0	595	54.0
V22-N139	0	7782	7200	582	0.0	1	0.0
	10	7472	6950	522	0.0	436	8.9
	30	7080	6500	580	21.2	3600	29.5
	50	7080	6500	580	24.2	3600	30.4
	100	6858	6250	608	16.9	3600	68.8
V23-N126	0	7349	6700	649	0.0	200	0.0
	10	6994	6450	544	16.6	3600	9.9
	30	6574	6000	574	27.6	3600	21.8
	50	6820	6250	570	27.1	3600	29.7
	100	6574	6000	574	25.8	3600	19.8
V24-N154	0	8484	7900	584	0.0	6	0.0
	10	7800	7200	600	0.0	3195	8.9
	30	7610	7000	610	24.8	3600	29.8
	50	7411	6750	661	24.7	3600	50.0
	100	7190	6500	690	22.4	3600	54.8
V24-N96	0	7436	6900	536	0.0	1	0.0
	10	6756	6200	556	0.0	156	5.2
	30	6472	5950	522	0.0	1328	29.9
	50	6464	5950	514	0.0	2155	40.3
	100	6464	5950	514	0.0	1460	36.4
V40-N269	0	14184	12950	1234	0.0	980	0.0
	10	13960	12700	1260	33.8	3600	7.9
	30	13454	12050	1404	52.0	3600	22.2
	50	13886	12700	1186	51.7	3600	25.0
	100	13378	12050	1328	52.4	3600	19.4
V46-N192	0	12702	11450	1252	0.0	692	0.0
	10	12656	11450	1206	28.9	3600	6.5
	30	13122	11900	1222	59.2	3600	22.7
	50	12394	11200	1194	55.7	3600	29.2
	100	12388	11200	1188	55.7	3600	28.6
V50-N270	0	14252	12950	1302	19.9	3600	0.0
	10	14873	13450	1423	66.2	3600	5.6
	30	14680	13400	1280	72.5	3600	13.0
	50	14291	12950	1341	79.0	3600	35.6
	100	14916	13450	1466	79.4	3600	26.4
V51-N234	0	15334	14050	1284	15.1	3600	0.0
	10	15648	14300	1348	55.6	3600	8.0
	30	15155	13850	1305	80.1	3600	26.6
	50	14242	12900	1342	72.0	3600	36.2
	100	13650	12250	1400	78.0	3600	37.2
V51-N283	0	16288	14850	1438	19.2	3600	0.0
	10	15613	14150	1463	61.9	3600	9.7
	30	15852	14400	1452	80.7	3600	28.6
	50	15376	13900	1476	77.8	3600	36.1
	100	16113	14600	1513	80.8	3600	45.8

Data	$\beta(\%)$	Obj	C^f	C^d	Gap	T(s)	$\hat{s}$
V54-N321	0	17350	15800	1550	17.5	3600	0.0
	10	17334	15800	1534	43.5	3600	8.9
	30	16986	15350	1636	63.7	3600	28.4
	50	16239	14650	1589	71.4	3600	41.6
	100	16907	15150	1757	70.4	3600	40.9
V55-N318	0	17823	16300	1523	15.0	3600	0.0
	10	17060	15600	1460	50.2	3600	9.8
	30	17185	15600	1585	82.3	3600	24.7
	50	17351	15800	1551	82.6	3600	28.6
	100	16385	14900	1485	76.9	3600	27.8
V57-N336	0	17835	16300	1535	14.7	3600	0.0
	10	18212	16750	1462	62.3	3600	7.1
	30	17814	16300	1514	93.0	3600	26.8
	50	17851	16300	1551	80.9	3600	29.7
	100	18322	16600	1722	85.4	3600	36.8
V57-N341	0	18524	17000	1524	22.9	3600	0.0
	10	18476	16800	1676	64.4	3600	8.8
	30	18656	17000	1656	77.9	3600	17.6
	50	18535	16800	1735	78.8	3600	30.4
	100	18149	16350	1799	78.2	3600	41.8
V57-N347	0	18371	16800	1571	21.7	3600	0.0
	10	18611	16850	1761	73.9	3600	9.7
	30	18310	16600	1710	89.3	3600	23.4
	50	18796	17050	1746	84.4	3600	36.0
	100	17872	16100	1772	89.0	3600	37.4
V60-N281	0	16826	15300	1526	24.0	3600	0.0
	10	16687	15100	1587	63.1	3600	9.3
	30	15405	13900	1505	74.5	3600	16.9
	50	15711	14150	1561	74.9	3600	27.1
	100	16492	15050	1442	76.0	3600	20.4
V61-N337	0	19035	17250	1785	21.1	3600	0.0
	10	19382	17700	1682	56.5	3600	10.0
	30	18686	17050	1636	76.9	3600	19.3
	50	19110	17250	1860	79.5	3600	25.2
	100	18211	16350	1861	94.7	3600	44.4
V66-N321	0	19246	17450	1796	33.7	3600	0.0
	10	20215	18400	1815	74.1	3600	7.8
	30	19592	17700	1892	80.3	3600	24.1
	50	20028	18150	1878	82.6	3600	29.2
	100	19359	17450	1909	84.7	3600	30.0
V72-N381	0	20764	18700	2064	25.7	3600	0.0
	10	21915	19850	2065	69.8	3600	8.9
	30	22119	19900	2219	82.4	3600	23.9
	50	21849	19650	2199	95.7	3600	37.0
	100	21850	19450	2400	82.2	3600	53.8
V77-N409	0	23092	21050	2042	37.0	3600	0.0
	10	24148	21800	2348	74.7	3600	8.2
	30	24672	22250	2422	84.4	3600	25.0
	50	24157	21800	2357	86.0	3600	26.2
	100	23779	21550	2229	85.7	3600	31.4
V78-N391	0	22232	20150	2082	38.2	3600	0.0
	10	23917	21750	2167	71.6	3600	8.9
	30	23923	21500	2423	83.7	3600	23.6
	50	23581	21100	2481	83.6	3600	38.0
	100	23065	20800	2265	83.1	3600	31.3
V106-N495	0	31826	28900	2926	97.0	3600	0.0
	10	31982	28950	3032	98.6	3600	6.3
	30	31593	28700	2893	100.0	3600	27.0
	50	31475	28300	3175	100.0	3600	30.6
	100	31142	28000	3142	100.0	3600	37.1

