ANT/OR - University of Antwerp Operations Research Group
Faculty of Applied Economic

Metaheuristics for school bus routing problem

Thesis submitted for the Degree of Doctor in Department of Engineering Management at the University of Antwerp to be defended by **Mohammad Saied Fallah Niasar**

Supervisors:
Prof. Kenneth Sörensen
Prof. dr. Mehdi Sajadifar

March 2024

Acknowledgement

First and foremost, I would like to express my gratitude to my supervisors Professors Kenneth Sörensen, Johan Springael, and Mehdi Sajadi for their inspiring guidance, constructive criticism, and thoughtful advice and feedback during these years. Their enormous knowledge and experience have invigorated me throughout my academic research and in my daily life. Moreover, they were always so kind and available even for a small talk. My gratitude extends to the staff of the Faculty of Applied Economics for their continuous guidance during these years. I would also like to thank my cherished wife Elnaz who supported me in writing this thesis, and encouraged me to reach this stage. She always supported me whenever I was alone and needed to take important decisions. Last but not least, I appreciate my parents and my friends Amir and Ali. Without their understanding and encouragement in these years, it would be impossible for me to finish my PhD studies.

Abstract

This thesis focusses on a special part of the supply chain that is relevant to the student transportation problem. Considering previous studies, it appears that addressing emergent issues such as increased traffic load, high student population, lack of resources, safety, and risks can play a substantial role in designing an efficient plan for the student transportation system. The significance of this issue is highlighted when we take into account the needs and expectations of all stakeholders, including students, the private sector, and municipalities. In this regard, this dissertation considers a number of realistic and innovative characteristics for the school bus routing problem (SBRP). In doing so, two main trajectories have been followed. First, the existing gap and concerns in the literature and real life are considered to extract a new model-based variant of SBRP characteristics.

Second, an attempt is made to construct proper metaheuristic algorithms (solution approaches) to efficiently solve the problems identified in the first phase. To put it more clearly, in the first trajectory, we consider different problem features and propose new model and problem for SBRP, and in the second trajectory, we design and construct a metaheuristic approach germane to the defined problem. To do so, there are challenges that must be addressed concerning how to design an appropriate metaheuristic that corresponds to the specific type of problem and makes a trade-off between computing time and solution quality as well as a trade-off between intensification and diversification.

Following the above phases entails two advantages. It helps the decision-maker in urban planning to adopt the right course of action and presents alternatives in choosing the appropriate solution approach. In the first and second chapters of this thesis, the existing school bus routing problems along with different kinds of solution approaches are discussed, while in chapters three to five a new model and, correspondingly, new metaheuristics are presented. In other words, the first two chapters we present new solution approaches for the existing current problem, and in the remaining chapters we explore and attempt at a new model as well as solution approaches.

Briefly, regarding the solution approach, we have considered the strategic oscillation (searching between feasible and infeasible parts of the solution space), different large neighborhood search algorithms (presenting different kinds of removal and insertion heuristics), neighborhood selection mechanisms, and a number of diversification strategies.

We have also developed new mathematical models for SBRP that consider mixed-load effect, transporting morning and afternoon students, and the existing risks of student transportation. Further analyses are also executed to address real life concerns.

List of Topics

List of Figures

List of Tables

List of Algorithms

Introduction

Introduction

Urban logistics, a key aspect of city planning, faces challenges in sustainability and mobility due to inadequate identification and analysis in contemporary cities. Traffic congestion is a widespread issue in large cities, with its volume increasing daily due to rising vehicle numbers. Integrating safety, environmental, and social considerations adds complexity to transportation planning when employing cost-benefit approaches. Especially in urban areas with intricate transportation systems, careful attention is necessary. Concerns vary based on customer intentions. Identifying stakeholder needs early is crucial for effective transportation planning. Public transportation emerges as an effective strategy, offering benefits like reduced pollution, increased safety, and fewer accidents compared to other modes, proving cost-efficient and environmentally friendly.

Efficient student transportation scheduling is crucial in public transit, drawing special attention. Ensuring students' safe onboarding/offboarding is a top community priority. To promote student use of municipal or community-provided transportation, a safe and convenient system is crucial.

Recently, a confluence of factors, including traffic management control, environmental considerations, technological advancements (such as tracking systems), and urban policies, has prompted an increasing number of schools to transition toward the utilization of public transportation systems.

Inadequate bus network planning can lead to negative outcomes like increased pollution, noise, accidents, lower satisfaction, and higher expenses. Rising fuel costs and traffic push families to opt for public transportation for their children's school commute.

Efficient student assignment and route planning can significantly reduce costs and enhance services. School bus services require consideration of parameters like bus capacity, school start time, and students' maximum riding time.

The school bus routing problem is a recognized practical challenge that impacts major cities worldwide, such as Tehran, the capital of Iran.

Inadequate planning in Tehran schools leads to longer travel times, higher congestion, increased fuel use, and greater fleet depreciation. With buses operating during rush hours, establishing an efficient mechanism is crucial to avoid traffic and generate profit within available resources for the transportation company.

Additionally, as school buses operate during rush hour, establishing an efficient mechanism for student transport is crucial. This involves creating a viable bus route to avoid city traffic and ensure profitability within available resources. The routing should prioritize students' convenience and satisfaction throughout their trip. This thesis focuses on the distribution process in student transportation, a critical part of the supply chain. The increasing traffic load, resource shortages, and high student population highlight the need for integrated school transportation management, doubling the importance of addressing these challenges.

This dissertation addresses various issues discussed in the School Bus Routing Problem (SBRP) literature. Theoretical aims of SBRP involve transporting students safely and conveniently, while real-life applications consider additional constraints and factors related to transportation network serviceability.

Furthermore, the operational attributes of the school bus system add extra intricacies to formulating the SBRP in contrast to the traditional VRP model. (Bowerman and Calamai, 1995).

SBRP is a real-life application of the Vehicle Routing Problem (VRP), extensively studied in operations research (Toth and Vigo, 2002). While traditional VRP finds optimal routes for vehicles, SBRP adapts this to school bus transportation, involving buses traveling from a school to student stops and back.

Similar to the Vehicle Routing Problem (VRP), the School Bus Routing Problem (SBRP) is NP-hard and lacks a polynomial time solution, making exact methods suitable only for smaller instances.

Metaheuristic approaches are essential for managing medium and large instances. The complexity of SBRP underscores the necessity for designing and analyzing it with commercial software offering effective decision-making tools, as seen in VRP and its variants.

Vehicle routing software aligns with academic concepts, utilizing metaheuristic methods, saving algorithms like Clarke and Wright, and multiple neighborhood approaches. Despite logistics optimization software success, there's a notable gap between scientific evidence and practical application, necessitating extensive research for alignment.

This dissertation follows two primary trajectories. Firstly, it explores existing features in the literature and real-life concerns to incorporate into the current model. These include mixed-load mode, multi-load mode, split-load mode, location-allocation, and routing problems, as well as considerations related to risk and hazard.

In the second phase, the focus is on constructing suitable metaheuristic algorithms to efficiently solve the identified problems. The challenge is to design a metaheuristic that aligns with the problem's nature, balancing computing time and solution quality, as well as intensification and diversification. We adopt an approach aligned with the problem's characteristics and scope to ensure an efficient metaheuristic.

More precisely, the primary objective is to identify various characteristics related to the School Bus Routing Problem (SBRP). Secondly, the goal is to advance and develop a metaheuristic approach tailored specifically for this defined problem.

The first chapter delves into a study of previous work, aiming to review various dimensions of the school bus routing problem, including the composite nature of the problem and proposed solutions, and other aspects.

The second and third chapters introduce the established mathematical formulation of the School Bus Routing Problem (SBRP) by Schittekat et al. (2013) along with two distinct metaheuristics. The second chapter presents a simple yet effective approach with three key features: it employs diverse neighborhood structures for thorough search space exploration, efficiently transitions between feasible and infeasible sections, and utilizes a restore operator to navigate back to the feasible solution space when capacity constraint violation increases. The problem consists of three sub-problems: selecting the minimal set of bus stops for each route, allocating students to stops while respecting bus capacity, and defining bus routes to minimize the total distance travelled by all buses.

In the third chapter, metaheuristics based on a strategic oscillation method are proposed, exploring both feasible and infeasible solution spaces. To handle small, medium, and large instances, two configurations are devised:

Simple Large Neighborhood Search (LNS) Configuration: This configuration incorporates only one removal and one insertion heuristic.

Full Adaptive Large Neighborhood Search (ALNS) Configuration: This configuration includes sub-heuristics, and their selection frequency is determined based on their performance.

In the fourth chapter, a novel mathematical model and metaheuristic approach are introduced. This chapter presents a new variant of the School Bus Routing Problem (SBRP) with distinct characteristics: determining the set of possible stops, allocating students to these stops, and creating routes that allow students from different schools to share a bus (mixed-load mode). An optimal mathematical formulation is proposed for solving the problem. Various instances of small, medium, and large sizes are randomly generated to assess the effectiveness of the developed model. To address medium and large instances, an Adaptive Large Neighborhood Search (ALNS) and different removal and insertion configurations are proposed.

In the fifth chapter, we introduce an urban school bus routing problem with features like mixed-load planning, a homogenous fleet, and multi-shift loading. This allows buses to simultaneously transport students from different schools and morning and afternoon shifts. To efficiently solve this problem, we present an iterative local search algorithm, ILS-ANS, with embedded adaptive neighborhood selection. The local search block incorporates three general and five specialized neighborhood structures. Our findings are categorized into two scenarios.

In the first, the proposed heuristic runs on generated instances using a traditional mechanism for local search operators.

In the second scenario, an adaptive mechanism selects and implements operators based on their performance in the current iteration.

In the sixth chapter, the proposed model is developed through two successive mechanisms. Initially, a pre-processing approach is employed to identify risks associated with student safety and health, with the more significant ones being integrated into the model. Subsequently, the focus shifts to mixed-loading, location-allocation-routing problems, and risk analysis. Various metaheuristic configurations are implemented to rationalize the diversification mechanism and neighborhood structures:

Multi-start Structure: Repeating both the constructive phase and local search heuristic for several iterations.

Perturbation Structure: Running the constructive phase once but repeating both the local search and diversification phases for several iterations.

The local search exploration for both configurations involves two mechanisms: traditional order (e.g., VND, starting exploration from lower size neighborhoods and progressing to larger ones) and performance-based (operating according to the adaptive layer mechanism). In summary, the multi-start and perturbation configurations justify the diversification structure, while VND and adaptive layer-based heuristic explain the neighborhood selection mechanism.

The key messages of our Ph.D. trajectory are:

1. Cost reduction: Creating efficient bus routes offers potential cost savings for school districts by minimizing total transportation time and distance.
2. Problem size dependency: As the problem size increases, the algorithm can further decrease total travel time or distance, particularly when the distance between student drop-off and pick-up points is shorter.
3. Adaptability to changing conditions: Solutions to the School Bus Routing Problem must adapt to changing conditions, such as fluctuating school time windows and maximum student riding time.
4. Community engagement: Efficient school bus routes can positively impact the community by reducing traffic congestion, noise, and pollution associated with school transportation.
5. Interdisciplinary collaboration: Addressing the school bus routing problem often involves incorporating expert opinions, particularly crucial for identifying risk factors.
6. Compatibility with efficiency and effectiveness: School bus routing problems can be tailored to criteria like efficiency (minimizing total travel time) and effectiveness (the total travel time spent by students on buses).
7. Student safety: Formulating school bus routing problems can consider health and safety concerns.
8. Dependency on features: Incorporating features like mixed-loads and multi-concepts can significantly reduce total travel time and the number of buses needed.
9. Metaheuristic viewpoints:
    1. Problem-Specific Knowledge Utilization: Leveraging neighborhood-based problem-specific knowledge is a promising approach for generating better solutions.
    2. Metaheuristic Expertise: Acquiring knowledge about a problem is crucial for designing efficient heuristics.
    3. Intensification and Diversification Strategies: Recognizing intensification and diversification as operators, actions, or acceptance criteria strategies in metaheuristics is essential.
    4. Parameter Impact: Identifying parameters significantly affects computing time and solution quality.
    5. ALNS Comparison: ALNS (with capacity constraint violation) demonstrates greater reliability in providing improved solution quality and computation time performance compared to ALNS (with tight capacity constraints).

6. Flexibility with Oscillation Strategy: Metaheuristics with an oscillation strategy offer increased flexibility in the search trajectory.
7. ALNS Implementation Caution: When using ALNS, a cautious approach is necessary, and a tailored course of action should be taken instead of blindly adopting algorithms from other studies.

Savas (1978) discusses three measures for evaluating the performance of public services: efficiency, effectiveness, and equity. Efficiency is defined as the ratio of the level of service to the cost of the resources required to provide such a service. In terms of efficiency, we consider either the total travel time or the total travel distance by bus in our objective function.

The effectiveness of a service is measured by how well the demand is satisfied. In our analysis, we incorporate effectiveness into constraints, such as the total travel time spent by students on buses. Equity considerations assess the fairness or impartiality of the provision of the service; we consider the walking distance of students to bus stops. In addition to equity, we emphasize equilibrium by (1) achieving a balance between routes to avoid excessively large variations in route loading (i.e., ensuring a desirable distribution of students between routes) and (2) assigning a reasonable number of students to each stop.

In the following sections, we delineate the differences across chapters, placing a specific emphasis on the metaheuristic approach utilized and the definition of the problems under consideration.

Chapter 2: In this chapter, we develop Iterated Local Search with the oscillation strategy. This strategy aids in exploring the infeasible part of the solution space without restrictions concerning capacity constraints.

Chapter 3: The adoption of the oscillation strategy necessitates the use of a set of large neighborhood search techniques to facilitate smoother transitions between feasible and infeasible parts of the solution space. Consequently, we introduce different configurations of Large Neighborhood Search (LNS) and Adaptive Large Neighborhood Search (ALNS) metaheuristics with the oscillation strategy.

Chapter 4: This chapter presents the ALNS metaheuristic, albeit with some differences from the one discussed in Chapter 3. These differences include introducing removal and insertion heuristics aligned with problem-specific knowledge, providing pairwise selection of each deletion and insertion heuristic, incorporating a time-saving strategy for insertion heuristics, implementing the meta-destruction operator (if no improvement is made for non-replication), and applying redistribution operators to ensure load balancing.

Chapter 5: In this chapter, we develop two metaheuristics that differ based on the neighborhood selection mechanism for exploration—traditionally (based on size) versus systematically.

Chapter 6: Finally, this chapter introduces different types of metaheuristics that vary in diversification and neighborhood selection mechanisms. The neighborhood scoring mechanism is calculated based on its performance and role in the intensification and diversification mechanism."

Concerning the problem definition, the Chapters 2 and 3 adhere to the problem model proposed by Schittekat et al. (2013).

Chapter 4 introduces augmentations to this model by incorporating additional features such as mixed-load effects and time windows for stops and schools.

Chapter 5 advances the problem by integrating mixed-load and multi-load concepts. Moreover, each student, rather than schools and stops, is now assigned a time window constraint in both shifts. This chapter also explores the split load concept upon the bus's arrival at the stop, and a maximum riding time constraint is introduced to enhance convenience for students.

In Chapter 6, our problem definition evolves further to encompass safety considerations (size of bus stops), health factors (prevalence of coronavirus and population density), and traffic concerns (traffic volume).

Chapter 1:

Literature review

1-1- Introduction

In recent years, a substantial body of literature has explored SBRP from various perspectives. To enhance clarity, our literature reviews are divided into four sections as follows.

Specifically, in section 1.2, the literature review aligns with problem characteristics, objectives, solution methodology, and the composite nature of the problem. In section 1.3, recent works in the area of mixed-load planning are presented (aligned with Chapter 4), and in section 1.4, studies on the morning and afternoon concept for the school bus routing problem are reviewed (compatible with Chapter 5). Finally, in section 1.5, issues related to the safety and health of students are addressed (compatible with Chapter 6).

1-2- Literature review concerning problem characteristics, composite nature and solution approach

In this section various aspects such as, composite nature of problem, considerations of problem characteristics (e.g., multi-schools-multi-depots), objectives (multi-objective function), evaluation methods (e.g., partial evaluation), and mathematical formulations (bilevel approach) are reviewed. For a thorough exploration of the school bus routing problem, Park and Kim (2010) and Ellegood et al. (2019) provide a comprehensive description and a broad survey.

Park and Kim categorize existing literature on SBRP into five key areas: data preparation, bus stop selection, bus route generation, school bell time, and bus scheduling. Data preparation involves organizing routing data, encompassing student residence locations, school geographical positions, and the types of fleet used. Bus stop selection entails determining reachable stops for students, either for pick-up or drop-off, requiring students to walk to urban bus stops or having stops at their residences in rural areas. In urban areas, students walk to bus stops from their homes, whereas in rural areas, due to a smaller student population, they are often picked up directly from their homes.

The bus scheduling problem focuses on assigning a series of trips to the same bus, considering school time windows. Lastly, school bell time introduces varying start and end time constraints, sometimes leading to multiple trips for a bus schedule, reducing overall travel distance.

Certain studies have addressed the composite nature of the School Bus Routing Problem (SBRP). A substantial and expanding body of literature has delved into the bus stop selection. This process involves choosing a set of bus stops and then assigning students to these designated stops. In rural areas, students are typically picked up from their homes, while in urban regions, specific bus stops serve as pick-up points. Heuristic methods for bus stop selection fall into two categories: location-allocation-routing (LAR) strategy and allocation-routing-location (ARL) strategy, differing in the sequence of solving SBRP. The LAR strategy entails selecting bus stops first and then assigning students accordingly. Subsequently, the Vehicle Routing Problem (VRP) is formulated to generate routes incorporating the chosen bus stops.