Table 3: Results of MILP model for artificially generated instances

Data	β(%)	Obj	C^f	C^d	Gap'	Gap	T(s)	$\hat{s}$
V17-N108	0	6410	6000	410	0.0	0.0	20	0.0
	10	6146	5750	396	0.0	0.0	17	3.4
	30	5800	5300	500	0.0	0.0	17	29.9
	50	5760	5300	460	0.0	0.0	15	48.3
	100	5760	5300	460	0.0	0.0	15	48.3
V19-N114	0	6266	5750	516	0.0	0.0	21	0.0
	10	5850	5300	550	0.0	0.0	18	7.6
	30	5840	5300	540	0.0	0.0	30	19.6
	50	5840	5300	540	0.0	0.0	17	19.6
	100	5600	5050	550	0.0	0.0	22	51.1
V21-N108	0	6290	5750	540	0.0	0.0	25	0.0
	10	6020	5500	520	0.0	0.0	25	8.0
	30	5845	5300	545	0.0	0.0	21	17.2
	50	5845	5300	545	0.0	0.0	21	17.2
	100	5625	5050	575	0.0	0.0	22	54.0
V22-N139	0	7782	7200	582	0.0	0.0	36	0.0
	10	7472	6950	522	0.0	0.0	39	8.9
	30	7080	6500	580	0.0	0.0	28	27.7
	50	6895	6250	645	0.0	-2.6	30	48.2
	100	6858	6250	608	0.0	0.0	27	68.8
V23-N126	0	7349	6700	649	0.0	0.0	37	0.0
	10	6850	6250	600	0.0	-2.1	30	9.9
	30	6574	6000	574	0.0	0.0	28	21.8
	50	6574	6000	574	0.0	-3.6	38	19.8
	100	6570	6000	570	0.0	-0.1	31	55.4
V24-N154	0	8484	7900	584	0.0	0.0	53	0.0
	10	7800	7200	600	0.0	0.0	39	9.7
	30	7610	7000	610	0.0	0.0	34	29.8
	50	7411	6750	661	0.0	0.0	34	48.4
	100	7190	6500	690	1.8	0.0	31	54.8
V24-N96	0	7436	6900	536	0.0	0.0	41	0.0
	10	6756	6200	556	0.0	0.0	33	5.2
	30	6472	5950	522	0.0	0.0	32	29.9
	50	6464	5950	514	0.0	0.0	35	36.4
	100	6464	5950	514	0.0	0.0	55	40.3
V40-N269	0	14184	12950	1234	0.0	0.0	141	0.0
	10	13468	12250	1218	0.0	-3.5	90	9.3
	30	13035	11600	1435	0.1	-3.1	79	29.6
	50	12883	11550	1333	0.5	-7.2	105	44.0
	100	12883	11550	1333	0.6	-3.7	79	44.0
V46-N192	0	12702	11450	1252	0.0	0.0	128	0.0
	10	12374	11200	1174	0.0	-2.2	129	9.7
	30	11698	10500	1198	1.8	-10.9	102	27.9
	50	11258	10050	1208	0.0	-9.2	96	50.0
	100	11258	10050	1208	0.0	-9.1	98	44.8
V50-N270	0	14010	12700	1310	0.0	-1.7	140	0.0
	10	13861	12500	1361	0.0	-6.8	135	9.7
	30	13515	12250	1265	0.1	-7.9	126	29.2
	50	13515	12250	1265	0.1	-5.4	121	29.6
	100	13515	12250	1265	0.1	-9.4	121	30.1
V51-N234	0	14912	13600	1312	0.0	-2.8	188	0.0
	10	13218	11950	1268	2.0	-15.5	199	9.6
	30	12424	11250	1174	0.0	-18.0	123	25.0
	50	12424	11250	1174	0.0	-12.8	121	35.1
	100	12424	11250	1174	0.0	-9.0	127	35.1
V51-N283	0	16239	14850	1389	0.0	-0.3	181	0.0
	10	14808	13450	1358	0.1	-5.2	153	9.3
	30	13880	12500	1380	0.7	-12.4	239	29.5
	50	13845	12500	1345	0.1	-10.0	132	37.9
	100	13678	12250	1428	0.1	-15.1	133	55.9

Data	β(%)	Obj	C^f	C^d	Gap'	Gap	T(s)	$\hat{s}$
V54-N321	0	17328	15800	1528	0.0	-0.1	263	0.0
	10	16592	15100	1492	0.0	-4.3	314	7.4
	30	15820	14200	1620	0.1	-6.9	234	29.6
	50	15692	14150	1542	0.3	-3.4	292	40.5
	100	15694	14150	1544	0.2	-7.2	217	50.6
V55-N318	0	17823	16300	1523	0.0	0.0	295	0.0
	10	15860	14400	1460	0.2	-7.0	239	9.4
	30	15140	13700	1440	1.9	-11.9	235	29.8
	50	15025	13500	1525	0.6	-13.4	210	39.2
	100	15030	13500	1530	1.8	-8.3	281	51.8
V57-N336	0	18210	16750	1460	0.0	2.1	444	0.0
	10	16784	15350	1434	0.0	-7.8	369	8.9
	30	16202	14650	1552	0.0	-9.0	248	27.5
	50	15745	14200	1545	1.6	-11.8	366	43.1
	100	15795	14200	1595	1.1	-13.8	242	47.6
V57-N341	0	17916	16350	1566	0.0	-3.3	270	0.0
	10	16970	15400	1570	0.7	-8.2	257	9.5
	30	16310	14700	1610	0.6	-12.6	305	28.2
	50	16215	14500	1715	0.2	-12.5	232	39.9
	100	16215	14500	1715	0.2	-10.7	250	39.9
V57-N347	0	18324	16800	1524	0.9	-0.3	358	0.0
	10	16950	15400	1550	0.2	-8.9	296	8.6
	30	16290	14700	1590	1.4	-11.0	241	28.8
	50	16245	14700	1545	0.1	-13.6	339	42.4
	100	16185	14500	1685	0.1	-9.4	213	46.8
V60-N281	0	16622	15050	1572	0.1	-1.2	306	0.0
	10	15122	13650	1472	0.9	-9.4	276	9.8
	30	14364	12950	1414	0.1	-6.8	231	28.4
	50	14364	12950	1414	0.0	-8.6	252	27.1
	100	14364	12950	1414	0.0	-12.9	279	27.1
V61-N337	0	18684	17000	1684	0.5	-1.8	428	0.0
	10	17246	15600	1646	1.0	-11.0	302	10.0
	30	16580	14900	1680	0.3	-11.3	329	29.3
	50	16420	14700	1720	0.2	-14.1	396	46.3
	100	16385	14700	1685	0.3	-10.0	240	51.5
V66-N321	0	18008	16300	1708	0.2	-6.4	355	0.0
	10	16728	15100	1628	0.2	-17.2	346	8.6
	30	16070	14400	1670	0.1	-18.0	313	27.6
	50	16039	14400	1639	0.5	-19.9	314	35.8
	100	16039	14400	1639	0.7	-17.1	323	35.8
V72-N381	0	20336	18450	1886	2.1	-2.1	454	0.0
	10	19125	17300	1825	1.0	-12.7	433	9.8
	30	18465	16600	1865	0.8	-16.5	364	29.2
	50	18455	16600	1855	0.5	-15.5	464	33.4
	100	18455	16600	1855	0.1	-15.5	319	34.1
V77-N409	0	21190	19200	1990	0.6	-8.2	503	0.0
	10	19870	18000	1870	1.2	-17.7	437	9.5
	30	19555	17600	1955	0.2	-20.7	391	23.2
	50	19311	17350	1961	0.5	-20.1	430	44.8
	100	19530	17600	1930	0.1	-17.9	439	30.8
V78-N391	0	20630	18700	1930	1.1	-7.2	570	0.0
	10	19614	17750	1864	1.1	-18.0	598	9.3
	30	19045	17100	1945	0.3	-20.4	444	28.4
	50	18819	16850	1969	0.9	-20.2	453	35.5
	100	18827	16850	1977	0.4	-18.4	387	35.1
V106-N495	0	27038	24450	2588	0.5	-15.0	1239	0.0
	10	25388	23000	2388	1.9	-20.6	1096	9.6
	30	24233	21850	2383	1.0	-23.3	1092	29.3
	50	23960	21600	2360	1.1	-23.9	956	38.4
	100	23960	21600	2360	0.4	-23.1	780	40.2

Table 4: Results of meta-heuristic for artificially generated instances

It can be observed that our exact algorithm can solve to optimality only 27 instances, specifically those of smaller size with fewer than 46 bus stops. Its performance tends to deteriorate as the number of bus stops and students increases. This behavior is easily predicted as the size of the MILP formulation increases. The instances with $\beta = 0$ tends to be the easier to solve. When $\beta = 0$, the MILP model can solve all datasets up to the instance V46-N192. One possible reason is that the strict limitation on β results in solutions for the next semester that are not significantly different from those of the current semester, reducing the search space.

We now examine the performance of the MS-ILS algorithm in comparison to the exact algorithm. Table 4 presents the results of the metaheuristic with the same metrics as presented in 3, with two key differences: (1) Column "Gap'" represents the percentage deviation of the metaheuristic solutions over 10 runs from the best solution found, and (2) Column "*Gap*" now indicates the percentage gap between the solution obtained by the MS-ILS algorithm and the best known solution from the MILP formulation. The MS-ILS successfully matches the solutions in all 27 instances where the MILP obtains the optimal solutions within the time limit. For the remaining 93 instances, the solutions provided by MS-ILS are at least as good as the upper bounds obtained by the exact algorithm, and better in 87 of these instances. This improvement is particularly significant for instances with more than 60 bus stops, where the gap consistently exceeds 10%.

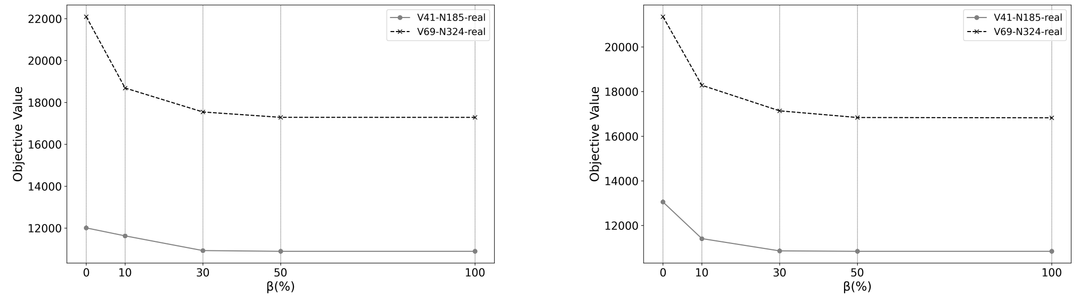
The small values in Columns "Gap'" (never exceeding 2.1%) demonstrate the stability of our method over multiple runs. In terms of running times, the MS-ILS algorithm runs in less than 21 minutes for all instances, which is significantly faster than the exact method. Given that the SOBRGP is required to re-solve once a semester, these results demonstrates that the hybrid algorithm is capable of providing solutions of very good quality within a short computing time.

We now turn our attention to the impact of the parameter β. First, the objective value of solutions obtained by MS-ILS generally decreases as the value of β increases. This is because relaxing the constraint on the maximum number of negatively impacted students provides more flexibility in assigning bus stops to each route. Secondly, the solutions of the instances with $\beta \geq 50\%$ are quite similar for almost all cases. Closer inspection of the output revealed that the quantity $\hat{s}$ plays a key role. If the solution for the case $\beta = 100\%$ has fewer than 50% of the students experiencing a negative impact, it is also a solution for the case where $\beta = 50\%$.

6.4. Computational results for real instances

Since the exact formulation yields low-quality solutions for test instances with more than 40 bus stops, we decided to use the MS-ILS in this experiment. We also experiment with different values of β, following a similar experimental design as with the test instances, since the cost savings information for each β value can benefit the school in negotiations with parents.