The LAR strategy mandates that students be assigned to bus stops before route generation, making both bus stop selection and student allocation independent of route creation. This approach has drawbacks, such as neglecting bus capacity constraints during the location-allocation step, potentially leading to more bus routes than needed in the SBRP solution. This becomes critical when a large number of students can be theoretically assigned to multiple bus stops. Additionally, because routing is performed independently of location-allocation, achieving a balanced distribution of students per bus becomes more challenging. Further insights into this issue are provided in Bowerman et al. (1995), Bodin and Berman (1979), Dulac et al. (1980), and Desrosiers et al. (1981).

Dulac et al. (1980) provide an example of the LAR strategy. In their approach, students are initially situated in street segments (likely their residences) and then assigned to street intersections with the minimum distance to potential bus stops. The stop with the highest allowable number of students, based on walking distance, is chosen. Subsequently, students within a maximum walking distance are assigned to this stop, continuing until all students are assigned. In the second stage, a Vehicle Routing Problem (VRP) is solved, considering the selected bus stops. The authors' objective function includes minimizing the number of stops and the distance between selected bus stops. A notable drawback is that route generation occurs only after students are assigned to bus stops. This approach results in generating excessive routes and higher routing costs.

In the ARL strategy, students are initially grouped into clusters that adhere to bus capacity constraints. Following this, bus stop selection and route generation phases are executed for each cluster, aiming to visit the specified stops. Lastly, students within each cluster are assigned to bus stops to meet all constraints, including the maximum distance from the bus stops and the maximum number of students at each stop. Heuristic algorithms based on the ARL strategy are proposed by Chapleau et al. (1985) and Bowerman et al. (1995).

Bowerman et al. (1995) exemplify the ARL strategy in an urban SBRP with a multi-objective formulation. This problem aims to minimize various objectives, including the number of routes, total route length, variation in students assigned per route, variation in route lengths, and total walking distance from students' homes to bus stops. A key advantage of the ARL strategy lies in its effective load balance when allocating students to each cluster. Furthermore, it enables minimizing the number of bus routes, as both route-related objectives (route number minimization and load balancing) are independent of bus stop locations and the routes generated to serve them. The only drawback is in efficiently balancing route lengths due to potential student dispersion in clusters.

Corresponding to Park's survey, two alternative approaches to creating bus routes exist: the first-route-then-cluster and first-cluster-then-route strategies, both heuristic in nature. Bodin and Berman (1979) explore the "first-route–then-cluster" approach. Initially, a long route visiting all nodes is generated by solving a traveling salesman problem. This long route is then divided into smaller routes while adhering to predefined constraints, such as bus capacity, maximum travel time for students, and the maximum number of students allocated to a bus stop. For more details, refer to Newton and Thomas (1974) and Bodin and Berman (1979). In the "first-cluster–then-route" strategy, student clusters are first defined, followed by determining the number of stops for each cluster, and finally, generating a route for each cluster while considering predefined constraints.

The first-cluster–then-route strategy has been explored in several studies (e.g., Dulac et al., 1980; Chapleau et al., 1985; Bowerman et al., 1995). In both the cluster-first and route-first approaches, an initial solution is constructed, followed by an improvement phase to enhance the solution. Studies by Newton and Thomas (1969), Dulac et al. (1980), Chapleau et al. (1985), and Desrosiers et al. (1986) investigate the 2-opt method, while Bennett and Gazis (1972) and Bodin and Berman (1979) employ a 3-opt approach.

Researchers commonly employ a two-step strategy, ARL and LAR, to address SBRP by solving its sub-problems separately. However, both methods present limitations in achieving strong global optimization solutions. An underexplored area of research involves considering bus stop selection and route generation simultaneously, aiming to combine the advantages of both strategies for a more robust solution to SBRP. This integrated approach has been investigated by Schittekat et al. (2006), Schittekat et al. (2013), Riera-Ledesma et al. (2013), and Kinable et al. (2014). Schittekat et al. (2006) tackle three decisions simultaneously: (1) determining potential stops, (2) allocating students to bus stops, and (3) generating routes to minimize total travel distance. They propose an exact algorithm using Mixed Integer Programming (MIP) for small benchmark instances. Riera-Ledesma et al. (2013) devise an exact algorithm for a heterogeneous fleet problem, minimizing total route length while considering walking distance and bus capacity constraints. Schittekat et al. (2013) introduce a compact metaheuristic, highlighting the benefits of an integrated approach to bus stop selection and route generation. Comparative results demonstrate the integrated procedure's superiority, reducing the cost function by up to 25% compared to a sequential approach.

In works by Calvete et al. (2020) and Worwa et al. (2020), the sub-problems of bus stop selection, student allocation, and route generation are jointly considered, with the researchers employing efficient metaheuristics to solve the SBRP. Similarly, Sciortino et al. (2021) propose new capacity and time constraints for the open vehicle routing problem, integrating allocation-location and routing aspects. The study includes considerations like bus capacity, student eligibility, maximum student travel time, maximum walking distance, multi-stop scenarios, and bus dwell times. The objective is to minimize route journey time, student walking distance, and the number of deployed buses. The authors validate the effectiveness of their proposed heuristic algorithm on twenty real-sized instances from Malta, the UK, and Australia.

Melis and Sörensen (2022) introduce the "on-demand bus routing problem (ODBRP)" as an innovative optimization concept for flexible urban transportation. This problem encompasses buses with predetermined capacity, bus stations with time intervals, and passenger requests specifying groups with time constraints. The key decisions involve bus stop selection and routing, combining elements from three distinct problems: dial-a-ride problem (DARP) (Cordeau and Laporte, 2007), school bus routing problem (SBRP) (Schittekat et al., 2013; Kim et al., 2012), and pick-up and delivery problem with time windows (PDPTW) (Ropke and Pisinger, 2006). The objective is to minimize user ride time (URT) , and an LNS heuristic is developed, demonstrating efficacy compared to traditional public bus instances. The study suggests that introducing an on-demand bus system can enhance public transportation effectiveness, particularly for problems with a larger fleet size and lower bus capacity.

Concerning the objective function, most SBRP studies focus on minimizing bus routes and required buses (Li and Fu, 2002; Pacheco and Martí, 2006). Some also consider load balancing and maximum route length as objectives (Angel et al., 1972; Newton and Thomas, 1969; Verderber, 1974; Gavish and Shlifer, 1979; Bodin and Berman, 1979; Dulac et al., 1980; Desrosiers et al., 1981; Swersey and Ballard, 1984; Park and Kim, 2010; Eguizábal et al., 2018), student riding times (Bennet et al., 1972; Thangiah et al., 1992; Li and Fu, 2002), and student walking distance to a bus stop (Bowerman et al., 1995). De Souza Lima et al. (2017) propose a heuristic for multi-objective capacitated rural SBRP, considering total weighted student travel time, route balance among drivers, and routing costs.

Similarly, Mokhtari and Ghezavati (2018) use a hybrid ant colony and heuristic method for a multi-objective mixed-load school bus routing model, aiming to minimize buses and average student riding time.

Authors often incorporate SBRP features (e.g., number of buses, bus driver distance, student riding distance, walking distance, load balancing, maximum route length) into the objective function. Some SBRP applications include time constraints, such as minimum and maximum school arrival times, limiting each bus's student pickups and travel to school within specified periods (e.g., Swersey and Ballard, 1984; Braca et al., 1997). Newton and Thomas (1974) explore a multi-school model with varying starting times, suggesting time window division for each school.

Another survey examines various SBRP variants, focusing on multiple schools in rural or urban areas. Spada et al. (2005) propose a multiple SBRP sequencing schools by opening time, using a greedy method for route construction. Bögl et al. (2015) introduce a mathematical model and heuristic for the school bus routing and scheduling problem with transfers, allowing pupils to change buses from home to school. Studies on multi-school SBRPs are often categorized into single-load and mixed load-based approaches.

In the former case, students heading to different schools cannot travel on the same bus simultaneously, whereas the latter permits this (see, e.g., Ellegood et al., 2015). For further literature on the mixed-load routing problem, please see Section 1.3.

Han et al. (2022) address a variant of the SBRP, named MDSBRP, involving multiple schools and buses. They employ a simulated annealing metaheuristic considering school time windows and bus capacity constraints to minimize the number of buses and total operating mileage. The algorithm, tested on a benchmark case, exhibits substantial reductions in the number of required buses and operating costs, effectively handling the challenges posed by the MDSBRP feature.

Sciortino et al. (2023) tackle the SBRP with a heterogeneous fleet and a single-load strategy, introducing realism through features like student eligibility, maximum walking distances, riding time limits, multiple bus stops, and diverse bus types with varied dwell times. Their objective function focuses on total route journey time, encompassing bus travel and dwell time. They develop an algorithm to address location, allocation, and routing in SBRP, tested on real-world instances from Malta, the United Kingdom, and Australia with over 1800 potential bus stops and 750 students. The algorithm efficiently provides high-quality solutions within a reasonable computing time, particularly excelling in suggesting optimal subsets of bus stops.

Recently, various solution approaches have gained attention for addressing components of the SBRP, including Tabu search (Pacheco et al., 2013) and approximation algorithms (Bock et al., 2011 & 2013). Yigit et al. (2018) apply Ant Colony Optimization (ACO) and Genetic Algorithm (GA) to the dynamic school bus routing problem (DSBRP). Gawande and Lokhande (2018) emphasize the advantages of employing GA and Artificial Intelligence (AI).

Li and Fu (2002) recognize the absence of a singular approach to studying SBRP, introducing multiple solution methods aligned with the problem's context.

Camila Pérez et al. (2022) introduce a partial evaluation approach for the SBRP, focusing on specific aspects of the solution to reduce execution time. The model retains and examines additional information from the previous solution, like modified routes or altered objective function values, in each iteration. It identifies the cost improvement of the new solution compared to the previous one and determines which algorithmic element contributes to this enhancement. The proposed model is tested on 112 samples, ranging from simple instances with 12 bus stops and 50 students to complex scenarios with 800 students across 50 bus stops.

The results indicate that for small samples, the computational time for partial evaluation (PE) is notably worse than that for total evaluation (TE). However, in the majority of cases (80%), especially those with large and complex instances, PE outperforms TE in terms of speed. Regarding costs, it is observed that TE costs are higher than PE costs in 20% of cases.

Metaheuristic algorithms have been widely developed and proposed as effective solution methodologies in various studies, with their efficacy evaluated through multiple comparative examples (Hou et al., 2022; Xiong et al., 2022). In a recent study by Caldas et al. (2022), an iterated local metaheuristic is suggested to enhance the bus routing system for 13,664 students in rural areas of Rio de Janeiro state, Brazil. The problem involves a diverse fleet of buses, aiming to minimize the overall route cost while considering constraints like bus capacity and maximum travel distance. The proposed solution demonstrates a 40.5% decrease in the average cost of routes and a 46.0% reduction in the average distance per student traveled by bus compared to existing model. Developing effective metaheuristics faces a crucial challenge in selecting moves for fruitful exploration. Hou et al. (2022) investigate focusing on the impact of metaheuristic components on solution quality. They propose a unique approach for selecting low-level heuristics, utilizing a Q-learning technique during exploration to choose the most effective heuristic from a set. The selected heuristic acts proactively to enhance the solution. The algorithm seeks an optimal course of action by maximizing collective scores for superior outcomes. Findings indicate that the Q-learning-based selection approach outperforms random and roulette wheel selection strategies in terms of competitiveness.

Bilevel mixed-integer programming is widely employed to model hierarchical decision-making processes. Parvasi et al. (2017, 2019) introduce an optimization model enabling demand resourcing, incorporating forecasting of student responses using the LAR strategy.

The higher-level decision-maker selects bus stops and routes, while the lower level decides on outsourcing services. Notably, the model creates a fair and realistic transportation system, accommodating students' preferences. Each student has a priority order for selecting bus stops and alternative distribution services.

Calvete et al. (2023) employ a bilevel approach with three subproblems: 1) selecting a subset of bus stops, 2) allocating students to chosen bus stops based on preferences, and 3) creating optimal bus routes while adhering to capacity constraints. The study aims to convert the bilevel optimization model into a single-level mixed-integer linear programming form. Students, following a priority order, freely decide which bus stop to reach when assigned. The approach involves a two-level decision process, with the higher level minimizing total travel cost by selecting a bus stop, and the lower level allowing students to choose the most convenient bus stop. A bilevel school bus metaheuristic is proposed and compared with existing benchmarks. The results show that the free selection mechanism for students leads to conflicts with bus capacity constraints and feasibility issues, especially when the bus has limited capacity.

Table 1-1 summarizes key features addressed in the literature concerning the school bus routing problem.

Table (1-1) Features studied in the literature

Reference	Urban/ Rural	Mixed load	Fleet mix	Cost	School	Constraint	Oscillation strategy	Sub problems considered	Solution methodology
Angel et al. (1972)	Urban		HO	N RC	Multiple	C,MRT		BRG	Traveling salesman algorithm
Newton & Thomas (1969)	Urban		HO	Not specified	Single	C, MRT		BRG	Constructive heuristic combined with improvement (2-opt)
Newton & Thomas (1974)	Urban		HO	N, RC	Multiple	C, MRT		BRG RS	Constructive heuristic combined by improvement procedure
Verderber et al. (1974)	Urban		HO	N, RC	Multiple	C, MRT		BRG	Minimum distance linking algorithm (Dial-Moore algorithm) combined with special techniques
Bodin & Berman (1979)	Urban	✓	HO	N	Multiple	C, MRT TW		BSS BRG RS	Shortest chain algorithm combined with 3-opt procedure
Gavish & Shlifer (1979)	Urban		HO	N, RC	Single	C, MRT		BRG	Branch and bound procedure
Dulac et al. (1980)	Urban		HO	N, RC	Single	C, MRT, MWT		BSS BRG	Constructive heuristic combined with improvement (2-opt)
Swersey &Ballard (1984)	Urban		HO	N	Multiple	TW		RS	NLMIP, Two discretized MIP
Boweman et al. (1995)	Urban		HO	N SWD LB	Single	C MWT		BSS BRG	Districting algorithm combined with set covering and traveling sale man problem

Desrosiers et al. (1981-1986)	Both		HO	FC RC	Multiple	C, MRT, MWT		BSS BRG RS	Constructive heuristic combined with improvement (2-opt)
Chapleau et al. (1985)	Urban		HO	N, SWD	Single	C, MRT, MWT		BSS BRG	Districting algorithm combined with 2-opt operator
Braca et al. (1997)	Urban	✓	HO	N	Multiple	C, MRT, TW, EPT, MSN		BRG BSS	Location based heuristic
Spada et al. (2005)	Rural	✓	HT	TL	Multiple	C, TW		BRG RS	Simulated annealing technique
Schittekat et al. (2006)	Urban		HO	RC	Single	C		BRG BSS	integer programming formulation (VRP-like model)
Park et al. (2012)	Rural	✓	HO	N	Multiple	MRT TW C		BRG RS	Constructive heuristic combined with Post improvement
Riera-Ledesma & Salazar-Gonzlez (2013)	Urban		HO	RC	Single	C MWT		BRG BSS	Branch –cut approach
Schittekat et al. (2013)	Rural		HO	RC	Single	C MWT		BRG BSC	Greedy randomized adaptive search procedure combined with a variable neighborhood decent improvement method
Pacheco et al. (2013)	Rural		HO	TSD, MRL	Single	C		BRG	no dominated sorting genetic algorithm (NSGA) –II with tabu search
Kinable et al. (2014)	Urban		HO	RC	Single	MWT C		BSS BRG	column generation approach
Bögl et al. (2015)	Urban	✓	HO	RC	Multiple	TW C MWT		BRG RS	Constructive heuristic combined with local search operators
Hernan Caceres et al. (2015)	Sub Urban	✓	HO	RC N	Multiple	TW C MRT		BRG	Column generation approach
Ellegood et al. (2015)	Semi-rural	✓	HO	RC	Multiple	C TW		BRG RS	general strategic analysis using continuous approximation models
Yao et al. (2016)	Urban	✓	HO	RC	Multiple	MRT C		BRG	Aggregation-based clustering algorithm and improved ACO
De Souza Lima et al. (2017)	Rural	✓	HO	RC TSD RB	Multiple	C		BRG	Different heuristics based on an Iterated Local Search (ILS) framework
Mokhtari et al. (2018)	Rural	✓	HT	N TSD	Multiple	C MRT TW		BRG BSS	A hybrid multi-objective ant colony,
Miranda et al. (2018)	Rural	✓	HT	FC RC	Multiple	C, TW, MRT, MWT		BRG BSS	Different heuristics based on an (ILS) framework
Yigit et al. (2018)	Urban		HO	RC	Single	C		BRG	(ACO) and (GA)
Parvasi et al. (2018)	Urban		HO	IPS, RC, SC		C, MWT		BRG BSS	two hybrid approaches of GA-EX-TS and SA-EX-TS
Our study	Urban		HO	RC	Single	C, MWT	✓	BSS BRG	Constructive heuristic combined with adaptive large neighborhood search

- Fleet mix
  - . Homogeneous fleet (HO)
  - . Heterogeneous fleet (HT)
- Constraint
  - . Vehicle capacity (C)
  - . Maximum riding time (MRT)
  - . School time window (TW)
  - . Maximum walking time or distance to bus stop (MWT)
  - . Earliest pick-up time (EPT)
  - . Minimum student number to create a route (MSN)
  - . Maximum riding distance of bus (MRD)
  - . Depot departure time (DDT)
  - . Maximum number of students in each stop (MNS)
  - . Maximum route length (MRL)
- Objective
  - Fleet cost (FC)
  - Routing cost (RC)
  - Total student riding distance or time (TSD)
  - Student walking distance (SWD)
  - Load balancing (LB)
  - Route balancing (RB)
  - Maximum route length (MRL)
  - Child's time loss (TL)
  - Number of bus (N)
  - Income provided by service (IPS)
  - students' costs (SC)
- Sub problems considered in the literature
  - Bus stop selection (BSS)
  - Bus route generation (BRG)
  - Route scheduling (RS)
  - School bell time adjustment (SBT)
- Solution method
  - Ex =Eexcat method
  - ALNS =Adaptive large neighborhood sarch
  - TS =Tabu search
  - SA=Simulated annealing
  - ILS = Iterated Local Search
  - ACO=Ant colony optimization
  - AI=Artificial intelligence
  - NLMIP= Nonlinear mixed integer programming

1-3- Literature review concerning the mixed-load approach

An interesting application area receiving significant attention involves transporting students using a mixed-load framework, known as a mixed-loading plan. This approach entails moving students from different schools on the same bus simultaneously.