In addition to solving the scheduling problem for the next semester, the operations manager of the school also wants to check the current manually arranged schedule to consider the feasibility of using an automatically generated schedule as a replacement. Therefore, we have conducted two scenarios. In the first scenario, we keep the number of students at each bus stops and the locations of the bus stops unchanged, whereas in the second scenario, we generate new data instances for the next semester based on real instances, using a similar approach to artificially generated instances to account for



(a) The scenario when the students and bus stops are kept unchanged

(b) The scenario when the students and bus stops are modified.

Figure 3.: Objective values when β varies.

fluctuations in student bus registrations.

Figure 3 presents the objective value for both scenarios with different values of β. One can see that relaxing the negative impact constraints by increasing β from 0% to 30% significantly reduces the total cost. However, further increases in β beyond 30% have little effect until β reaches 50%, at which point no additional cost reduction is observed. Therefore, from a cost-effectiveness perspective, the ideal β value falls between 50% and 100%. However, a β value of 30% appears to be a suitable compromise and is likely to achieve a mutual agreement between parents and the school. More discussion on the value of β is provided in Subsection 6.7.

We now turn our attention to the number of buses required, their composition and travel cost. These results are summarised in Table 5 for the first scenario and Table 6 for the second scenario. The first and second columns show the instance names and β values, respectively. Note that rows where β is denoted by a hyphen (-) represent the current semester's manually created schedule. The remaining columns display the following for each solution: the number of used buses by type (Columns "$|\mathcal{M}_{15}|$", "$|\mathcal{M}_{28}|$", and "$|\mathcal{M}_{44}|$"), the total number of unoccupied seats (Column "Emp"), travel cost, the proportion of students experiencing a negative impact, and the running time of MS-ILS in seconds.

One can see that, for the V41-N185-real instance, the manual schedule incurs a lower travel cost than our solution, even when the negative impact constraints are fully relaxed ($\beta = 100\%$). However, inefficient bus utilisation and suboptimal route design result in a high number of empty seats. The reliance on higher-capacity buses further increases fixed costs, leading to a higher overall cost. In contrast, for the V69-N324-real instance, our solution yields both lower travel and total costs, regardless of how restrictive the negative impact constraints are. A further investigation of the output revealed that the current schedules violate constraints on the maximum number of bus stops per route and maximum student riding time, highlighting the limitations of manual scheduling in ensuring feasibility and solution quality.

When β is small, the negative impact constraints become highly restrictive, limiting the flexibility of route planning. As a result, buses are often underutilised, leading to more empty seats and higher travel cost. In contrast, as β increases, these constraints become more relaxed, allowing for greater flexibility in designing routes that cover more stops and serve more passengers, which can help reduce travel costs. Consequently, larger buses are used more often, and the overall number of vehicles required is reduced.

However, one can see that buses with a capacity of 45 seats are rarely used. This suggests that although covering more stops per route can yield cost savings, those savings do not justify the high fixed costs associated with operating such large buses. Additionally, a closer inspection on the output showed that when β exceeds 30%, the maximum riding duration constraints become more influential than the negative impact constraint in determining bus sizes and routes.

Lastly, the running time of the metaheuristic on the real instances is always under 11 minutes, demonstrating its practical applicability.

Data	$\beta(\%)$	$\lvert\mathcal{M}_{15}\rvert$	$\lvert\mathcal{M}_{28}\rvert$	$\lvert\mathcal{M}_{44}\rvert$	Emp	C^d	$\hat{s}$	T(s)
V41-N185-real	-	1	8	0	54	1078	-	-
	0	5	5	0	30	1596	0.0	88.6
	10	6	4	0	17	1488	9.7	101.0
	30	4	5	0	15	1494	29.7	91.7
	50	4	5	0	15	1488	46.5	83.8
	100	4	5	0	15	1486	46.5	96.6
V69-N324-real	-	3	11	1	73	1926	-	-
	0	14	6	0	54	1258	0.0	604.7
	10	8	8	0	20	1128	9.0	497.0
	30	3	11	0	29	1126	29.3	458.1
	50	4	10	0	16	1090	43.8	417.4
	100	4	10	0	16	1090	54.6	357.7

Table 5: Results for real instances when the students and bus stops are kept unchanged

Data	$\beta(\%)$	$\lvert\mathcal{M}_{15}\rvert$	$\lvert\mathcal{M}_{28}\rvert$	$\lvert\mathcal{M}_{44}\rvert$	Emp	C^d	$\hat{s}$	T(s)
V41-N185-real	-	1	8	0	54	1078	-	-
	0	10	2	0	21	1556	0.0	184.8
	10	7	3	0	4	1532	9.5	106.3
	30	4	5	0	15	1537	27.0	126.3
	50	4	5	0	15	1488	42.6	135.7
	100	4	5	0	15	1475	47.3	131.0
V69-N324-real	-	3	11	1	73	1926	-	-
	0	12	7	0	52	1154	0.0	583.0
	10	7	7	1	21	1164	10.0	499.4
	30	4	7	2	20	1064	27.3	438.8
	50	3	9	1	17	1044	39.6	436.4
	100	5	6	2	7	1044	51.9	430.1

Table 6: Results for real instances when the students and bus stops are modified

6.5. Sensitivity analysis of α

As highlighted in Subsection 6.4, the maximum riding duration constraints relative to direct travel time play a more significant role in bus route design than the negative impact constraint when β exceeds 30% for real instances. These constraints limit a student's bus ride to at most α times their direct travel time from their bus stop to school. To provide further insight, we now conduct a sensitivity analysis on α.