Bodin and Berman (1979) introduce the mixed-load problem, emphasizing its common use in rural areas to improve school bus service flexibility and reduce operational costs. They note that the mixed-load approach can reduce transportation costs associated with the number of buses, total travel time, and distance. However, they stress the importance of carefully considering the distance between schools and imposing maximum route length restrictions to effectively reduce students' maximum riding time on the bus.

Chen et al. (1988) argue that adopting a single load assumption results in an excessive number of buses needed for student transportation, particularly in low-density areas.

The first computational algorithm for the mixed-load problem is proposed by Braca et al. (1997). In this paper, they develop an insertion heuristic where each bus stop and its respective school are inserted in the cheapest position while satisfying time window and bus capacity constraints. The authors assert that the mixed-load problem enhances flexibility and yields significant cost savings. Braca et al. (1997) note that mixed loading has been allowed for most parts of New York City.

In a related study, Spada et al. (2005) address the problem with multiple schools and propose a heuristic procedure for its solution. The suggested approach enhances the service level offered by the bus operator while accommodating mixed-load cases. Schools are sorted based on their starting times, and routes are created using a greedy method. Subsequently, local search frameworks (simulated annealing and Tabu search) are employed to enhance the initial solution.

Park and Kim (2012) enhance the model proposed by Braca et al. (1997) by incorporating post-improvement procedures. They also conduct a quantitative study to assess the specific effects of utilizing the mixed-loading method. The problem encompasses various features, including a homogeneous fleet, different starting times, time windows, and capacity constraints. The outcomes demonstrate savings in the number of buses required. They also apply the proposed algorithm to the real-world operation of school buses, resulting in reduced bus numbers compared to current practices.

Bogl et al. (2015) examine bus stop selection, pupil assignment, bus routing, and bus scheduling. They compare results using two different modeling approaches, namely DARP (Dial-A-Ride Problem) and OVRP (Open Vehicle Routing Problem). Campbell et al. (2015) utilize a strategic continuous approximation to explore the value of mixed loading for school bus routing problems and develop three-phase heuristics to assess mixed bus trips.

The findings indicate that mixed trips are more advantageous when (1) students are sparsely distributed, (2) there are numerous bus stops, and (3) a significant percentage of stops can be shared. The results underscore that mixed routing is particularly beneficial with an appropriate distribution of students between schools and a large percentage of shared stops between schools.

Kang et al. (2015) investigate the assumptions of mixed-loading, homogeneous vehicles, and schools with different starting times. The objectives are to minimize the number of buses used (N), the sum of the travel distances of the buses (TBD) and the sum of the travel distances of the students (TSD). In case an infeasible solution arises after either a mutation or crossover operator, time-consuming repair operators are employed to return the solution to a feasible state.

Chen and Kong et al. (2015) tackle a bi-objective school bus routing problem, factoring in fleet fixed costs and routing costs. They consider different school starting times and heterogeneous fleet assumptions.

Maciel Silva et al. (2015) address a similar problem, incorporating mixed-load, heterogeneous fleet, and simultaneous school starts. Using the GRASP heuristic, they achieve a reduced fleet size (up to 37%) and lower traveled distance (up to 20%) while accounting for mixed-load effects.

Chen et al. (2015) introduce a unique problem characteristic by considering split demand for each stop, allowing multiple buses to visit the same stop.

Lima et al. (2016) develop five metaheuristic-based algorithms, considering mixed load and heterogeneous fleet scenarios.They also compare the results of the proposed algorithms in their paper. The findings show that the mixed-load approach leads to greater cost savings and a smaller fleet size compared to the single load approach.

Yao et al. (2016) propose a two-stage heuristic algorithm for the SBRP with the mixed load approach. They devise a two-stage metaheuristic, combining an aggregated clustering algorithm (AC) with improved ant colony optimization, considering mixed load characteristics.The paper explores two modes: SBRP with a virtual stop (assuming each common stop is virtualized into different stops with the same position) and interscholastic transportation (inserted in the route framework to improve the order of visiting common stops). The results indicate that mixed-load can achieve a shorter time compared to a single load. Importantly, it shows that the virtual stop scenario works well only for small cases, while interscholastic transportation provides better performance for larger instances. Regarding computing time, the SBRP with virtual stop mode takes longer running time than the case considering interscholastic transportation mode.

In a recent study, Lima et al. (2017) tackle multi-objective meta-heuristic algorithms for the multi-objective SBRP, incorporating mixed-load and a heterogeneous fleet. The proposed objectives encompass the total traveling time of students, balance of routes between drivers, and routing cost. Four multi-objective ILS metaheuristics are developed, demonstrating improved performance compared to current literature.

In the realm of mixed-load SBRP, Miranda et al. (2018) present an interesting paper introducing research that considers both mixed-load (students from different schools sharing the same bus) and multi-load problems (simultaneous pickup and delivery of students, regardless of their shift or commuting direction). They develop four versions of heuristics within an Iterated Local Search (ILS) framework, incorporating different strategies and features. The results indicate that the local search with a small-time window strategy produces better results than other versions.

Additionally, Mokhtari et al. (2018) propose a bi-objective mixed-integer linear programming formulation for mixed-load SBRP.

Several studies have explored bus scheduling approaches (Desrosiers et al., 1981; 1986a; Swersey and Ballard, 1984; Graham and Nuttle, 1986; Fügenschuh, 2009; Kim et al., 2012). Fügenschuh (2009) addresses a school bus scheduling problem allowing adjustments to school starting times and transshipment of students between trips. An integer programming model based on VRPTW (vehicle routing with time windows) is introduced and solved using a branch-and-cut method with various pre-processing procedures and valid cuts.

Kim et al. (2012) propose a bus scheduling problem where trips for each school are given separately, each containing a sequence of stops and a related school. The problem is formulated as a vehicle routing problem with time windows. Buses are assigned to predefined trips using two approaches: an exact method for small cases and a heuristic approach for larger cases. The insights from this literature provide a comprehensive understanding of various aspects of the bus routing problem, particularly in the context of mixed-load planning.

Hou et al. (2020) investigate the Heterogeneous Fleet School Bus Routing Problem, considering limited and unlimited fleet conditions. Their objective is to minimize both fixed and variable costs, and they present an ILS metaheuristic using the set partitioning algorithm. The goal is to enable this metaheuristic to obtain a globally optimal route through set-partitioning.

To better understand mixed-load effect in the current study, Table 1-2 summarizes the main features considered in rural and urban school bus routing.

Table (1-2) Features Studied in the Literature

Reference	Urban/ Rural	Mixed load	Fleet mix	Cost	Constraint	Area	Load balancing	Starting and ending location of the bus	Share flexible depot	Sub problems considered
Bodin and Berman (1979)	Rural	✓	HO	FC	C, MRT TW	Brentwood New York		School		BSS BRG RS
Hargroves et al. (1981)	Urban	✓	HT	FC RC	C MRT MNS	Albemarle, Virginia		School		BRG RS
Boweman et al. (1995)	Urban		HO	FC SWD RC LB	C MWT	Ontario Canada	✓	School		BSS BRG
Desrosiers et al. (1981-1986)	Both		HO	FC RC	C, MRT, MWT	Drummondville, Canada		Depot		BSS BRG RS
Chen et al. (1988)	Rural	✓	HO	FC, RC	C, MRT	Choctaw Alabama		Depot		BRG BS
Li and Fu (2002)	Urban		HT	FC, TSD RC	C	Hong Kong	✓	Depot		BRG
Braca et al. (1997)	Urban	✓	HO	FC	C, MRT, TW, EPT, MSN	Manhattan, New York		Depot		RG RS
Spada et al. (2005)	Rural	✓	HT	TL	C, TW	Switzerland		School		BRG RS
Fügenschuh et al. (2009)	Rural		HO	FC RC	TW	Germany		Depot		RS SBT
Park et al. (2012)	Rural	✓	HO	FC	MRT TW C	Artificial		Depot		BRG RS
Campbell et al. (2015)	Rural	✓	HO	RC FC	MRT C	Missouri USA		Depot		BRG RS

Kang et al. (2015)	Urban	✓	HT	RC FC TSD	TW MWT TW DDT C	USA		Depot		BSS BRG RS
Bögl et al. (2015)	Urban	✓	HO	RC	TW C MWT	-		Depot School		BRG RS
Hernan Caceres (2015)	Sub Urban	✓	HO	RC FC	TW C MRT	New York United States		Depot		BRG RS
Ellegood et al. (2015)	Semi-rural	✓	HO	RC	C TW	Missouri USA		Depot		BRG RS
Silva et al. (2015)	Rural	✓	HO	RC	C MWT MRD	Brazilian city		School		BSS BRG
Chen et al. (2015)	Urban		HO, HT	RC FC	TW C MRT	from literature		Depot		BRG RS
Yao et al. (2016)		✓	HO	RC	MRD C	-		School		BRG
Lima et.al (2016)	Rural	✓	HT	FC RC	C	Minas Gerais, Brazil		Depot		BRG
Lima et.al (2017)	Rural	✓	HT	TSD LB FC	C	Artificial and from the literature (Park et. (2010)		Depot		BRG
Caceresa et .al (2015)	Urban		HO	N RC	C MRT MWT	Western New York		Depot		BRG
Rodríguez-Parra et.al (2017)	Urban	✓	HO		C	Bogota		School		BRG RS
Mokhtari et.al (2018)	Rural	✓	HT	N TSD	C MRT	-		Depot		BRG
Miranda et.al (2018)	Rural	✓	HT	FC RC	C MWT MRT	Esp´ırito Santo, Brazil		Depot		BRG BSS
Our study	Urban	✓	HO	RC	C MWT TW MNS	Tehran Iran	✓	Depot	✓	BSS BRG

• Fleet mix
. Homogeneous fleet (HO)
. Heterogeneous fleet (HT)
• Constraint
. Vehicle capacity (C)
. Maximum riding time (MRT)
. School time window (TW)
. Maximum walking time or distance to bus stop (MWT)
. Earliest pick-up time (EPT)
. Minimum student number to create a route (MSN)
. Maximum riding distance of bus (MRD)
. Depot departure time (DDT)
. Maximum number of students in each stop (MNS)
. Maximum route length (MRL)

• Objective
Fleet cost (FC)
Routing cost (RC)
Total student riding distance (TSD)
Student walking distance (SWD)
Load balancing (LB)
Maximum route length (MRL)
Child's time loss (TL)
Number of bus (N)

• Sub-problems considered in the literature
Bus stop selection (BSS)
Bus route generation (BRG)
Route scheduling (RS)
School bell time adjustment (SBT)

1-4- Concerning transporting students for both morning and afternoon shifts.

Numerous studies have focused on optimizing SBRP while accounting for characteristics during both morning and afternoon shifts. In the morning, buses start from the depot (usually the driver's location) to pick up students from assigned stops and transport them to their respective schools. Conversely, in the afternoon, buses pick up students from schools and take them to their homes or stops. Existing studies have approached the morning and afternoon bus routing problem from various perspectives, as outlined in Table 1-3.

Savas (1978) suggests treating the afternoon trip as a replicated sequence of the morning version to balance student-riding time between routes. Braca (1997) contends that solving the morning routing problem is more intricate than its afternoon counterpart due to a varying range of school time windows and traffic congestion. They note that the traffic congestion constraint is higher in the morning than the afternoon, and the school time window is less restricted in the afternoon. This rationale justifies why the afternoon problem has been investigated less than the morning case. The results of Braca et al.'s (1997) study highlight that the morning requires more buses than the afternoon, with the average number of buses reduced by 5% in the afternoon.

Li and Fu (2002) assert that the afternoon problem could be transformed into a morning problem with minor modifications. Bodin and Berman (1979) suggest presenting the afternoon problem in a replication format, where the stops considered in the afternoon shift are replicated from the morning shift. In practice, the afternoon is treated as the reverse of the morning to minimize travel time. Importantly, this reverse sequence necessitates careful attention to balancing the riding time in the generated routes.

Desrosiers et al. (1981 & 1986) address route scheduling for both morning and afternoon shifts, introducing constraints such as an upper limit on the time students spend between their arrival at school and the actual starting time. Similarly, Kim et al. (2012) tackle the morning and afternoon routing problem based on a real case.
The proposed algorithm demonstrates the ability to reduce the number of buses by up to 17% in the morning and 14% in the afternoon. Moreover, the solution shortens total travel time by 12% and 16.1% in the morning and afternoon shifts, respectively.

Shafahi et al. (2017) introduce a new mathematical model aimed at maximizing compatibility among trips while minimizing the total number of required buses.
They adapt scheduling problems to the school bus routing problem, defining following objectives: a bi-objective approach maximizing trip compatibility and minimizing travel time; maximizing trip compatibility by excluding travel time from the proposed objective; minimizing travel time; and minimizing the number of buses. The authors emphasize the importance of considering route compatibility when aiming to minimize the number of buses. They include trip compatibility in the scheduling problem within the routing case. Additionally, the authors suggest that their model could yield better results with a 30-minute time interval between the earliest and latest school time windows.

Compared to the traditional scheme, the proposed model reduces the number of required buses by 13%. In a subsequent study, the authors (2017) extend their previous work, incorporating both routing and scheduling problems to enhance their model's performance. The modified objective differs from the original one by Shafahi et al. (2017), assigning a lower weight to trip compatibility and a higher weight to schools with shorter dismissal times. Employing a new solution approach, a two-step heuristic algorithm, they solve a proposed problem in the literature, resulting in a 25% savings compared to the findings reported in Shafahi et al. (2017).

Following Shafahi et al. (2017), Wang et al. (2017) advocate for defining the financial objective of the school bus routing problem to address both routing and scheduling sub-problems. They argue that simultaneous consideration of routing and scheduling enhances compatibility. The problem includes known school bell times and defines the number of students at each stop. In their novel integrated model, all buses are assumed to have the same capacity and follow a single load scheme. The afternoon trip in this model is constructed in the reverse order of the morning trip, with pickup and drop-off times drawn from Braca et al. (1997). In the afternoon, the bus picks up students from stops and sequentially transports them to their respective schools. Results show that the integrated model outperforms the one proposed by Shafahi et al. (2017), reducing the number of required buses and total travel time by 25% and 7%, respectively.

The morning and afternoon trips are consistently scheduled, with the afternoon trip being the reverse version of the other trip. Morfoulaki et al. (2017) present a framework for optimizing the school bus routing problem, considering both morning and afternoon characteristics. The study is conducted at a private school in Thessaloniki, Greece.

The model is based on two morning trips (starting at 7:00 and 8:00 am) and three afternoon trips (starting at 1, 2, and 3 pm), resulting in decreased total travel time and emission criteria. Notably, NOX and CO2 emissions are reduced by 2.7 and 0.02 tons, respectively. The results highlight the advantage of an effective mixed-load scheme when schools and students' locations are widely dispersed. However, implementing the mixed-load scheme is problematic when a student of the same cluster belongs to the same school, emphasizing the priority of the single load scheme in such situations. The experimental design, when compared to the real case, has the potential to reduce computing time by up to 20%.

Ellegood et al. (2015) utilize continuous approximation modeling to investigate how a mixed-load mechanism can reduce total travel distance during morning and afternoon shifts. They discover that when the distance between schools (destination points) is less than the critical value, the mixed-load policy leads to a greater reduction in total travel distance. The concept of school bell time involves adjusting the time window for morning arrival time at the school or the departure time window in the afternoon. This framework integrates routing and scheduling sub-problems, consequently adjusting the start and end time windows of schools on a daily basis.

Caceres et al. (2017) introduce the school bus routing problem incorporating stochastic demand and duration constraints. This study represents the first attempt in the SBRP field to combine stochastic demands with constraints aimed at calculating overcrowding probability and students' lateness to school.

The authors account for both ridership and travel time uncertainty, with the primary objective being the minimization of the number of buses and the secondary objective being the minimization of total travel distance. The novelty lies in the mathematical formulation, utilizing chance-constrained programming to address the overbooking policy.

Shafahi et al. (2018) introduce a special case of the balanced load-scheduling problem, simultaneously considering two objectives. The problem aims to assign stops to each route to minimize the number of routes and optimally shorten the route length.

The second objective focuses on minimizing deviation from the target duration, set at 75 minutes for all instances. The case study draws from two real problems: California with 54 trips and the Howard County Public School System (HCPSS) with 994 trips. The balanced version proves capable of reducing school operational costs by 16% and the standard deviation of trip duration to 47%. The study's results offer a more suitable planning approach for drivers.

Shafahi et al. (2018) recommend combining bus routing and scheduling using a two-step heuristic.

They first generate an initial solution with an "iterative minimum cost matching-based insertion heuristic." Then, they enhance the solution with simulated annealing and Tabu search under a single load assumption. Results show a 25% reduction in required buses compared to Park et al. (2012) and a 10% decrease in the maximum riding time of 2,700 seconds. The authors propose a post-improvement that could further strengthen the solution by 4.9%. The suggested mixed-load framework needs 4% fewer buses than Campbell et al.'s (2015) solution.

Oluwadare et al. (2018) propose a multi-objective function to minimize the number of routes and total travel distance for effective management of morning and afternoon trips. They make the following assumptions: Each route can have a maximum of two buses, with one at the source point and another at the destination point.

Students in a route can take more than one bus, disembarking at a special stop and getting picked up at another stop before reaching their respective school. Additionally, all buses have the same capacity.

Levin et al. (2016) suggest employing a decision support system for developing effective bus routing schemes. The solution utilizes the Clarke and Wright heuristic with the aim of minimizing bus operating time for both morning and afternoon shifts, adhering to predefined constraints.

To account for travel time and ridership uncertainty, the authors incorporate varying travel times at peak traffic congestion points and set maximum ridership per stop. The results indicate a substantial operational saving in the public transport system.

Bertsimas et al. (2020) develop an algorithm allocating students to stops, combining stops on routes, and assigning buses. Applied to two school years in Boston, Massachusetts, their algorithm reduces required buses by 7%, translating to an annual saving of $12 million. Introducing time analysis as an innovation, the authors conduct a survey to determine the community's preferred school start time. Using preference scores and the algorithm, they calculate the required buses, optimizing between preference score (i.e., respondent’s satisfaction) and transportation cost. Survey results indicate 85% are willing to change their school start time by an average of one hour. In summary, the authors propose a novel optimization model for the School Time Selection Problem (STSP).

Expanding the Social Bus Routing Problem (SBRP), Orejuela Cabrera et al. (2021) address its social dimension by introducing affinity as a factor in analyzing positive relationships between students on the bus. They develop a solution that assigns students to the bus, evaluates their affinity, and defines routes accordingly. The paper's key contribution lies in emphasizing the role of affinity in deciding which students should be assigned to a given bus.

Table (1-3) Features studied in the literature

Reference	Year	Sub-problem type	Service Environment	Split Load	Mixed loads	Fleet mix	Obj.	Con	Area
Bodin and Berman	1979	BSS, BRG, BRS	Rural		YES	HO	N	C, MRT, TW	United States
Hargroves et al.	1981	BRG, BRS	Urban		YES	HT	N, TBD	C, MRT, MSN	United States
Boweman et al.	1995	BSS, BRG	Urban			HO	N, SWD, TBD, LB	C, MWT	Canada
Desrosiers et al.	1981-1986	BSS, BRG, BRS	Both			HO	N, TBD	C, MRT, MWT	Canada
Chen et al.	1988	BRG,BSS	Rural			HO	N, TBD	C, MRT	Choctaw
Braca et al	1997	BRG, BRS	Urban		YES	HO	N	C, MRT, TW, EPT, MSN	United States
Li and Fu	2002	BRG	Urban		NO	HT	N, TSD TBD	C	Hong Kong
Spada et al.	2005	BRG, BRS	Rural		YES	HT	TL	C, TW	Switzerland
Kim et al	2012	BRS	Urban			HT	N	C, TW	United States
Park et al	2012	BRG, BRS	Rural		YES	HO	N	MRT, TW, C	Artificial
Ellegood et al.	2015	STP	Both		YES	HO	TBD	C,TW	United States
Campbell et al.	2015	BRG, BRS	Rural		YES	HO	N, TBD	MRT, C, TW	United States
Kang et al	2015	BSS, BRG, BRS	Urban		YES	HT	TBD, N ,TSD,NS	C,MRT,TW	Artificial
Bögl et al.	2015	BRG, BRS	Urban		YES	HO	TBD	TW, C, MWT	United States
Caceres et al	2018	BRG	Urban		YES	HT	N, TBD	TW, C, MRT	United States
Silva et al	2015	BSS,BRG	Rural		YES	HO	TBD	C, MWT, MRD	Brazil
Ruiz et al	2015	BRG, BRS	Urban		YES	HO	N	C	Artificial
Levin & Boyles	2016	BRG	Urban		NO		TBD	C, MRT	United States
Caceres et al	2017	BRG, BRS	Urban		NO	HO	N, TBD	C, MRT, COO, COL	United States
Bertsimas et al	2020	BRG,BRS	Urban		NO	HT	N	C,TW	United States
Yao et al.	2016	BRG,BRS	Urban		YES	HO	TBD	C	Artificial
Lima et.al	2016	BRG	Rural		YES	HT	N, TBD	C	Brazil
Siqueira et al.	2016	BRG	Both		NO	HT	TBD	C,TW	Brazil
Rodríguez-Parra et.al	2017	BRG, BRS	Urban		YES	HO	TBD	C	Colombia
Lima et.al	2017	BRG, BRS	Both		YES	HT	TBD, LB ,TC	C	Brazil
Mokhtari et.al	2018	BRG	Rural		YES	HT	TSD, LB ,TC	C MRT	Artificial
Miranda et.al	2018	BRG, BSS	Rural		NO	HT	N, TBD	C, MWT, MRT	Brazil
Shafahi et al	2017	BRG, BRS	Rural		NO	HO	N, TBD,TRC	C,TW	Artificial
Worwa	2017	BRG, BSS			NO	HO	N	MWT	Artificial
Shafahi et al	2018	BRG, BRS	Both		NO	HO	TC	C, TW, MRT	Artificial

Oluwadare et al	2018	BRG	Both	YES	HO	N, TBD	C	Nigeria
Bertsimas et al	2019	BSS, BRG, BRS, SBA	Urban	NO	HO	SWD, N, TSD	C, MRT, TW	United States
Calvete	2020	BRG, BSS	TBD	NO	HO	TBD	C,MWT	Artificial
E HOU	2020	BRG	Rural	YES	HT	N	C,TW,MRT	Artificial
E HOU	2020	BRG	Urban	NO	HT	N,TBD	C,MRT,MNB	China
Sciortino	2021	BRG, BSS	Urban	NO	HO	N,TBD,SWD	C,MRT	UK and Australia
Ansari	2021	BRG	Urban	NO	HT	TBD	C	Artificial,
Li	2021	BRG, BSS	Urban	YES	HT	TBD,SWD	C,MRT	United States
Komijan	2021	BRG, BSS	Urban	YES	HO	N,TBD	C	Iran
Orejuela	2021	AC	Urban	NO	HO	AC,TBD	C	Colombia

Number of buses used (N),Total bus travel distance or time (TBD), Total student riding distance or time (TSD),Total student walking distance (SWD), Load or ride time balance(LB),Total cost (TC), Trip compatibility (TRC) , affinity of children (AC),Number of stops (NS),Maximum riding time (MRT),Vehicle capacity (C),School time window (TW),Maximum walking time or distance(MWT),Earliest pick-up time (EPT),Minimum student number to create a route(MSN),Chance of overcrowding(COO),Chance of being late(COL),Child's time loss (TL), Maximum no of buses (MNB),mixed ride(MR)

1-5- Concerning the safety and health issues

Limited research exists on student safety in transportation. Chalkia et al. (2016) address this gap by introducing a new method for safer and more efficient transportation of students to and from school. The method involves creating a safety map that assigns risk safety features to considered arcs and nodes, considering criteria like traffic flow, speed, and road lighting. To solve this problem, three algorithms are proposed, with the genetic algorithm proving superior to the others.

Fernandes et al. (2023) suggest in their review that school-based interventions focusing on the built environment may moderately enhance students' physical activity, health, and active commuting.

Ensuring student safety and well-being during school bus transportation poses challenges and concerns for school boards, families, and transportation companies. The transportation of students with disabilities adds further complexities for transportation companies.

Ross et al. (2023) conducted a scoping review on transportation services for students with disabilities. The review involves analyzing 20 documents from various perspectives, including the viewpoints of students with disabilities, stakeholders' understanding of essential education, disability rights legislation, challenges and concerns in school transportation, and alternative transportation options. The findings suggest a need for considering alternative practices. Strategies to enhance transportation for students with special needs may include consolidating transportation with other students, providing training en route to school, utilizing inclusive technology, instructing practical skills with medical professionals, and collaborating with non-transportation stakeholders for safe travel planning.

In a cross-sectional study, Rothman et al. (2021) explore the correlation between built environment factors and Active School Transportation (AST) safety in seven

Canadian communities. They analyze factors like child population density, multi-dwelling housing density, pre-1960 housing density, school design elements (e.g., parking, proximity, crossing guards, cycling infrastructure, sidewalks, car drop-off, traffic calming devices), and road design features (e.g., intersection density, traffic signal density, local road density). Social and environmental factors, such as school population and new immigrants, are also considered.

The study suggests that crossing guards, cycling infrastructure, and traffic signs significantly contribute to enhancing AST safety, with their importance varying based on school type, travel culture, and city-specific challenges (Rothman et al., 2018). Implementing initiatives and new programs in drop-off zones can be a feasible solution to improve school routing safety. Looking at it differently, parents of disabled children may demand a robust response to perceived travel risks, expressing worries about their children's use and access to school transportation services.

These concerns stem from issues such as non-compliance with safety regulations and protocols (Falkmer et al., 2004), insufficient knowledge and skills of drivers, particularly with regard to students with disabilities, and inadequate seating arrangements on buses (Falkmer and Gregersen, 2002). Parents also express concern about the training and regulations provided to bus drivers, particularly their ability to assist disabled children with medical needs during transportation (Falkmer et al., 2004).

Falkmer et al. (2002) highlight the primary risk as the potential for severe injury or death resulting from traffic accidents. To ensure passenger safety, factors such as driver expertise, knowledge, and abilities; safe locations for student pick-up and drop-off; efficient traffic management; and prevention of driver fatigue should be carefully considered.

A Swedish organization, SAFEWAY2SCHOOL, has developed a technologically advanced program aimed at improving school travel safety. The program utilizes measures such as smart bus stops, GPS tags on buses, and onboard computers to monitor and control student safety comprehensively during bus trips (Falkmer et al., 2014).

Improvements are needed in various aspects of school travel safety, as highlighted by Dubée et al. (2017). They stress the significance of maintaining appropriate bus schedules and establishing a clear emergency protocol for unforeseen events during school bus trips. Attention should also be directed towards addressing built environment issues, encompassing pick-up and drop-off procedures, bus-to-schoolyard arrangements, and individualized education programs (IEP) for each student.

To tackle safety concerns effectively, it is crucial for school staff, boards, service providers, and bus drivers to enhance engagement with students and families, especially those who are vulnerable, and provide support to alleviate their concerns.

Chapter 2:

Iterated local search algorithm with strategic oscillation for school bus routing problem with bus stop selection

2-1- Introduction

Transporting students to and from school poses a budgetary challenge for local governments aiming to optimize spending. Efficient coordination and planning of urban transportation networks are crucial. “Students are not simple packages, as in the case of pick-up and delivery of goods, and because these services are provided through the public sector” (Bowerman & Calamai, 1995). Moreover, as these services are public sector-provided, the School Bus Routing Problem has been extensively researched. SBRP focuses on safely, economically, and conveniently transporting students to and from school (Corberán et al., 2002).

In real-world applications, the School Bus Routing Problem (SBRP) must account for additional constraints and factors related to the usability of the transportation network. Consequently, SBRP involves a more intricate formulation and generally requires a more complex solution strategy compared to the Vehicle Routing Problem (VRP). SBRP aims to determine routes that minimize/maximize specific objectives. These objectives focus on enhancing the overall public transportation system while adhering to various constraints such as bus capacity, maximum student riding time, and school time windows.

This chapter focuses on a specific variant of the school bus routing problem for a single school without time windows. The School Bus Routing Problem (SBRP) involves following decisions: creating potential bus stops within a maximum allowable distance from student locations, defining optimized bus routes with selected stops to ensure students reach school within capacity limits. SBRP breaks down into three sub-problems: (1) selecting the minimal set of bus stops for each route, (2) allocating students to stops without exceeding capacity, and (3) defining bus routes with selected stops to minimize total bus travel distance.

This chapter tackles the three sub-problems by integrating them into a single optimization procedure.

Figure 2-1 exemplifies this issue. It features students (dots), potential stops (small squares), and a school (large square). In 2-1(a), dotted lines represent possible allocations to reachable stops based on walking distance. With a bus capacity of 6, a feasible solution is shown in 2-1(b). Note that assigning students to different stops in the same route is optional, while allocating students to different stops on different routes depends on satisfying bus capacity.

As evident in Figure 2-1, student 1 can be assigned to stops A and B on separate routes. To adhere to bus capacity constraints, student 1 is compelled to walk to stop A. It highlights the need to address both interconnected sub-problems (1) and (3) simultaneously. The figure illustrates that student 1 must identify an allowable stop to visit (sub-problem 1). Allocating student 1 to stop B results in an infeasible solution, whereas allocating to stop A on the other route yields a feasible solution. Therefore, choosing an allowable bus stop (sub-problem 1) and determining the route to visit the selected stops (sub-problem 3) must be addressed concurrently. Sub-problem 2 examines the feasibility of a selected stop in a given route.

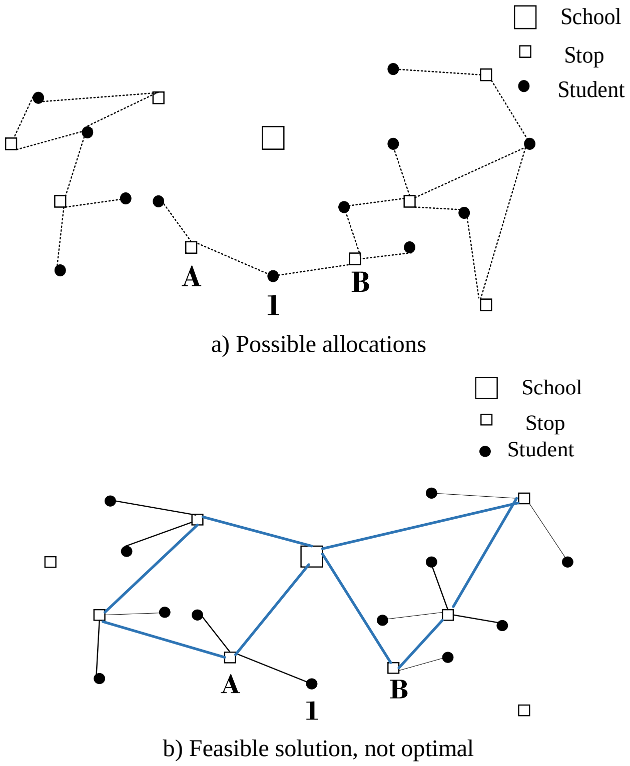


Figure (2-1) possible allocations of students to stops (a) and a feasible solution (b)

Until now, research has largely overlooked the context of addressing both bus stop selection and route generation problems concurrently. Only a handful of studies, such as those by Schittekat et al. (2006), Riera-Ledesma and Salazar-Gonzalez (2013), Schittekat et al. (2013), and Kinable et al. (2014), have explored these challenges in tandem (for more detail, see Section 1.2). Consequently, the primary focus of this chapter is to address both bus stop selection and route generation problems simultaneously.

Concerning the solution approach, previous studies have focused on heuristics exploring only feasible solutions, but efficient problem-solving involves considering both feasible and infeasible solutions across a broader search area. For example, Brandao (2006) introduces Strategic Oscillation, a strategy efficiently navigating between feasible and infeasible search areas. This involves the oscillating local search (OLS) strategy, emphasizing the exploration of infeasible portions of the search space. The goal of oscillation is to move continually between feasible and infeasible spaces, with controlled movements to manage the transition.

Applying the strategic oscillations is not a novel concept in Vehicle Routing Problems (VRP). There are several reasons for adopting this strategy:

1. Boundary identification: Investigating the extent to which it is possible to explore the infeasible space and the mechanism for returning to the feasible space.
2. Enhanced global search: It contributes to a deeper search, increasing the probability of discovering an improved overall solution.
3. Iterative refinement: Oscillating between feasible and infeasible regions provides opportunities for iterative refinement and solution improvement.
4. Transition speed analysis: Examining the speed of transition between feasible and infeasible parts of the solution space yields valuable insights.

This chapter introduces a novel application of strategic oscillation in the School Bus Routing Problem (SBRP). We present an iterated local search algorithm utilizing strategic oscillation for the dynamic transition between feasible and infeasible solutions. This method incorporates six neighborhood structures within a variable neighborhood descent approach. The heuristics, Insertion Iterated Local Search (I-ILS) and Nearest Neighborhood Iterated Local Search (N-ILS), are employed and abbreviated. To validate these metaheuristics, SBRP instances are used.

Contributions of this chapter are:

1) Solving SBRP with an iterated local search heuristic incorporating strategic oscillation.
2) Introducing two metaheuristics, N-ILS (a variant of the Nearest Neighborhood with Iterated Local Search) and I-ILS (a variant of Insertion with Iterated Local Search), structured around constructive, intensification, and diversification stages.
3) Obtaining the optimal solutions for small instances using the CPLEX solver, and tuning key components of each metaheuristic,
4) Comparing the proposed methods (I-ILS and N-ILS) with the best-known solutions by Schittekat et al. (2013).
5) Integrating both ARL and LAR strategies into a unified single-step optimization approach.

The chapter is organized as follows: Problem description and formulation in Section 2.2, solution approach and metaheuristic configuration in Section 2.3, computational experiments in Section 2.5, and conclusions with suggestions for future research in Section 2.6.

2-2- Problem definition

The studied problem, SBRP, is an extension of the familiar Vehicle Routing Problem (VRP). It involves a single school, one student type, and identical buses with fixed capacity, aiming to optimize total travel distance, akin to traditional VRP. Given its extension from VRP, SBRP is likely NP-hard. The table below provides a summary of symbols used in the model.