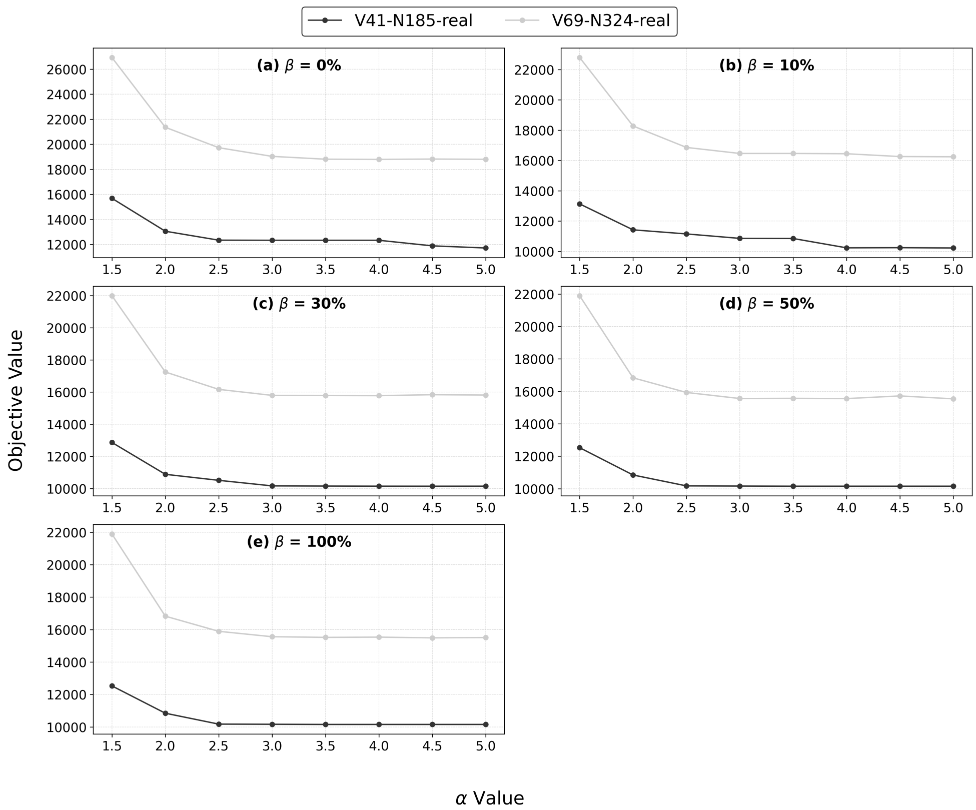


Figure 4.: Solutions obtained by the meta-heuristic for real instances under varying α parameters.

Figure 4 presents total costs for two real instances across eight values of $\alpha \in \{1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ and five values of $\beta \in \{0\%, 10\%, 30\%, 50\%, 100\%\}$. We note that we set β is set to 2 in our default experiment.

A significant decrease in total costs is observed as α increases from 1.5 to 2.5. This confirms our previous observations for $\alpha = 2$, where some students' riding times reached the upper limit of approximately α times their direct travel time in the obtained solutions. Consequently, even a slight relaxation of this constraint led to a substantial reduction in total costs.

When $\alpha \geq 3$ and $\beta \geq 30\%$, the total costs remain stable. A closer inspection of the results revealed that, at these values, both the maximum riding duration constraints and the negative impact constraint are no longer restrictive.

However, when $\beta \leq 30\%$, two sudden drops in total costs are observed. The first occurs when $\beta = 10\%$ as α increases from 3.5 to 4, and the second when $\beta = 0\%$ as α increases from 4 to 4.5. A closer inspection of data from the previous semester indicates that four bus stops had students experiencing travel times between 4.6 and 5.8 times their direct travel time. Therefore, relaxing their maximum riding duration in the next semester to a certain threshold would help reduce total costs. For $\beta = 0\%$, this relaxation could improve total costs without negatively impacting students. Similarly, for $\beta = 10\%$, this relaxation occurs without violating the negative impact constraint.

Remarkably, when $\beta \in \{0\%, 10\%\}$, the negative impact constraint has a slightly more significant effect on total costs compared to the maximum riding duration constraints. Closer inspection revealed that about half of the bus stops serve students whose travel times in the previous semester were already below 1.5 times their direct travel time.

6.6. Sensitivity analysis of L and γ

For completeness, we conduct a sensitivity analysis on the values of L and γ across different values of $\beta \in \{0\%, 10\%, 30\%, 50\%, 100\%\}$. Figures 5 and 6 show the resulting total costs on different values of L and γ, respectively.

Number of stops per route

In our default scenario, $L = 7$, meaning that each bus route can have at most seven stops. We explore the effect of changing L to 4, 5, 6, 8, and 9.

The results show that total costs remain stable when $L \geq 7$. A closer inspection of the output suggests that the maximum number of stops per route is not a binding constraint in the obtained solutions. In these solutions, most buses pass through only 3 to 5 stops, with only a few reaching 6 stops at most. A possible explanation is that other constraints, such as the maximum riding duration relative to direct travel time, the negative impact constraints, and the maximum bus route duration (fixed at 1 hour), play a more significant role in limiting route length. Additionally, the high fixed cost of a 45-seat bus may further discourage longer routes. The above observation also helps explain why increasing L from 4 to 6 generally reduces total costs as some routes are constrained by the limit on the number of stops.

A more significant drop in total costs is observed as the value of β increases. This occurs because a relaxed negative impact constraint allows for greater flexibility in redesigning routes.

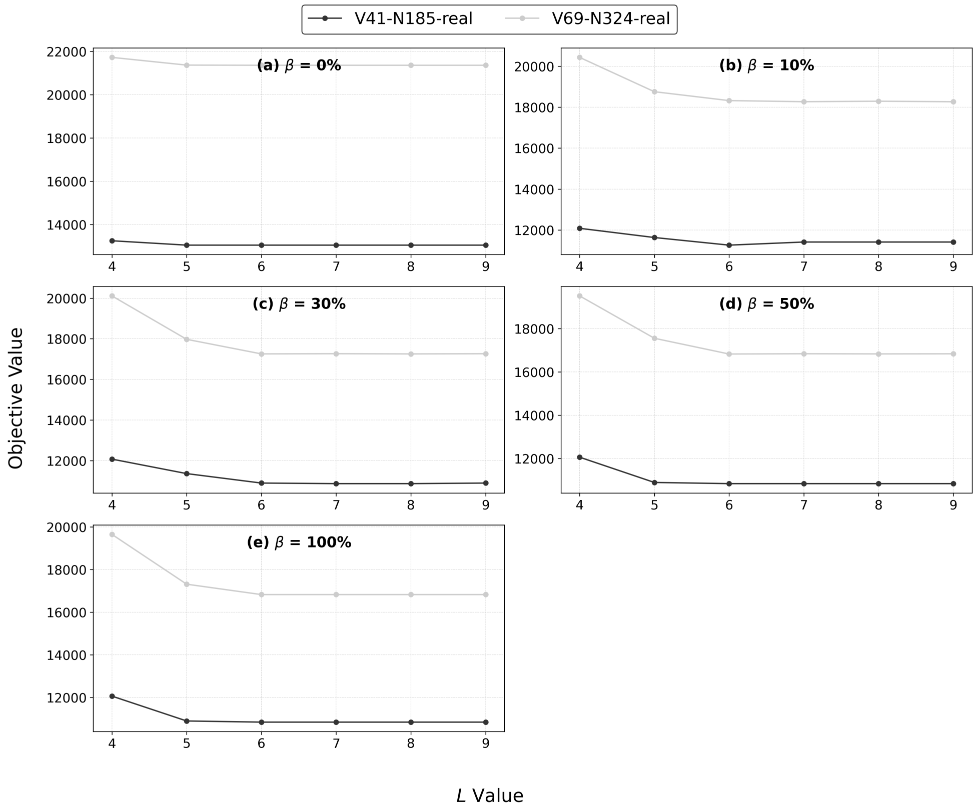


Figure 5.: Solutions obtained by the meta-heuristic for real instances with varying L parameters.

Negative Impact Condition

In our default scenarios, $\gamma = 1.05$, which means if a student travel exceeds 1.05 times their previous semester's travel time, they are considered negatively impacted. We now explore the effect of various different values for $\gamma \in \{1.01, 1.25, 1.45, 1.65, 1.85\}$.

When $\beta \geq 50\%$ or $\gamma \geq 1.65$, the total cost remains stable. This suggests that increasing β beyond 50% or γ beyond 1.65 does not provide additional flexibility in bus route design. In other words, the constraints are already non-restrictive at these values, so further increases do not impact the total cost. A possible explanation is that other constraints now more actively influence the solution, and the objective function already keeps total costs as low as possible at these values.

In contrast, when both β and γ are smaller, total costs decrease as expected, with a more significant reduction observed when γ is between 1.05 and 1.25. However, when γ is between 1.01 and 1.05, total costs exhibit slight fluctuations. This suggests that such small changes in γ may not provide enough flexibility to redesign routes efficiently. The small increase in total cost when $\beta = 30\%$ for the V69-N324-real instance may be due to a convergence issue of the metaheuristic.

6.7. Sensitivity analysis of β over time

To provide insight, we present the total cost for each β value from the set $\{0\%, 10\%, 30\%, 50\%, 100\%\}$ over the next six semesters in Table 7, based on real-

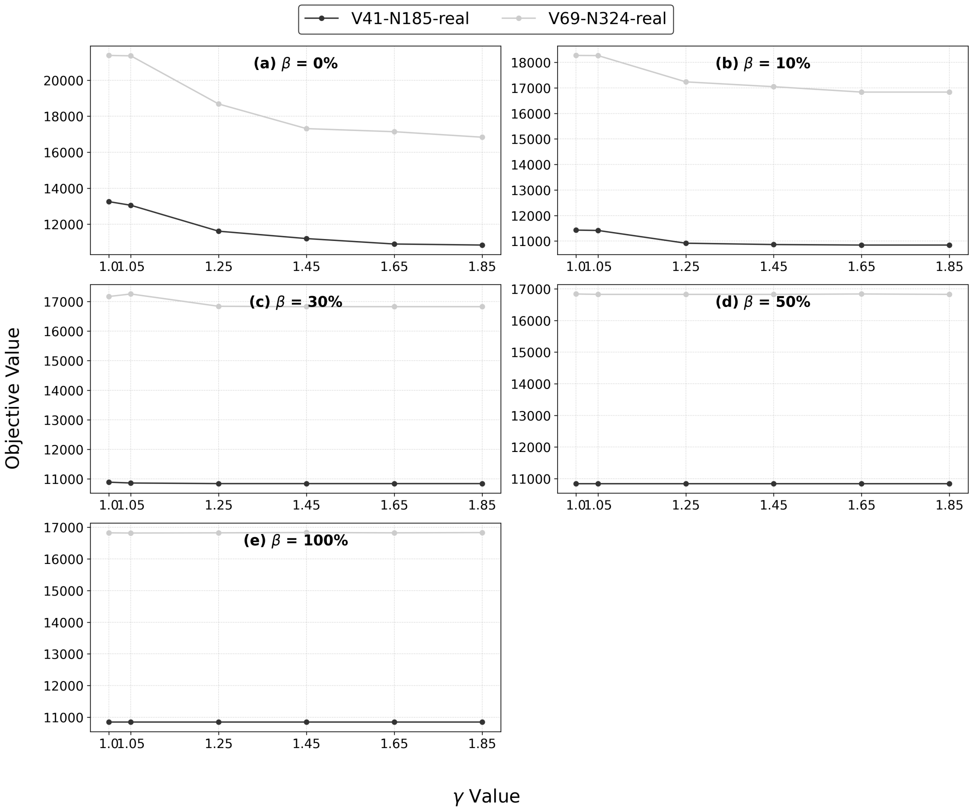


Figure 6.: Solutions obtained by the meta-heuristic for real instances under varying γ parameters.

world instances. Additionally, we calculate the percentage increase in cost for β values $0\%, 10\%, 30\%$, and 50% relative to $\beta = 100\%$ to examine how incremental costs incurred for the school to meet certain levels of parents' and students' preferences stabilise over time. This also helps assess whether $\beta = 30\%$ remains a good compromise between cost-effectiveness and the restrictiveness of negative impact in the long run. Since the composition of students and the locations of bus stops vary each semester in practice, we focus solely on this scenario. The experimental results are presented in Table 7.

For $\beta = 0\%$ and $\beta = 10\%$, the incremental cost percentage generally increases over semesters. A possible explanation is that 80% of students are set to continue using the bus system in the next semester, and a highly restrictive negative impact policy reduces flexibility in bus route planning over time. This trend does not appear when β is at least 30%, suggesting that higher β values allow for more adaptable solutions in later semesters.