Table (2-1) Symbol used in mathematical model

Data	
C	Bus capacity
V	Set of potential stops with |V|=n
S	Set of students
B	Set of buses
C_{ij}	Travel cost from stop i to stop j
S_{il}	Binary parameter equal to 1 if student l can reach stop i , 0 otherwise. (When $S_{il} = 1$, the distance of student l from stop i is within the maximum walking distance).
$i = 0$	Index for school
Decision variable	
x_{ijk}	1 if bus k traverses the arc from stop i to j , 0 otherwise
y_{ik}	1 if the bus k meets stop i , 0 otherwise
z_{ilk}	1 if student l is picked up by bus k at stop i , 0 otherwise

The presented model is constructed following the formulation by Toth and Vigo (2002) for the SBRP. The SBRP formulation below is taken from Schittekat et al. (2013).

$$Min \sum_{i\in V}\sum_{j\in V} C_{ij} \sum_{k=1}^{n} x_{ijk} \tag{2-1}$$

$$\sum_{j\in V} x_{ijk} = \sum_{j\in V} x_{jik} = y_{ik} \qquad \forall i \in V\,, k = 1, \dots, n \tag{2-2}$$

$$\sum_{k=1}^{n} y_{ik} \le 1 \qquad \forall i \in V\,\backslash\{0\} \tag{2-3}$$

$$\sum_{k=1}^{n} z_{ilk} \le S_{il} \qquad \forall l \in S\,, \forall i \in V \tag{2-4}$$

$$\sum_{i\in V}\sum_{l\in S} z_{ilk} \le C \qquad k = 1, \dots, n \tag{2-5}$$

$$z_{ilk} \le y_{ik} \qquad \forall i, l, k \tag{2-6}$$

$$\sum_{i\in V}\sum_{k=1}^{n} z_{ilk} = 1 \qquad \forall l \in S \tag{2-7}$$

$$\sum_{i,j\in Q} x_{ijk} \le |Q| - 1 \qquad \forall Q \subseteq V\backslash\{v_0\}, \forall k \tag{2-8}$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in V\,, k = 1, \dots, n \tag{2-9}$$

$$x_{ijk} \in \{0,1\} \qquad \forall i\,, j \in V\,, i \ne j\,, k = 1, \dots, n \tag{2-10}$$

$$z_{ilk} \in \{0,1\} \qquad \forall i\,, j \in V\,, i \ne j\,, l \in S, k = 1, \dots, n \tag{2-11}$$

The objective function (2-1) minimizes the total travel distance handled by all buses. Due to constraints (2-2), for each stop i the number of arcs entering is exactly the same as the number of arcs going out from it. Constraints (2-3) guarantee that each stop is visited only once, except for stop 0 which is associated with the school. Constraints (2-4) enforce that each student is to be picked up at the stop where he/she walks to. Constraints (2-5) guarantee that the capacity of buses is not to be exceeded. Inequalities (2-6) impose that picking up a student in a non-visited stop by bus k is not possible. Constraints (2-7) state that each student is picked up only once. Constraints (2-8) force the connectivity of the route performed by bus k. This constraint shows sub-tour elimination constraints. Finally, constraints (2-9), (2-10), and (2-11) define the domain of the decision variables which are all binary. The MIP formulation presented in this section was solved using the CPLEX solver in GAMS software and tested on small instances (see Appendix 1). As the instance size increases, the computing time escalates. The exact method achieved optimality in less than one hour for 43 instances, involving up to 10 stops and 200 students. However, for larger instances, the CPLEX solver becomes impractical. To address this limitation, a metaheuristic approach is developed and discussed in the following section.

The sub-tour elimination constraints (2-8) in the current implementation have been replaced with the Miller–Tucker–Zemlin constraints (Miller et al., 1960). Here u_i represents the order in which stop i is visited in the sequence and serves as the variable to prevent sub-tours.

$$u_i - u_j + n x_{ijk} \leq n - 1 \qquad \forall i, j = 2, \dots\dots n, \quad i \neq j \qquad (2\text{-}12)$$
$$u_1 = 1 \qquad (2\text{-}13)$$

2-3- Metaheuristic configurations

Results indicate that the exact method solves only the easiest 43 instances in less than an hour, proving impractical for larger cases. Therefore, a fast and robust solution is essential. Emphasizing solution quality, specifically determining the minimum travel distance for buses, is crucial. To address this, two metaheuristics, N-ILS and I-ILS, are introduced. Both are iterated local search (ILS) metaheuristics (Lourenco et al., 2010), comprising constructive, intensification, and diversification stages.

Notably, the metaheuristics differ in the constructive stage for generating initial SBRP solutions (see Algorithm 2-1). The intensification and diversification stages are identical in both metaheuristics. In N-ILS, the constructive stage employs a nearest neighborhood heuristic, while I-ILS utilizes an insertion heuristic. The resulting solutions from these heuristics become inputs for the intensification stage, driven by an oscillating local search (OLS) heuristic. OLS comprises two levels: (1) improvement through a variable neighborhood descent (VND) heuristic, a variant of variable neighborhood search (VNS) (Hansen and Mladenovic, 1999), and (2) re-optimization using remove and redistribution operators. These stages are executed sequentially (refer to Section 2.3.5).

The VND heuristic in the improvement phase incorporates six local search operators: three intra-route and three inter-route operators. Intra-route operators focus on enhancing the solution by modifying one route at a time, while inter-route operators alter multiple routes concurrently.

The VND heuristic concludes when local optima are attained. To enhance the search process further, the outcomes from the initial stage feed into the second step (the re-optimization stage), involving remove and redistribution heuristics.

To search unexplored regions of the solution space and avoid local optima, two diversification strategies are employed (see Section 2.4).

The first strategy perturbs the current solution by partially destroying and rebuilding a limited number of routes. The second strategy involves the double-swap move (1, 1) (Subramanian & Drummond, 2010), executing two swap movements randomly in sequence. In our metaheuristic, we relax the capacity constraints, allowing oscillation between feasible and infeasible solutions. The following sections detail the main components of the metaheuristic, discussing elements influencing algorithm performance.

The analysis of metaheuristic configuration emphasizes the application of each element separately. In Algorithm 2-1, the ILS metaheuristic employs the construction phase once and iterates the local search multiple times, starting from a perturbed solution. The ILS metaheuristic terminates upon reaching a maximum number of iterations (φ).

Algorithm (2-1) Iterated local search (ILS)

$S_o \leftarrow Generate\ Initial\ Solution;$ $\quad$ // {$Constructive\ phase$}

$S_{best} \leftarrow OLS\ (S_0)$; $\quad$ // $Improve\ solution\ using$ "OLS" $heuristic$

$While\ (\varphi\ is\ not\ reached)\ do$

$\quad S_{perturb} \leftarrow Perturb\ (S_{best})$; $\quad$ // $Perturb\ the\ best\ solution\ found\ so\ far\ S_{best}$

$\quad S_{ols} \leftarrow OLS\ (S_{perturb})$; $\quad$ // $Improve\ solution\ using$ "OLS" $heuristic$

$\quad\quad If\ Cost(s_{ols}) < Cost(s_{best})\ \ then$

$\quad\quad\quad S_{best} \leftarrow\ S_{ols}$;

$\quad\quad End\ if$

End while

Report best solution S_{best} *found*

2-3-1- Nearest neighborhood with greedy randomized selection mechanism (NNg)

The first constructive heuristic developed in this chapter is a nearest neighborhood heuristic with a greedy randomized selection process (Talarico et al., 2015). Initially, a student allocation problem is solved for each stop. Following this, a variant of the nearest neighborhood constructive heuristic is applied, where a greedy randomized selection mechanism is used instead of a traditional greedy process. The next stop in the current route is randomly chosen from a restricted candidate list (RCL) containing the α first closest non-visited stops. After selecting non-visited stops, the capacity constraints are checked. If a feasible solution is found, a new stop is added to the current route; otherwise, the current route is closed, and a new one is constructed from the school to new non-visited stops.

2-3-2- Insertion heuristic with greedy randomized selection mechanism (Ig)

The second constructive heuristic follows an insertion approach, akin to Campbell and Savelsbergh (2004). Similar to the modified nearest neighborhood heuristic, a student allocation problem is initially solved for each stop. Our version of the insertion heuristic is then adjusted in two stages. First, a greedy randomized selection mechanism replaces the traditional greedy heuristic, determining a restricted candidate list of α least-cost positions for inserting each non-visited stop into current routes. Second, a feasibility check is performed for each non-visited stop considered for insertion regarding the capacity constraints. If no feasible insertion position is found for the remaining stops, insertion occurs at the cheapest possible position without considering a feasibility check. This implies that only the first stage is considered, potentially resulting in either a feasible or less-violated capacity solution, without ensuring feasibility in the order of insertions.

2-3-3- Oscillating local search (OLS) heuristic

Strategic oscillation aims to temporarily explore the infeasible part of the solution space (refer to Toth & Vigo, 2003). The strategy employs a method to identify promising feasible search spaces. Notably, the excursion into the infeasible area is governed by a dynamically adjusted penalty function, which is multiplied by the cost of the infeasible solution. The cost function, as detailed in this chapter, is introduced by Brandao (2006) and comprises two terms:

$$cost(s,\lambda) = cost(s) + \lambda d(s) \quad (2\text{-}14)$$

The first term represents the total travel distance covered by all buses (objective function), while the second term, denoted by $d(s)$, is the sum of excess loads for each bus (violations of capacity). This is multiplied by a penalty value, λ, dynamically updated during exploration. If all bus routes satisfy capacity, the second term is zero. The penalty, λ, accentuates the capacity violation's impact on the total solution cost. The pseudo-code for the oscillating local search (OLS) strategy is in Algorithms 2-2. OLS takes initial parameters: λ_0, an initial solution (S_0), and a penalty increasing factor (β). S_0 is generated in the constructive or diversification phases.

In Algorithm 2-2's first level, the OLS heuristic initiates by employing a variable neighborhood descent heuristic with six well-known operators: three intra-route (Relocate, Swap, Two-Opt) and three inter-route (Relocate, Swap, Two-Opt). These operators are applied sequentially (from small to large size) until a local optimum is reached (line 11 in the pseudo-code). Post the first level exploration, the feasibility of the locally optimal solution, S_{act}, is evaluated (line 12). If S_{act} is infeasible, λ increases to enhance exploration in the feasible region (lines 22 and 23). Conversely, if a feasible solution is found, the algorithm proceeds as follows. The solution from the first stage (VND heuristic) becomes input for the second stage (re-optimization level) for further improvement, involving two operators: remove and redistribution (described in Section 2.3.5).

After exploring the second stage, a verification process checks if the solution of the second stage, denoted as $S_{act}^{*\prime}$, surpasses the best feasible solution identified thus far (refer to lines 16 and 17 of the pseudo-code). If it does, the new solution, $S_{act}^{*\prime}$, is accepted, and the value of λ resets to λ_0 (see line 19 of the pseudo-code). This adjustment occurs because the exploration may have focused on the same feasible local optimal solution (cycling around it) or the algorithm might be searching in an unpromising region of the solution space containing a weak feasible solution.

In summary, the algorithm extensively explores the feasible solution space for a considerable number of iterations, with the value of λ set to λ_0 to restrict deeper exploration into infeasible areas. Conversely, if the algorithm prioritizes exploring infeasible segments for a larger iteration count, λ increases to guide the exploration back to the feasible region.

Therefore, λ aids in assessing the exploration effectiveness in the solution space. In OLS, λ increases when no feasible local optimal solution is found and resets to λ_0 upon achieving the best feasible solution. The values of λ and β significantly impact OLS performance, with λ representing the heuristic's ability to explore infeasible solution space portions, associated with the maximum violation of the bus capacity constraints in the OSL heuristic.

As depicted in Algorithm 2-2, a higher λ value guides the algorithm back into the feasible solution space, resulting in a reduced ability to explore the infeasible segment of the solution space. In line 22, λ_u presents the higher value of λ, serving as the upper bound for the penalty value. Initially set at λ_0, and then λ is systematically multiplied by β. Here, β represents the transition speed from the infeasible to the feasible space, impacting the ability to discover feasible regions with improved solutions.

A larger β value facilitates an increase in the penalty λ, compelling OLS to swiftly reach a feasible solution.

Algorithm (2-2) Oscillating Local Search (OLS) Heuristic

Input: initial solution S_0, lower end point for penalty value λ_0 (initial penalty value), upper end point for penalty value λ_u, penalty increase factor β ,
$If\ S_0\ is\ feasible\ then$
$S_{best-feasible} = S_0$
$cost_{best-feasible} = cost(S_0)$
$Else$
$cost_{best-feasible} = \infty$
$End\ if$
$\lambda = \lambda_0$
$S_{act} = S_0$
$//Improvement\ in\ first\ level$
$S_{act} = Applying\ VND\ (S_{act}, \lambda)$
$if\ S_{act}\ is\ feasible\ then$
$//Re-optimize\ in\ second\ level$
$S_{act}^{*} = Remove\ operator(S_{act})$
$S_{act}^{*\prime} = Redistribution\ operator\ (S_{act}^{*})$
$if\ cost(S_{act}^{*\prime}) \leq\ cost_{best-feasible}\ then$
$cost_{best-feasible} = cost(S_{act}^{*\prime})$
$S_{best-feasible} = S_{act}^{*\prime}$
$\lambda = \lambda_0$
$End\ if$
$Else$
$if\ (\lambda < \lambda_u)\ then$
$\lambda = \lambda \times \beta$
$Else$
$S_{act} = Restore\ operator(S_{act}, \lambda)$
$End\ if$
$End\ if$
$Output: S_{best-feasible}$

During the algorithm, there's a possibility that the λ value increases and reaches the predefined value λ_u . This can lead to a situation where the current solution deviates significantly from feasibility. To address this issue, a restore operator is implemented, aiming to return the solution to a feasible state in two stages. In the first stage, the procedure attempts to allocate students to allowable stops in other routes, iterating through all eligible stops and routes. If, by the end of the first stage, no violated route remains, a feasible solution is achieved. Otherwise, the solution enters the second step, known as the split procedure (for more detail see Chapter 2.3.6).

2-3-4- Local search operators

- **Remove-insert within and between routes**

This operator removes one stop from the current location and inserts it in another location (either in the same route or a different route).

- **Swap within and between routes**

This operator swaps the positions of two stops. The intra-route version concentrates on swapping movements within the same route, whereas the inter-route operator exchanges stops between different routes.

- **Two-Opt within and between routes**

The two-opt intra-route operator creates new routes by removing two edges from one route and then reconnecting the route, thus reversing the order of the route. Before the above moves are performed, cost check is executed. The move is executed if it results in cost reduction; otherwise, it is discarded.

The overall complexity of intra and inter-route operators relies on the feasibility check, student allocation, and the necessary operations for move execution. Given the problem's nature, which permits exploration into infeasible solution spaces, the complexity is considered only for the move operations as outlined below:

- **Remove-Insert within route:**

Intra-route for loop for each route (r) nested with a for loop for each stop (n) + for loop to find another location to place that stop in the same route. This results in a complexity of O(|r|·|n|·|n|), simplifying to o(n^2).

- **Remove-Insert between routes:**

Inter-route for loop for each route nested with a for loop for each stop + for loop for each route and for each stop to find another location to place that stop in a different route. The overall complexity is O(|r|·|n|·|r|·|n|), also simplifying to o(n^2).

- **Swap within route:**

Intra-route loop for each route, nested with a for loop for each stop. Additionally, there is a loop for each stop to find an alternative location to swap the stops within the same routes. This results in a complexity of O(|r|·|n|·|n|), simplifying to o(n^2).

- **Swap between routes:**

Involving an intra-route loop for each route, nested with a for loop for each stop. Additionally, there are loops for each route and each stop to find an alternative location to swap the stops between different routes. The overall complexity is O(|r|·|n|·|r|·|n|), also simplifying to o(n^2).

- **2-opt within and between routes:**

The 2-opt heuristic requires o(n^2) time to examine the entire neighborhood of a solution.

2-3-5- Allocation heuristic sub-problem

Student allocation is considered in two positions: firstly, in the constructive phase where stops are selected, routes are constructed, and the solution's feasibility with respect to student allocation is maintained; secondly, during the second level of the oscillating strategy stage when the current solution undergoes re-optimization. Before applying remove and redistribution operators, a feasibility check or student re-allocation is necessary. However, during the first stage of OLS in our algorithm, feasibility checking is not required due to its nature.

The heuristic for student-to-stop assignment in the constructive phase operates as follows: each student is assigned to a potential stop based on reachability, with a preliminary stage generating a list of stops within the maximum walking distance.
The student list is then sorted by the increasing number of allowable stops. The heuristic allocates students sequentially from the top of the list to the first available stop within reach. This prioritizes critical students with few allowable stops. However, a drawback arises when no stop with available capacity is left for some students, triggering a repair procedure to allocate unassigned students.

To address this, congested stops with no remaining capacity for unassigned students are identified. A list is created for students already assigned to these stops, determining the highest number of non-congested alternative stops where students can be reallocated. A student is then randomly selected from this list and moved to another non-congested stop, creating space in the congested stop. This process continues until the list of unassigned students is empty.

To better understand student allocation heuristics in the constructive mechanism, consider examples in Table 2-2. It illustrates the student allocation problem during the constructive phase with 20 students, 4 stops, and a single school. Before generating routes, each stop is considered in one route. The matrix (Table 2-2(a)) allocates students to potential stops, with 1 indicating walking distance and 0 otherwise. In Table 2-2(b), students are sorted by allowable stops. Allocation begins from the top of the list to the first available stop, shown in Table 2-3.

The reallocation mechanism used before remove and redistribution operators differs slightly from the initial allocation in the constructive phase.
The remove operator aims to remove stops, reducing total travel distance. Before applying this, a student allocation sub-problem is solved, and if feasible, the removal is executed.