Among the tested values, $\beta = 30\%$ still remains the best compromise. Lower β values result in considerable cost increases. Specifically, setting $\beta = 0\%$ leads to 17.2–20.6% higher costs for V41-N185-real instance and 26.9–43.8% higher costs for V69-N324-real instance compared to a scenario without a negative impact policy. Similarly, setting $\beta = 10\%$ increases costs by 5.3–8.8% for V41-N185-real instance and 5.5–11.6% for V69-N324-real instance. While $\beta = 50\%$ does reduce costs slightly, the savings are too small to justify the reduction in parents and students' satisfaction. Specifically, $\beta = 50\%$ saves, on average, only 0.35% for V41-N185-real instance and 1.3% for V69-N324-real instance compared to $\beta = 30\%$.

Data	β(%)	2nd		3rd		4th		5th		6th		7th	
		obj	s(%)	obj	s(%)	obj	s(%)	obj	s(%)	obj	s(%)	obj	s(%)
V41-N185-real	0	13054	-20.4	12526	-17.2	12776	-20.1	12534	-19.9	12868	-20.2	13072	-20.6
	10	11418	-5.3	11586	-8.4	11570	-8.8	11302	-8.1	11564	-8.1	11588	-6.9
	30	10892	-0.4	10692	0.0	10634	0.0	10640	-1.8	10856	-1.4	10838	0.0
	50	10844	0.0	10692	0.0	10634	0.0	10466	-0.1	10856	-1.4	10838	0.0
	100	10844	-	10692	-	10634	-	10456	-	10702	-	10838	-
V69-N324-real	0	21356	-26.9	21836	-29.8	22382	-35.0	23754	-43.8	23414	-42.0	23452	-42.1
	10	18326	-8.9	17734	-5.5	18501	-11.6	18167	-10.0	18138	-10.0	17854	-8.2
	30	17253	-2.5	16873	-0.3	16806	-1.3	17088	-3.5	16487	0.0	16749	-1.5
	50	16835	-0.1	16812	0.0	16789	-1.2	16508	0.1	16487	0.0	16514	-0.1
	100	16825	-	16817	-	16584	-	16518	-	16487	-	16504	-

Table 7: Total costs and percentage incremental costs for different β values over 6 semesters.

7. Conclusion

In this study, we introduce a new variant of the BRGP that has not been previously addressed in the literature. Our objective is to determine the composition of the school bus fleet and routes for transporting students to school at minimal cost, subject to constraints including vehicle capacity, the maximum route duration, the maximum number of stops per route, the maximum travel time for each student, and the maximum number of students experiencing negative impacts. This final constraint is inspired by a new requirement implemented in transportation systems of several private high schools in Vietnam. The problem is broadly defined, and we believe that similar challenges may arise at schools globally when considering student experiences.

We presented a mathematical formulation for the problem and developed a hybrid approach by combining the GRASP and ILS meta-heuristics. The effectiveness of the hybrid algorithm was demonstrated through extensive computational experiments on artificially generated test instances, and its applicability was illustrated on two real instances. A potential direction for future research is to explore the development of an exact solution method aimed at generating improved solutions for medium-to-large-scale instances.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was funded by Vingroup Innovation Foundation (VINIF), Vietnam under project code VINIF.2024.DA072.

References

[1] C. Alabas-Uslu. A self-tuning heuristic for a multi-objective vehicle routing problem. *Journal of the Operational Research Society*, 59(7):988–996, 2008.

[2] A. Aljundi, M. Al Khatib, M. Kharbeche, and M. Haouari. Solving a bus routing problem arising in doha. In *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 0600–0604. IEEE, 2022.

[3] J. S. Arias-Rojas, J. F. Jiménez, and J. R. Montoya-Torres. Solving of school bus routing problem by ant colony optimization. *Revista EIA*, pages 193–208, 2012.

[4] P. Augerat. *Approche Polyèdrale du Problème de Tournées de Véhicules*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1995.

[5] J. E. Beasley. Route first—cluster second methods for vehicle routing. *Omega*, 11(4):403–408, 1983.

[6] L. D. Bodin. A taxonomic structure for vehicle routing and scheduling problems. *Computers & Urban Society*, 1(1):11–29, 1975.

[7] R. Bowerman, B. Hall, and P. Calamai. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research Part A: Policy and Practice*, 29(2):107–123, 1995.

[8] J. Brandão. A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 157(3):552–564, 2004.

[9] L. Chapleau, J. A. Ferland, and J. M. Rousseau. Clustering for routing in densely populated areas. *European Journal of Operational Research*, 20(1):48–57, 1985.

[10] A. Corberán, E. Fernández, M. Laguna, and R. Marti. Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society*, 53(4):427–435, 2002.

[11] E. Dasdemir, M. C. Testik, D. T. Öztürk, C. T. Şakar, G. Güleryüz, and Ö. M. Testik. A multi-objective open vehicle routing problem with overbooking: Exact and heuristic solution approaches for an employee transportation problem. *Omega*, 108:102587, 2022.

[12] C. R. Delgado Serna and J. Pacheco Bonrostro. Minmax vehicle routing problems: application to school transport in the province of burgos. *Computer-aided Scheduling of Public Transport*, pages 297–317, 2001.

[13] S. Deng, Z. He, J. Zhong, and J. Xie. Bus rescheduling for long-term benefits: An integrated model focusing on service capability and regularity. *Applied Sciences*, 14(5), 2024.

[14] U. Derigs and K. Reuter. A simple and efficient tabu search heuristic for solving the open vehicle routing problem. *Journal of the Operational Research Society*, 60(12):1658–1669, 2009.

[15] O. Díaz-Parra, J. A. Ruiz-Vanoye, M. de los Ángeles Buenabad-Arias, and A. C. Saenz. Vertical transfer algorithm for the school bus routing problem. *Transactions on Computational Science XXI: Special Issue on Innovations in Nature-Inspired Computing and Applications*, pages 211–229, 2013.

[16] G. Dulac, J. A. Ferland, and P. A. Forgues. School bus routes generator in urban surroundings. *Computers & Operations Research*, 7(3):199–213, 1980.