Specifically, in the remove operator, when selecting a stop for removal, the students assigned to it must be reassigned to a new stop or distributed among existing routes by solving an allocation sub-problem. The sub-problem proceeds as follows: first, choose the stop to be removed and create a list of students assigned to it. Sort these students based on an increasing number of stops in same or different routes where they can walk. Determine the number of stops where they can potentially be reallocated. Then, select a student from the top of the list and reallocate them to another alternative stop in a different route, ensuring capacity constraints are met. This process continues until all students in the list are investigated. If all students can be successfully assigned to potential stops, the remove operator is executed; otherwise, it is discarded.

The redistribution operator aims to balance student distribution among routes, countering the imbalance introduced by inter-route operators. This imbalance can result in some routes having many students while others have only a few. To rectify this, the redistribution operator optimizes the current capacity distribution among routes, minimizing deviation in route loading values by proportionally distributing students. It assumes the total number of students and occupied capacity of each route are represented by n_s and c_r, respectively, with A_{lr} denoting the average loads on all generated routes. The method starts by creating a list of routes ranked in decreasing order of occupied capacity. This list comprises routes with an occupied capacity greater than A_{lr}.

The process involves selecting a route from the top of the list and transferring some students to another route through a student allocation sub-problem. This iteration continues until $c_r - A_{lr} \leq 0$ becomes less than or equal to 0. If the selected routes cannot find positions for potential students, the next route in the list is considered. Importantly, this procedure influences the desirable distribution of students in the current solution.

Prior to implementing the redistribution operator, it is crucial to conduct a preliminary analysis to assess the significance of balancing students for the given problem. This analysis aims to determine whether the presence of a redistribution operator worsens solutions. If a redistribution operator is employed, it becomes essential to identify its optimal placement within the algorithm—whether in the improvement stage or diversification stage. To investigate this, a pilot analysis is conducted on a subset of the sample, revealing that the presence of the redistribution operator in the improvement stage led to a superior solution compared to cases where this operator was inactive.

The complexity of remove and redistribution heuristics comprises three stages: 1) feasibility check, 2) student allocation (if feasibility is not satisfied), and 3) move operation. The following calculations are presented to determine the total complexity of each operator:

- ❑ To execute a move, a for loop is employed for each route to remove one stop, resulting in a complexity of O(|r|·|n|), which is approximated as $o(n)$).
- ❑ For student allocation, considering each stop intending to be removed in a route, where student |s| can be assigned to another stop, the complexity is O(|n|·|s|·|r|·|n|), approximated as $o(n^2)$.
- ❑ The feasibility check ensures that a student can be allocated to an allowable stop in the same or another route, with a complexity of O(|r|·|n|), approximated as $o(n)$.

Therefore, the total complexity encompasses move complexity $o(n)$+ student allocation $o(n^2)$+ feasibility check $o(n)$. For the redistribution operator, total complexity includes student allocation o(n^2) + feasibility check o(n).

Table (2-2) Example of Student Allocation Problem before Constructive Phase

a-Matrix of student–stop allocation candidate

Student \ stop	1	2	3	4
1	0	0	0	1
2	1	0	0	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	0	1	0	0
7	1	1	1	0
8	0	0	1	0
9	0	0	1	0
10	0	0	1	0
11	0	0	1	1
12	0	0	0	1
13	0	1	0	0
14	1	0	0	0
15	0	1	1	0
16	1	1	0	0
17	0	0	1	0
18	1	0	1	0
19	0	0	1	1
20	0	1	0	0

b-No of available stops for each student

No of Students	No of allowable stops in increasing order
1	1
3	1
5	1
6	1
8	1
9	1
10	1
12	1
13	1
14	1
17	1
20	1
2	2
4	2
11	2
15	2
16	2
18	2
19	2
7	3

Table (2-3) Matrix of Allocating Student to Allowable Stops

List of allocated students to the allowable stops before applying repair operator, resulting to infeasible solution

No of students	Allocated stops
(2,5,14,16,18,7)	1
(3,4,6,13,20)	2
(8,9,10,15,17)	3
(1,11,12,19)	4

2-3-6- Restore operator

While the algorithm runs, a scenario may arise where λ becomes large, reaching the predefined value of λ_u, indicating that the current solution is far from feasible.
To address this, a restore operator is implemented in two steps, contingent on how infeasible solutions are detected. Initially, routes are sorted by increasing violated bus capacity. Subsequently, the procedure attempts to allocate students to allowable stops in other routes when possible.

This repeats until all allowable stops and routes are examined. If, at the first step's end, no route is violated, a feasible solution is found; otherwise, the solution proceeds to a second step, the split procedure, addressing remaining routes violating bus capacity constraints.

This step addresses routes still violating the bus capacity constraints, emphasizing the crucial task of dividing these routes into two or more sub-new-routes accurately. For each infeasible route, a specific number of stops is removed and inserted into the new route. The process involves calculating the ratio $r_i = \frac{d_i}{rg_i}$ for each stop i within the route, where d_i is the demand of each stop (the number of students allocated to the stop), and rg_i is the removal gain. The gain rg_i is defined as: $dist(prev(i), i) + dist\big(i, next(i)\big) - dist(next(i) - prev(i))$.

The list of stops is then sorted in ascending order of r_i, with the intention of serving the first stop with the lowest demand and the highest removal gain. Starting from the first stop on the list, stops are successively removed and inserted into the new route. This method iterates until the considered route satisfies the bus capacity constraints.

The execution of the restore operator involves addressing interrelated sub-problems 1 and 3 simultaneously at the same level.

Specifically, to handle infeasibility, each infeasible route must first identify the set of allowable stops in the other route (representing sub-problem 1). Additionally, if a student is allocated to a stop in the candidate route, it results in an infeasible solution, necessitating the exploration of alternative possibilities. Consequently, the selection of a bus stop (sub-problem 1) and the selection of the route containing this bus stop (sub-problem 3) need simultaneous consideration at the same level. Sub-problem 2 solely determines whether the selected stop on the relevant route is feasible or not.

2-4- Metaheuristic structures and diversification strategies

Following the intensification phase, a local optimum is reached, prompting the metaheuristic to escape from local optimum by perturbing the current solution. This increases the chance of reaching a global optimum through local search from the perturbed solution.

Diversification mechanisms aim to provide a good starting point for the intensification stage. Two variants, Repair-Destroy and Double Swap, are used. In case one operator struggles to escape local optima, the other may yield more efficient outcomes. Both diversifications are applied through a random selection mechanism.

2-4-1- Destroy and repair method

The perturbation mechanism, known as the destroy-and-repair method, iteratively explores different solution space parts by removing and reinserting stops. It considers two input parameters: the percentage of routes to be destroyed (ε) and the total number of routes in the current solution (k). This mechanism utilizes a destroy-and-repair operator during the perturbation heuristic.

In the destroy phase, a random route is selected; all stops are removed and added to the list of non-visited stops (U). This step is repeated $\varepsilon * k$ times. In the repair phase, a new solution (x) is generated by creating routes that include all non-visited stops from the U list.

These routes are generated using the nearest neighborhood with a greedy randomized selection mechanism. The next stop in the route is randomly chosen from a restricted candidate list containing the first α closest non-visited stops until the list is empty.

2-4-2- Double swap

The method involves repeating a swap movement twice between randomly selected routes. During each iteration, the swap operator exchanges the positions of two stops between different routes. The double swap operator, working on the basis of exchanging the positions of two stops between routes, may result in infeasible cases. Specifically, stops with a large number of students to be exchanged between routes might lead to infeasible solutions. This situation could arise to the extent that neither the constructive nor the diversification heuristics ensure a feasible solution, contrary to expectations from the oscillating local search heuristic.

2-5- Computational experiments

This section outlines the experiments conducted to evaluate the two metaheuristics, N-ILS and I-ILS, for the SBRP. Using two datasets, the first dataset (Dataset I) comprises 104 instances from benchmark instances proposed by Schittekat (2013). These instances are categorized into small (set S, 5 to 10 stops), medium (set M, 20 stops), and large (set L, 40 to 80 stops) instances.

The instances vary in features such as the number of stops, students, bus capacity, and maximum walking distance. The instances can be downloaded from http://antor.uantwerpen.be/metaheuristic-for-the-school-bus-routing-problem-with-bus-stop-selection/.

The second dataset (Dataset II) consists of 30 newly generated instances. Instances in this dataset have a problem size with the number of stops ranging from 6 to 8, the number of students ranging from 30 to 80, and walking distances ranging from 5 to 40 (The values of walking distance are determined based on the scale of x_{max} and $y_{\max}$).

Similar to Dataset I, only one school is considered for these instances.

The purpose of creating a new dataset is twofold: first, to broaden the scope of our work; and second, to demonstrate, for a new sample, the extent to which the algorithm can deviate from the exact solution.

To generate Dataset II, five parameters per instance are defined in the first stage: n_p (number of stops), n_s (number of students), x_d, y_d (x and y coordinates of the school), and w_{max} (maximum walking distance for each student to reach a bus stop). The school's coordinates are set at (75,75). All instances are randomly generated in the Euclidean square between (0,0) and (x_{max}, y_{max}), where (x_{max}, y_{max}) is set to (150, 150). The procedure for creating Dataset II follows the approach proposed by Schittekat et al. (2013).

The experimental analysis unfolds in three stages. In the first stage, the crucial components of each metaheuristic, as outlined in Section 2.5.1, are examined and fine-tuned. The goal is to ensure that each metaheuristic, when applied to instances drawn from Dataset I, generates the best solutions on average. In the second stage, after determining the optimal parameter settings for each metaheuristic, we compare our solution methods (I-ILS and N-ILS) with the best-known solutions from the literature (i.e., those found by the metaheuristic as published in Schittekat et al., 2013).

All instances used in the second stage are from Dataset I. In the third stage, we use 30 instances from Dataset II for comparison.

Since Schittekat et al. (2013) did not provide solutions for Dataset II, we aim to compare our methods with the exact solutions found by the GAMS software. The first stage is detailed in Sections 2.5.1 and 2.5.2, while the second and third stages are covered in Sections 2.5.3 and 2.5.4, respectively.

It is worth mentioning that before focusing on designing a metaheuristic with strategic oscillation, several experiments have been conducted to understand the performance of the proposed algorithm and determine whether strategic oscillation is employed.

In practice, an analysis is carried out to investigate the algorithm's performance while the bus capacity constraints are relaxed, either by means of strategic oscillation or not. It is observed that applying tight capacity constraints, on average, provides a worse solution than the strategic oscillation approach. Therefore, in this chapter, only the results of the metaheuristic with the strategic oscillation are presented.

2-5-1- Parameter configuration

Both metaheuristics, detailed in sections 2.3 and 2.4, involve crucial parameter settings. The goal is to pinpoint the components significantly impacting solution quality and computation time. To calibrate these metaheuristics, a subset of instances is subjected to a full factorial experiment, encompassing all parameter combinations. The test set comprises 14 instances (8 from set S, 4 from set M, and 2 from set L), each run five times. Table 2-4 provides a concise overview of tuned parameters and their tested values. Notably, two performance metrics are considered: average solution cost and average computation time.

Notably, the maximum number of iterations is not included in the parameters analyzed list, as a higher number of iterations yields improved solutions at the expense of longer computational time. Therefore, in this stage, the number of iterations is fixed at 300. In the subsequent phase, tests are conducted to evaluate algorithm convergence (refer to Section 2.5.2).

The multi-way ANOVA method, implemented through SAS software, analyzes these runs and provides the P-value of the F-test (see Table 2-5). A bold P-value indicates a significant effect of the associated parameter on both solution cost and computation time. Table 2-6 displays the optimal parameter settings for both N-ILS and I-ILS.

Table (2-4) Heuristic parameters and the tested levels

Parameters	Description	Value	No of levels
Repetition	Number of iterations	300	1
N1=Relocate- within	Relocate intra-route operator	On, off	2
N2= Relocate - between	Relocate intra-route operator	On, off	2
N3=Swap - within	Exchange intra-route operator	On, off	2
N4= Swap-between	Exchange intra-route operator	On, off	2
N5= Two -opt- within	Two -opt intra-route operator	On, off	2
N6= Two -opt - between	Two -opt intra-route operator	On, off	2
N7=Redistribution	Student transfer operator	On, off	2
N8=Remove	Remove operator	On, off	2
ε	Maximum percentage number of routes in best solution found so far to be destroyed	10%,15%,20% 25%,30%, 40%, 50%	7
λ_0	Initial penalty	0, 1, 2, 5, 10, 100	6
β	The multiplicative factor employed to increase the penalty	1,2,5,10	4
α	Size of the restricted candidate list	1, 2, 3, 4, 5	5

Table (2-5) P-Values of F-tests

Parameters	Average solution cost	CPU time
N1=Relocate- within	**<0.05**	**<0.05**
N2= Relocate - between	**<0.05**	**<0.05**
N3=Exchange - within	**<0.05**	**<0.05**
N4= Exchange-between	**<0.05**	**<0.05**
N5= Two -opt- within	**<0.05**	**<0.05**
N6= Two -opt -between	**<0.05**	**<0.05**
N7=Redistribution	**<0.05**	**<0.05**
N8=Remove	**<0.05**	**<0.05**
ε	**<0.05**	**<0.05**
λ_0	<**0.05**	<**0.05**
β	0.3765	0.1247
α	**<0.05**	0.0654
$\lambda_0 * \varepsilon$ (interaction of 2 parameters)	**<0.05**	**<0.05**

Table (2-6) Best parameter settings for both metaheuristics

Parameters	I-ILS	NILS
N1 =Remove_ Insert -within	On	On
N2=Remove _Insert -between	On	On
N3=Exchange -within	On	On
N4= Exchange-between	On	On
N5=2-opt. -within	On	On
N6= 2-opt -between	On	On
N7=Redistribution	On	On
N8=Remove	On	On
ε	25%	30%
λ_0	2	1
β	5	2
α	3	2

The P-value analysis indicates that all local search operators, the percentage of routes to be destroyed (ε), and the initial penalty value (λ_0) are crucial parameters influencing both solution quality and computation time. Unexpectedly, neither the multiplicative factor (β) nor the size of the restricted candidate list (α) impact computation time.

This suggests that metaheuristics performance is insensitive to the values of (β) and α.

Furthermore, the ANOVA results highlight the significant interaction between parameters ε and λ_0 ($\lambda_0 * \varepsilon$), affecting both solution quality and computation time. Hence, the algorithm's ability (λ_0) to transition between feasible and infeasible solution spaces and the percentage of routes to be destroyed (ε) are key features of our metaheuristics. The focus is on exploring the infeasible solution space, regardless of accelerating the transition speed from infeasible to feasible regions.

2-5-2- Effect of the number of iterations on the performance of metaheuristics

In this section, we examine the impact of the number of iterations on the performance of I-ILS and N-ILS metaheuristics. To this end, balancing computing time and solution quality is crucial. Both metaheuristics are executed 10 times on instances from Section 2.5.1, varying the number of iterations (ϕ) with values of 100, 200, 400, 800, 1000, and 1200.

Other parameters use values from Table 2-6, optimized through full factorial analysis. For each test set (12 instances), we report: (1) the percentage gap between the best solutions after 10 runs and the best-known SBRP solutions, indicating algorithm's capacity to discover better solutions; (2) the percentage gap between the average cost of the solutions after 10 runs and the best-known SBRP solutions, reflecting robustness.

The best-known solutions are sourced from Schittekat et al. (2013), and the combined results are presented in Table 2-7. In the table, the second column denotes the used metaheuristic, the third column shows the percentage of the best gap (% Best Gap), the subsequent column represents the average percentage gap (% Avg. Gap), and the final column indicates the total computing time for solving 12 instances.

Larger values of ϕ enhance solution quality and metaheuristic robustness. However, an increase in ϕ from 100 to 200 results in a roughly 1.9-fold increase in computational time. Further, as ϕ goes from 200 to 400, the computational time rises by a factor of 1.43 (for I-ILS) and 1.34 (for N-ILS).

This behavior primarily stems from the stopping criterion embedded in the OLS heuristic. The heuristic terminates execution when it either cycles around the same feasible local optimal solution (repeated exploration) or explores an unpromising region of the solution space containing a weak feasible solution.
Overall, the high quality of the initial solution ensures that the OLS heuristic requires fewer iterations to achieve better results.

Consequently, shorter execution times are needed, especially when the search explores the most promising region of the solution space, closer to the optimal solution. The likelihood of this exploration increases with a significant number of iterations.

Table (2-7) Computational results

ϕ	Metaheuristic	%Best Gap	%Avg. Gap	Time (in seconds)
100	N-ILS	3.25	4.46	47
	I-ILS	3.50	4.68	51.97
200	N-ILS	2.53	3.88	90
	I-ILS	2.59	4,11	98
400	N-ILS	2.18	3.34	121
	I-ILS	2.23	3.58	140
800	N-ILS	1.98	3.11	189
	I-ILS	2.08	3.21	230.5
1000	N-ILS	1.96	3.08	245.78
	I-ILS	1.99	3.16	312.45
1200	N-ILS	1.91	3.01	387.89
	I-ILS	1.96	3.13	507.56

In essence, setting the iteration value higher than 800 yields marginal improvement in both % Best Gap and % Avg Gap, while also increasing computing time. Thus, to strike a balance between computing time and solution quality, a number of iterations around 400 seems optimal for most instances in our problem. Table 2-7 indicates that N-ILS outperforms I-ILS across all metrics. Moreover, in terms of computing time, N-ILS is, on average, 19% faster than I-ILS.

2-5-3- Comparison of the metaheuristics on the basis of dataset I

With the optimal parameter settings established for each solution approach, a comparison between I-ILS and N-ILS is conducted, evaluating solution quality and computation time across small, medium, and large instances. Both metaheuristics are implemented in Java. For testing and comparison, each metaheuristic underwent 10 runs on all instances, with results compared against the best-known solutions identified by Schittekat (2013).

The experimental analysis covered 104 instances categorized into three subsets: Set S (48 instances with 5 to 10 stops), Set M (24 instances with 20 stops), and Set L (32 instances with 40 to 80 stops). Additionally, walking distances of 5, 10, 20, and 40 were considered.

Appendices 1 and 2 present the experimental results for each metaheuristic configuration, utilizing the parameter settings outlined in Section 2.5.
Aggregated results for each instance subset are summarized in Table 2-8(a). For both metaheuristics, this table illustrates the percentage gap between the best solutions found after 10 runs and the best-known solutions averaged across instances in sets S, M, and L.

Table 2-8(b) depicts the percentage gap between the average solution cost after 10 runs and the best-known solutions, averaged over all instances for sets S, M, and L. This reflects the robustness of each metaheuristic configuration. To ensure a fair comparison, a fixed number of iterations (400) is set for both I-ILS and N-ILS in solving each instance, with the best-known solutions sourced from Schittekat et al. (2013).In Table 2-8(a) and (b), the first column denotes the utilized metaheuristic, while the subsequent three columns categorize the instance sets (from small to large). Table 2-8(a) reveals that for small and medium sets, N-ILS generally produces lower average percentage gaps from the best-known solutions, while I-ILS tends to yield smaller gaps for instances in set L. In terms of robustness, N-ILS outperforms I-ILS on average for small, medium, and large sets.

On average, N-ILS excels over I-ILS concerning solution quality (best gap from the best-known solutions: 2.08% for N-ILS, 2.17% for I-ILS) and robustness (average gap from the best-known solutions: 3.29% for N-ILS, 3.48% for I-ILS).

Regarding computing time, Table 2-9 indicates that I-ILS is 10% slower than N-ILS. The computational results indicate that, for finding optimal solutions with each metaheuristic, the N-ILS metaheuristic outperforms the ILS metaheuristic. Specifically, 23 instances align with the best-known solutions, and improvements are observed in 2 instances. Following closely, the I-ILS is ranked second, with 19 instances matching the best-known solutions.

Table (2-8) Results obtained by solving the instances contained in Sets S, M, and L

Metaheuristics	Set S	SET M	SET L	Average
(a). Best gap from best-known solutions				
N-ILS	1.83%	2.09%	2.31%	2.08%
I-ILS	2.04%	2.19%	2.29%	2.17%
(b). Robustness of each metaheuristic				
N-ILS	3.06%	3.29%	3.53%	3.29%
I-ILS	3.32%	3.51%	3.60%	3.48%

Table (2-9) Total Computing Time in SET S, M and L in Seconds

Metaheuristic	SET S	SET M	Set L	All
N-ILS	347.33	1799.57	53326.50	55474
I-ILS	369.22	1928.52	59064.56	61362

2-5-4- Comparison of the metaheuristics on the basis of data Set II

The effectiveness of the proposed heuristic for SBRP is examined using the newly generated instance (data set II). We applied I-ILS and N-ILS metaheuristics to solve all 30 instances. Each metaheuristic underwent 10 runs on all instances, and the results are compared with the exact solutions obtained by the GAMS/CPLEX solver.

Table 2-10 summarizes the aggregated results for the instances. The %Best Gap calculation method is consistent with that presented in section 2.5.3.

Table 2-10 reveals that, in terms of the best gaps from the exact solution, N-ILS produced solutions with an average value at least 0.29% lower than that of the I-ILS heuristic. Additionally, N-ILS achieved optimal solutions for 33% of the instances and obtained relatively small optimality gaps in all other cases, while this number is 7 (23%) for I-ILS.

Table (2-10) Results obtained by solving the instances contained in data set II

Metaheuristic	Best gap from best-known solutions (Percent)	No. of optimal solution
N-ILS	0.91 %	10/30
I-ILS	1.20%	7/30

2-6- Conclusion

In this chapter, we have introduced two metaheuristics for solving the school bus routing problem, both centered around an oscillating local search with three key features. First, the metaheuristics explore infeasible segments of the solution space. Second, a restore operator is employed to navigate back to the feasible portion when violation increases. Lastly, set of neighborhoods are utilized to enhance exploration in a vast search space.

Experiments are conducted on two datasets: data set I, comprising 104 instances from the benchmark introduced by Schittekat et al. (2013), and data set II, consisting of newly generated instances. For data set I, the formulation presented in Section 2.2 is accurately solved using the CPLEX solver in GAMS software.

Two distinct metaheuristics are proposed to address small, medium, and large instances, as only the 43 easiest cases are solvable using an exact method.
Statistical analysis is performed for each metaheuristic to optimize heuristic parameters. After determining the best parameter settings, a comprehensive comparison is conducted for all instances, considering solution quality, robustness, and computing time.
The results of the computational experiments imply that N-ILS excels (concerning the quality of the solution, the number of optimal solutions, and computing time) more than I-ILS in comparison to the metaheuristic presented by Schittekat et al. (2013).

Future research could focus on incorporating additional constraints and features to better model real-life situations. Another research avenue involves exploring more efficient ways to check the feasibility of student allocations before applying each improvement operator. This could include investigating the use of efficient data structures to reduce the computational complexity of local search operators.

Chapter 3:

Adaptive large neighborhood search for school bus routing problem with bus stop selection

3-1- Introduction

As mentioned earlier, the second and third chapters explore the current problem (Schittekat et al., 2013) through various solution approaches. In this chapter, we avoid detailed explanations on the problem definition, mathematical modeling, and literature review covered in the second chapter. Here, the emphasis is on presenting the proposed solution for addressing the current problem (Schittekat et al., 2013).

In the preceding chapter, we have established that the exact method is proficient in achieving optimal solutions, yet it's limited to solving problems with a restricted number of stops/students (up to 43 instances). Real-world scenarios often involve hundreds of stops/students, surpassing the method's applicability. Hence, heuristic approaches become imperative to address larger instances and secure nearly optimal solutions within reasonable time frames.

Recently, various heuristic variants have been employed for Vehicle Routing Problems (VRP), with local search heuristics demonstrating notable efficiency. However, in cases of stringent constraints, implementing local search becomes less advantageous. In this case, transitioning from one promising point to another poses challenges.

One strategy is to explore an infeasible portion of the solution space as a new promising area. While previous studies focused on heuristics limited to the feasible search space, some researchers advocate considering both feasible and infeasible areas for effective problem-solving.

Brandoa (2006) introduces strategic oscillation, a local search strategy efficiently switching between feasible and infeasible search areas. The aim is to oscillate between feasible and infeasible spaces, with control over the number of movements between these states. The concept of strategic oscillation within the solution space is not novel in Vehicle Routing Problems (VRP) (e.g., Nagata and Braysy, 2009; Toth and Vigo, 2003; Cuervo et al., 2014).

An alternative strategy involves utilizing large standard moves instead of smaller ones. While larger moves may extend computing time, they often yield more favorable results in terms of solution quality.

The Large Neighborhood Search (LNS) operates on the principle of a large-scale neighborhood mechanism, employing construction and destruction principles throughout the search procedure. Each iteration involves selecting a heuristic to destroy a portion of current solutions and another to repair it by creating a new solution. In solving the SBRP, a new solution is initially derived by removing a number of stops and then reinserting them anywhere in the current solution, aiming to optimize it. Remarkably, LNS, particularly in its large neighborhood search aspect, yields result compatible with Vehicle Routing Problems.

The Adaptive Large Neighborhood Search (ALNS), an adaptive form of LNS, has demonstrated excellent solutions across various VRP scenarios (e.g., Ropke and

Pisinger, 2006; Azi et al., 2010; Ribeiro and Laporte, 2012). ALNS incorporates multiple destroy and repair heuristics, with the probability of selecting a heuristic for a new solution based on its past performance in earlier iterations.

To address the mentioned issue, the we employ ALNS and LNS algorithms, representing a novel approach in SBRP studies. These algorithms are detailed in the following section. The method's efficiency was evaluated by comparing it with existing metaheuristics across 104 instances. While ALNS and LNS heuristics with strategic oscillations are not new in VRP literature, this study, to our knowledge, presents the first application in the context of SBRP.

The contributions of this work include:

1) Solving SBRP using ALNS and LNS heuristics with strategic oscillation.
2) Assessing algorithm performance with and without strategic oscillation.
3) Analyzing the impact of different acceptance criteria on the average solution cost for each metaheuristic.
4) Comparing all solution approaches with the best-known solutions outlined in Schittekat et al. (2013).

3-2- Solution approach

Our findings indicate that an exact method effectively handles only 43 small instances (up to 10 stops and 200 students) within a reasonable time, making it impractical for larger cases (see Appendices 3 and 4). Metaheuristic algorithms offer a practical solution. For efficient and near-optimal SBRP solutions within a reasonable computation time, we introduce simple Large Neighborhood Search (LNS) and its adaptive variant, Adaptive LNS (ALNS), detailed in Algorithm 3-1, with LNS presented after ALNS.

The ALNS metaheuristic follows a three-stage process. In the first stage, the Clarke and Wright algorithm with a greedy randomized selection mechanism (CR.g) heuristic is employed to find an initial solution (refer to Section 3.2.1). This solution serves as the input for the second stage, known as the improvement configuration, which includes a set of large neighborhood structures comprising removal and insertion operators.

Four removal and four insertion operators are utilized. The removal heuristics, explained in Section 3.2.2.1, remove stops and requests from the current solution. The insertion heuristics, detailed in Section 3.2.2.2, then insert them back. Algorithm 3-1 operates by creating an initial solution x_o with the CR. g heuristic, followed by employing the improvement heuristic after the construction of the initial solution.

In the improvement stage, extensive exploration of large neighborhoods occurs through destroy (using removal heuristics) and repair (using insertion heuristics) mechanisms. Selection of removal and insertion heuristics at each iteration is governed by a roulette wheel mechanism, adjusting probabilities based on past successful operations. Each operator receives a score reflecting its past success in improving the solution, influencing higher probabilities for selection in subsequent stages.

Insertion and removal heuristics are chosen and weighted independently (refer to Section 3.2.3 for details).

Applying the roulette wheel mechanism involves selecting one removal and one insertion heuristic at each iteration. The removal operator removes q stops from the current solution, followed by employing the insertion heuristic to reinsert them into the current solution.

At each iteration, a specified number of q stops is disconnected using a removal heuristic and placed in the stops pool, referred to as the U bank list.

Subsequently, stops in U are reintegrated into the solution using the insertion heuristic. The parameter q, where $q \in \{0, \ldots\ldots n\}$, is pivotal, influencing the effectiveness and efficiency of our solution approach, with n being the number of stops in the problem. Essentially, q determines the neighborhood size.

Setting q to zero implies no search in the solution space, while q set to n transforms the algorithm into a multi-start, solving the problem a new. This parameter significantly impacts solution iteration, requiring low q levels for sustained exploration if an acceptable solution is found over several iterations.

Maintaining a low q value proves effective for finding improved solutions, emphasizing an intensification strategy within the solution space. Conversely, if worse solution is obtained over several iterations, increasing q becomes necessary for more efficient exploration of the solution space. This entails removing and re-inserting a larger number of q. Balancing diversification and intensification, we adjust q by initially setting it to q_{min} and systematically modifying it throughout the Adaptive Large Neighborhood Search algorithm.

Crucially, q 's update is contingent on the acceptance of the solution generated in the preceding iteration, ensuring a dynamic adaptation to the evolving search landscape.

The solution derived from any removal and insertion heuristic undergoes an acceptance criterion outlined in Subsection 3.2.4. If the solution x^*_{act} is accepted, the value of q is reset to q_{min}.

Furthermore, upon accepting x^*_{act} to update the penalty value α, the following scenarios are examined:

1) If the solution x^*_{act} contains violations of bus capacity (indicating an infeasible solution), the search prioritizes reducing the violated capacity of the route, and the value of α is set to $min\{\alpha_{max}, \alpha\beta\}$. The critical parameters α and β warrant consideration. α reflects the heuristic's ability to explore infeasible portions of the solution space, while β signifies the transition speed from the infeasible to the feasible space, impacting the capability to discover feasible regions with improved solutions.

2) If x_{act}^{*} indicates a feasible solution with no bus capacity violations, the focus shifts to minimizing travel distance, and α is set to $max\left\{\frac{\alpha}{\beta}, \alpha_{min}\right\}$.

In this context, infeasibility is related to the bus capacity constraints. α is linked to the maximum violation of this constraint. A higher α compels the algorithm back into the feasible solution space, limiting exploration of the infeasible segment. Conversely, a smaller α prompts exploration of a larger portion of the solution space. The value of α is set to α_{min} initially and then systematically updated by the value of β.

A high β allows the penalty α to increase, accelerating the algorithm toward a feasible solution. This approach directs the search into a broader space, facilitating infeasible moves. Optimal values for α_{min} and α_{max} are discussed in Section 3.3.1.1. If x_{act}^{*} is rejected and $q < q_{max}$, then $q = q + 1$ to efficiently explore the solution space. If x_{act}^{*} is rejected and $q = q_{max}$, q is set to $q = \left[\frac{q_{max}}{\eta}\right]$. This choice is driven by the understanding that when q is close to q_{max}, this large value is insufficient for improving the current solution. Thus, a smaller q can facilitate improvements, as outlined by Ropke and Pisinger (2006).

For the discussed problem with instances ranging from 5 to 80 stops, q_{min} and q_{max} align with the percentage of instances used in the problem. Specifically, q_{min} is set to $q_{min} = [n\xi_{min}]$, and $q_{max} = [n\xi_{max}]$, where n is the number of stops, and $\zeta_{min}$and ζ_{max} control the minimum and maximum percentage of stops to be removed from the solution. Detailed values of $\zeta_{min}$and ζ_{max} are discussed in Section 3.3.1.1.

After updating the value of q, a verification is conducted to determine whether the solution x_{act}^{*} obtained from ALNS is superior to the best feasible solution found thus far. If affirmative, the new solution x_{act}^{*} is accepted, replacing $x_{best\_feasible}$. Simple LNS configuration follows a similar methodology as ALNS, albeit with certain elements omitted from Algorithm 3-1. Specifically, in simple LNS, only one removal and insertion are performed instead of a set. Consequently, for the LNS metaheuristic, lines 16, 33, 35, and 36 of the pseudo-code are not applicable.

Algorithm (3-1) Adaptive Large Neighbourhood Search Metaheuristic (ALNS)

Input: V: Set of all potential stops, S: Set of all students, α_{min} (initial penalty value),R (set of Removal heuristics), I (set of Insertion heuristics), q_{max} (maximum number of stops to be removed), q_{min} (minimum number of stops to be removed)**,** ρ (total number of iterations that contains number of segments),
η (parameter to set q_{low},roulette wheel parameter), π(initial score of heuristic $(I \cup R)$), w (initial weight of removal and insertion heuristic $(I \cup R)$),*nsegs* (number of iterations in each segment).
// Constructive Stage
V_{stop}=allocate all students to the bus stops // Allocating using student allocation heuristic
x_o=Route generation $\left(V_{stop}, S\right)$ // Generating route using (CR.g)heuristic
If *x_o is feasible then*
$x_{best\_feasible} = x_o$
$f_{best-feasible} = f(x_o)$
Else
$f_{best\_feasible} = \infty$
End if
$\alpha = \alpha_{min}$
$x_{act} = x_0$
// Improvement Stage
$q = q_{min}$ Initialize the roulette wheel; initialize the adaptive parameters (π, w)
While Stopping criterion ρ is not met do
For *seg=1 to nsegs* do
Roulette wheel mechanism: Select **Removal** heuristic $h_{rem} \in R$ and **Insertion** heuristic $h_{ins} \in I$ based on scores $\{\pi\}$ and weights w
Remove q requests from solution x_{act} using h_{rem}, creating a partial solution
Insert customers into the partial solution using h_{ins} , creating a solution x_{act}^*
If accept (x_{act}^*, x_{act})then
$x_{act} = x_{act}^*$
$q = q_{min}$
Update the value of the penalty α
Else
If $q < q_{max}$ then
$q = q + 1\%$
Else
$q = \left(\frac{q_{max}}{\eta}\right)$
End if
End if
If $\boldsymbol{f(x_{act}) < f_{best\_feasible}}$ then
$x_{best_{feasible}} = x_{act}^*$
End if
Update the collected scores of h_{rem} and h_{ins}
End for
Update (weights w of h_{rem} and h_{ins})
Set scores $\pi = 0$ of h_{rem} and h_{ins}
End while
Output: Report best solution x_{best} found

The cost function is similar first chapter and addressed as follows:

$$cost(x, \alpha) = cost(x) + \alpha d(r) \tag{3-1}$$

The objective function comprises two terms: the first represents the total distance covered by all buses, while the second term, $d(r)$, accounts for the cumulative excess loads carried by each bus (bus capacity violations). This excess load is multiplied by a penalty term, α. This demonstrates that our search isn't confined to feasible solutions but operates with an oscillation strategy. In this approach, the bus capacity constraints are relaxed and incorporated into the objective function as a penalty term. The parameter α acts as a weight for the penalty function, dynamically adjusted within the interval $[\alpha_{min}, \alpha_{max}]$.

In simpler terms, the penalty parameter α underscores the impact of bus capacity violations on the total solution cost. The parameter α is bounded by α_{min} as the lower limit and α_{max} as the upper limit, preventing the weight from reaching infinity. If all bus routes adhere to the capacity constraints, the second term, $d(r)$, is set to zero.
There's a scenario in which, during the algorithm, the value of α might become large, approaching α_{max}. To address this situation, the restore operator comes into play (refer to Section 2.3.6 for more details).

3-2-1- Constructive stage

The Clarke and Wright algorithm is a well-known method used to address the Vehicle Routing Problem (VRP) and its variants (Clarke and Wright, 1964). The standard Clarke and Wright algorithm begins with an initial solution where each stop is visited in a separate route, meaning each stop is assigned to only one bus. Subsequently, the algorithm iteratively merges two routes, providing savings in travel costs.