[17] K. A. Eldrandaly and A. F. Abdallah. A novel gis-based decision-making framework for the school bus routing problem. *Geo-spatial Information Science*, 15(1):51–59, 2012.

[18] W. A. Ellegood, S. Solomon, J. North, and J. F. Campbell. School bus routing problem: Contemporary trends and research directions. *Omega*, 95:102056, 2020.

[19] G. S. Er and V. Dhir. Open vehicle routing problem by ant colony optimization. *International Journal of Advanced Computer Science and Applications*, 5(3), 2014.

[20] J. Euchi and R. Mraihi. The urban bus routing problem in the tunisian case by the hybrid artificial ant colony algorithm. *Swarm and Evolutionary Computation*, 2:15–24, 2012.

[21] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[22] P. Festa and M. G. C. Resende. An annotated bibliography of GRASP–Part I: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009.

[23] P. Festa and M. G. C. Resende. An annotated bibliography of GRASP–Part II: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.

[24] M. L. Fisher and R. Jaikumar. The local delivery problem: algorithms and applications. *Publication of: Elsevier Science Publishers BV*, 1984.

[25] K. Fleszar, I. H. Osman, and K. S. Hindi. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3):803–809, 2009.

[26] Z. Fu, R. Eglese, and L. Y. Li. A new tabu search heuristic for the open vehicle routing problem. *Journal of the Operational Research Society*, 56(3):267–274, 2005.

[27] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[28] S. R. A. Haddadene, N. Labadie, and C. Prodhon. A GRASP $\times$ ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems with Applications*, 66:274–294, 2016.

[29] Y. Hou, W. Gu, C. Wang, K. Yang, and Y. Wang. A selection hyper-heuristic based on q-learning for school bus routing problem. *IAENG International Journal of Applied Mathematics*, 52(4), 2022.

[30] M. H. Hà, N. Bostel, A. Langevin, and L-M. Rousseau. An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. *Computers & Operations Research*, 43:9–19, 2014.

[31] K. M. Kotoula, M. Morfoulaki, G. Aifadopoulou, and P. Tzenos. Calculating optimal school bus routing and its impact on safety and the environment. *Transportation Research Record*, 2647(1):142–150, 2017.

[32] F. Li, B. Golden, and E. Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34(10):2918–2930, 2007.
[33] J. Q. Li, D. Borenstein, and P. B. Mirchandani. A decision support system for the single-depot vehicle rescheduling problem. *Computers & Operations Research*, 34(4):1008–1032, 2007.
[34] J. Q. Li, P. B. Mirchandani, and D. Borenstein. Parallel auction algorithm for bus rescheduling. In *Proceedings of ninth international conference on computer-aided scheduling of public transport*, 2004.
[35] L. Y. Li and Z. Fu. The school bus routing problem: a case study. *Journal of the Operational Research Society*, 53:552–558, 2002.
[36] A. D. López-Sánchez, A. G. Hernández-Díaz, R. Vigo, D.and Caballero, and J. Molina. A multi-start algorithm for a balanced real-world open vehicle routing problem. *European Journal of Operational Research*, 238(1):104–113, 2014.
[37] H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search. In *Handbook of Metaheuristics*, pages 320–353. Springer US, Boston, MA, 2003.
[38] H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search: Framework and applications. *Handbook of Metaheuristics*, pages 129–168, 2019.
[39] B. Minocha and S. Tripathi. Solving school bus routing problem using hybrid genetic algorithm: a case study. In *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*, pages 93–103. Springer, 2014.
[40] R. M. Newton and W. H. Thomas. Design of school bus routes by computer. *Socio-Economic Planning Sciences*, 3(1):75–85, 1969.
[41] B. Ngonyani, E. Mujuni, and A. Mushi. Optimizing schedules for school bus routing problem: the case of Dar Es Salaam schools. *International Journal of Advanced Research in Computer Science*, 6(1), 2015.
[42] J. Pacheco, R. Caballero, M. Laguna, and J. Molina. Bi-objective bus routing: an application to school buses in rural areas. *Transportation Science*, 47(3):397–411, 2013.
[43] J. Pacheco and R. Martí. Tabu search for a multi-objective routing problem. *Journal of the Operational Research Society*, 57:29–37, 2006.
[44] J. Park and B. I. Kim. The school bus routing problem: A review. *European Journal of Operational Research*, 202(2):311–319, 2010.
[45] T. Pichpibul and R. Kawtummachai. A heuristic approach based on clarke-wright algorithm for open vehicle routing problem. *The Scientific World Journal*, 2013, 2013.
[46] L. S. Pitsoulis and M. G. C. Resende. Greedy randomized adaptive search procedures. *Handbook of Applied Optimization*, pages 168–183, 2002.
[47] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31:1985–2002, 2004.
[48] C. Prins. A GRASP × Evolutionary Local Search hybrid for the vehicle routing problem. *Bio-inspired Algorithms for the Vehicle Routing Problem*, 161:35–53, 2016.
[49] P. P. Repoussis, C. D. Tarantilis, O. Bräysy, and G. Ioannou. A hybrid evolution strategy for the open vehicle routing problem. *Computers & Operations Research*, 37(3):443–455, 2010.
[50] M. G. C. Resende and C. C. Ribeiro. Greedy randomized adaptive search procedures. *Handbook of Metaheuristics, International Series in Operations Research & Management Science*, 57:219–249, 2003.
[51] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis. An analysis of several heuristics

for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

[52] S. B. Sghaier, N. B. Guedria, and R. Mraihi. Solving school bus routing problem with genetic algorithm. In *2013 International Conference on Advanced Logistics and Transport*, pages 7–12. IEEE, 2013.

[53] M.S. Visentini, D. Borenstein, and J. Q. Li. Review of real-time vehicle schedule recovery methods in transportation services. *Journal of Scheduling*, 17:541–567, 2014.

[54] Y. Wang and P. Shang. Shuttle bus rerouting and rescheduling problem considering daily demand fluctuation. *Mathematical Problems in Engineering*, 2022(1):2917240, 2022.

[55] E. E. Zachariadis and C. T. Kiranoudis. An open vehicle routing problem metaheuristic for examining wide solution neighborhoods. *Computers & Operations Research*, 37(4):712–723, 2010.