To accomplish this, a saving matrix ($s_{ij} = c_{io+}c_{oj} - c_{ij}$) is created at the start of the algorithm for each pair of stops. This matrix illustrates the cost reduction achieved by connecting two stops, leading to the consolidation of two routes into one. The savings are then organized in decreasing order.
The two routes containing stops i and j, associated with the saving s_{ij} are merged only if (1) both stops i and j are connected to the depot, and (2) the total capacity linked with the new merged route does not exceed the vehicle capacity.

For SBRP, the original Clarke and Wright algorithm is modified as follows: (i) After the initial setup, where each stop is visited separately, students are allocated to the stops based on a student allocation sub-problem (refer to Section 2.3.5); next (ii) a greedy randomized adaptive search procedure (Feo and Resende, 1995) is employed to establish a proper balance between greediness and randomness.
Initially, the saving matrix is calculated for each pair of stops. Subsequently, a Restricted Candidate List (RCL) is built, containing the α first stop pairs sorted in decreasing order of savings.

In each step, a randomly selected pair from the Restricted Candidate List (RCL) undergoes a merge operation, implementing the associated saving. Post-merge, the RCL adapts based on the new solution configuration. This selection and update process repeats until the list is empty, culminating in a complete solution. The RCL's size, denoted by α, influences the solution construction.

At $\alpha = 1$, meaning that the largest possible saving is considered while building the current solution. Conversely, an increasing α, possibly matching the available saving elements in list, results in a more random construction. α serves as a parameter, providing variability in generating different initial solutions for the SBRP at each run of heuristic.

To save time, a feasibility check precedes the connection of stop pairs. If connecting a pair leads to an infeasible solution, the move is omitted and removed from the Restricted Candidate List (RCL). After the process concludes, the initial solutions undergo an improvement stage. This cycle continues until no non-visited stop pairs

3-2-2- Improvement stage

In this section, we introduce the Adaptive Large Neighborhood Search heuristic (ALNS) along with its components from Ropke and Pisinger (2006). For a given solution x_o, each iteration involves the removal of stops using one of the removal operators, followed by their reinsertion using one of the insertion operators. To enhance the solution, a removal operator is employed to eliminate stops, and an insertion operator is used to reintroduce them. Subsequent paragraphs detail the insertion and removal heuristics, along with other components of ALNS.

3-2-2-1- Removal heuristics

In this section, we outline four removal heuristics inspired by Ropke and Pisinger (2006) and adapted to the school Bus Routing Problem (SBRP). These heuristics, namely Shaw removal based on distance, Shaw removal based on demand, Worst removal, and Random removal, are described.

All heuristics in this section receive a solution and an integer q as input. The output of each heuristic is a solution from which q requests have been removed. Notably, Shaw removal and Worst removal include an additional parameter, p, which dictates the degree of randomization within the heuristic.

- **Shaw removal based on the distance heuristic**

The Shaw removal heuristic, grounded in the concept of similarity, aims to eliminate stops that are close to each other based on their distance in a given solution. This strategy facilitates the restructuring of similar stops, leading to the generation of an improved solution.

The process involves randomly selecting one stop from the U-bank, and if the U-bank is empty, a stop from the current solution is chosen and inserted into U. Subsequently, the degree of similarity between the selected stop in U and all other stops not yet removed in the current solution is measured using the relatedness measure, denoted as R. The relatedness between stops i and j is computed based on their distance from each other, following the method employed by Ropke and Pisinger (2006). The relatedness measure (represented by the Euclidean distance formula) is calculated as follows:
$R(i,j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, where each node $i \in V$, is defined by coordinates (x_i, y_i).

The index for a stop in the list L, encompassing all requests from the solution absent in list U, is determined by Index $_i$ $\{y^p|L|\}$, where y is a random parameter within the range $[0, 1]$, , and p defines the random selection of a stop to be removed (with p representing the degree of randomness). A lower p value signifies greater similarity between stops, while a higher value indicates that selected stops exhibit lower similarity. This process continues until q stops are selected from the current solution and added to the list U. (Refer to Ropke and Pisinger (2006) for further details).

- **Shaw removal based on demand heuristic**

This heuristic follows the Shaw removal based on distance procedure, differing only in the calculation of the degree of similarity between two stops. In this case, the relatedness between stops i and j is measured in terms of the amount of demand to be picked up by bus k, given by:$|\ Di - Di|$. Here, Di represents the number of students allocated to stop i. The objective is to rearrange stops based on the similarity in the allocated number of students, seeking to enhance the solution through this reshuffling process.

- **Worst removal heuristic**

This heuristic selects the q stops with the highest removal gain, defined as $gain(i,x) = cost(i,x) = f(x) - f_{-i}(x)$, where $f(x)$ is the cost of the solution with stop i, and $f_{-i}(x)$ is the cost of the solution without stop i. The rationale is to prioritize the removal of requests with high gain and relocate them within the solution, anticipating that the insertion heuristic can effectively reduce solution costs. Consequently, this removal heuristic focuses on eliminating requests with the highest $gain(i,x)$.

- **Random removal heuristic**

The random removal algorithm randomly selects q stops, removes them from the current solution, and adds them to the list q. It's worth noting that the random removal can be viewed as a special case of the Shaw removal heuristic with p set to 1. Consequently, it can be executed more efficiently than the full Shaw removal heuristic.

3-2-2-2- Insertion heuristics

Following the introduction of the destruction heuristic, the subsequent section details the insertion heuristics employed in the algorithm, drawing inspiration from Ropke and Pisinger (2006) and adapted for the school Bus Routing Problem (SBRP). The core concept of the insertion heuristic is to rearrange stops that are not currently included in the solution or have already been disconnected.

- **Basic greedy heuristic**

Among the four presented insertion heuristics, two follow a greedy construction approach. The basic greedy heuristic identifies the cheapest insertion position for all requests in the list U. Denoting Δf_{ik} as the change in the objective function value resulting from inserting request i into route k at the position that minimally increases the cost of the current solution x, the cheapest insertion cost is given by $f^{+}(i,x) = min_{k\in R}(\Delta f_{ik})$. This process continues until all stops in the list U have been inserted.

- **Second best heuristic**

This heuristic slightly deviates from the basic greedy approach by aiming to insert requests in the second-best position, thereby diversifying the search space. The selection procedure for requests from the list U remains similar to the basic greedy heuristic.

- **Deep greedy heuristic**

The deep greedy heuristic inserts unserved requests into the route in a manner that increases the solution's cost at least. Importantly, the basic and deep greedy heuristic algorithms vary in their approach to selecting a request from the U-bank.
The basic greedy heuristic follows a straightforward process: it selects the first stop from list U and inserts it into the cheapest position. This stop is then removed from the list, and the procedure repeats for the remaining stops in list U.
In contrast, the deep greedy heuristic calculates the lowest-cost solution for all stops in list U beforehand.

Subsequently, the stop with the lowest insertion cost is chosen for insertion. This process continues until no stops remain in list U or it becomes impossible to insert any stops. For a set of stops in list U, the cheapest insertion is formulated as $min_{i\in U} f^{+}(i,x)$. This heuristic diverges from the basic greedy heuristic by calculating the best insertion cost among all requests in list U, instead of just the first one, which leads to increased computational time.

□ Regret-K heuristic

Differing from the greedy heuristic, the regret heuristic extends beyond the best insertion position. To address potential myopic behavior in the greedy insertion heuristic, it explores an alternative insertion possibility. The aim is to insert the stop in an earlier position, recognizing that the better position may be unavailable or inserting the stop later could result in higher costs.

A reinserting order for stops is determined using a regret measure, computed based on the difference between the cost of inserting a request in the best position and the second-best to k_t best position. In essence, the regret heuristic prioritizes stops from the U list with the maximum cost difference.

To implement the regret heuristic, the initial step involves calculating the least insertion cost for all stops in the U list across all routes, similar to the deep greedy heuristic. These values are then sorted in ascending order. Subsequently, the summation of differences between the best and the second to k_{th} best insertion positions is computed, termed the "regret measure."

The stop with the maximum regret measure is selected and inserted into the current solution. This process iterates until no stops remain in the U list.

In each iteration, the regret value seeks stop i that maximizes $max_{i\in U} \sum_{j=1}^{k}(\Delta f_i^k - \Delta f_i^1)$, where Δf_i^k represents the insertion cost for request i in the k_{th} cheapest position. The regret-k-heuristic used in this chapter assumes k to be equal to 2, meaning it calculates the difference between the second-best and the best insertion positions.

3-2-3- Selection of removal and insertion heuristic

In each iteration, the selection of one removal and one insertion heuristic is crucial for the ongoing search process. The choice of these heuristics is influenced by their performance in the preceding iteration. The underlying idea involves assigning a weight to each heuristic, indicating its effectiveness in the previous stage.

To implement this, the entire search is divided into fixed segments, with each segment comprising a set number of consecutive iterations—here, set to 20 iterations per segment. This represents a fraction of the total iterations. In the initial segment, all heuristic scores are set to zero, and each heuristic (both removal and insertion) is assigned an equal weight of $w_i = 1$. At the conclusion of each segment, the weight for each pair of removal and insertion heuristics is updated based on the scores accumulated during that segment.

The score reflects the efficacy of heuristics in the preceding segment, and its utilization is updated based on conditions $\sigma_1, \sigma_2 \; or \; \sigma_3$. These conditions are as follows:

1) When a pair of removal and insertion heuristics achieves the global best solution, its score is increased by σ_1.

2) If it discovers a solution superior to the previous one, its score is set to σ_2.

3) When the pair obtains a solution worse than the previous one but is accepted, the value is increased by σ_3.

Otherwise, the score is set to zero. At the end of each segment, weights are computed based on the obtained scores, and these scores are reset to zero for the next segment. The weights for selecting the heuristic are calculated as follows:

$$w_{i,j+1} = \begin{Bmatrix} w_{i,j} & if \; O_{i,j} = 0 \\ (1-\gamma)w_{i,j} + \gamma\frac{\pi_{i,j}}{O_{i,j}} & if O_{i,j} \neq 0 \end{Bmatrix} \quad (3\text{-}2)$$

The optimal values for σ_1, σ_2 or σ_3 will be discussed in Section 3.3.1.1. In this context, w_{ij} indicates the performance of heuristic i in segment j, and O_{ij} represents how often heuristic i is chosen in segment j. Let $\pi_{i,j}$ denote the previously discussed score for each heuristic during segment j.

The reaction factor, $\gamma \in [0,1]$, signifies how rapidly the weight modification responds to changes in heuristic performance. Specifically, with $\gamma = 0$, we rely solely on the initial weight without considering scores. Conversely, when $\gamma = 1$, we consider scores obtained in the last segment. Our perspective suggests that the situation between these extremes, where $0 \leq \gamma \leq 1$, should be considered. Determining the optimal value for the reaction factor is explored in Section 3.3.1.1.

After determining the weight for each pair of heuristics, the next step involves selecting one removal and one insertion heuristic for use throughout the search. It's important to note that removal and insertion are chosen independently.

The selection of removal and insertion heuristics is inspired by a roulette wheel mechanism. This mechanism picks the candidate heuristic from the set of removal or insertion heuristics based on the following probabilities:

P (Removal heuristic i to be chosen in the segment j) $= \frac{w_{ij}}{\sum_{i \in I} w_{ij}}$

P (Insertion heuristic i to be chosen in the segment j) $= \frac{w_{ij}}{\sum_{i \in R} w_{ij}}$

Where R represents set of Removal heuristics and I denotes set of Insertion heuristics.

3-2-4- Acceptance criteria

The acceptance criteria of the ALNS algorithm play a crucial role in determining whether a newly generated solution should be accepted or rejected. Various methods can be employed to implement these criteria. One approach involves adopting the better acceptance method, where only a superior solution is accepted. This criterion emphasizes intensification, aiming for improved solutions. On the other end of the spectrum, the random walk acceptance criterion seeks to accept any solution, irrespective of its objective value. This approach promotes diversification, exploring a broader solution space.

Several intermediate methods exist, such as simulated annealing, as introduced by Kirkpatrick et al. (1983). Simulated annealing aims to strike a balance between diversification and intensification contexts, providing a nuanced approach to solution acceptance. The acceptance criterion for the new solution x^* follows a specific procedure. If $f(x^*) < f(x)$, indicating an improvement, the new solution x^* is accepted, replacing the current solution. However, if $f(x^*) > f(x)$, the solution x^* is accepted with a probability $P(acceptance) = e^{-\frac{f(x^*)-f(x)}{T}}$, where T is a parameter initially set to $T^{Initial}$ and gradually decreases throughout the search via a cooling rate $c\epsilon(0,1)$: $T \leftarrow T.C$. As T decreases, the probability of accepting a worse solution diminishes, aiding the algorithm in escaping local optima and introducing more diversity into the solution space.

This chapter focuses on three acceptance criteria. In Section 3.4, the discussion centers on the simulated annealing method. In Section 3.5, three types of acceptance criteria are examined: simulated annealing, better acceptance, and random walk.

3-2-5- Allocation heuristic sub-problem

This chapter specifically addresses the student allocation sub-problem, examining it in two stages:

1. **Constructive phase:** This stage involves the selection of stops.
2. **Restore operator:** Student re-allocation occurs during this phase, aiming to make a feasible solution.

The concept of addressing allocation in both the constructive stage and the restore operator aligns with the framework presented in Chapter 2 (Section 2.3.6).

3-3- Metaheuristic configuration

This section illustrates the functioning of different removal and insertion heuristics within an ALNS metaheuristic. The metaheuristic is implemented in Java, and an experimental analysis is conducted in two consecutive steps.

In the first stage, the focus is on examining and fine-tuning the key components of the ALNS metaheuristic to generate optimal average solutions. Given the similarity in configuration between ALNS and LNS components, this stage exclusively delves into the analysis of ALNS components.

In the second stage, following the determination of optimal parameter values, a comprehensive comparison is conducted among the proposed metaheuristics. This comparison is performed across instances categorized as small, medium, and large containers. The instances used in this analysis mirror the sample data outlined in Chapter 2, comprising 104 samples drawn from benchmark instances proposed by Schittekat et al. (2013) and grouped into sets based on three sizes.

3-3-1- Setup of the experiments and tuning

The analysis unfolds in two sequential stages as follows:

1. **Preliminary stage:** This stage involves the fine-tuning of ALNS parameters that significantly impact its performance, aiming to identify the most effective values.
2. **Second stage:** Once the parameters are tuned, the interaction of candidate removal and insertion heuristics is assessed to determine the optimal combination of operators. To enhance the evaluation process, two distinct representative instances are considered for this preliminary stage.

3-3-1-1- Parameter configuration

The ALNS heuristic relies on numerous components, and optimizing their values is crucial for enhancing algorithm performance. The primary goal is to fine-tune the metaheuristic, achieving the best parameter setting for improved solution quality while maintaining efficient computing time. This involves testing a subset of instances in a full factorial experiment, examining all parameter combinations in Table 3-1. The parameter setting is tested on 10 instances, with 4 from set S, 4 from set M, and 2 from set L.

Each instance undergoes five runs, and we consider the average values for both the objective and computation time. We highlight the parameters influencing the algorithm's behavior: p controls Shaw removal and worst removal; weighted adjustment involves $\sigma_1, \sigma_2, \sigma_3$ and γ; strategic oscillation is influenced by α and β; and controlling the removed stops uses ξ_{min}, ξ_{max}, and η. Table 3-1 displays the optimal settings for the heuristic parameters.

Table (3-1) Heuristic Parameters and best Parameter Setting

Parameters	Description	Value	#	Best value
$\boldsymbol{\rho}$	Defines the number of iterations	350,450,550	3	450
$\boldsymbol{\xi_{min}}$	Introduces minimum percentage of requests, stops, to be removed at each ALNS iteration	2%,5%,10%	3	5%
$\boldsymbol{\xi_{max}}$	Introduces maximum percentage of requests, stops, to be removed at each ALNS iteration	15%,20%,25% ,30%,35%,40 %,45,%,50%	7	35%
$\boldsymbol{\eta}$	Defines Parameter to set q	2,4	2	2
$\boldsymbol{p}$	Is responsible for randomness in the removal process	4,6	2	6
$\boldsymbol{\sigma_1}$	Is the weight adjustment of algorithm in roulette wheel mechanism	40,50,60	3	50
$\boldsymbol{\sigma_2}$	Is the weight adjustment of algorithm in roulette wheel mechanism	20,30,40	3	30
$\boldsymbol{\sigma_3}$	Is the weight adjustment of algorithm in roulette wheel mechanism	1,5,10	3	5
$\boldsymbol{\gamma}$	Is the reaction factor of the weights in roulette wheel mechanism	0.25,0.5, 0.75	3	0.75
$\boldsymbol{\alpha_{min}(\alpha_0)}$	Is initial penalty	1,2,5,10	4	1
$\boldsymbol{\beta}$	Is the multiplicative factor employed to increase the penalty	2,5,10	3	5
$\emptyset$	Size of the restricted candidate list	1,2,3,4	4	2

Table (3-2) p-Values of the F-tests to Determine the Significance of each Term

Parameters	Computing time	Average solution cost
$\boldsymbol{\rho}$	$p<0.05$	$p<0.05$
$\boldsymbol{\xi_{min}}$	$p<0.05$	$p<0.05$
$\boldsymbol{\xi_{max}}$	$p<0.05$	$p<0.05$
$\boldsymbol{\eta}$	0.162	0.084
$\boldsymbol{p}$	0.082	0.079
$\boldsymbol{\sigma_1}$	0.093	$p<0.05$
$\boldsymbol{\sigma_2}$	0.054	0.809
$\boldsymbol{\sigma_3}$	0.115	0.320
$\boldsymbol{\gamma}$	$p<0.05$	$p<0.05$

$\boldsymbol{\alpha_{min}}$	p<0.05	p<0.05
$\boldsymbol{\beta}$	p<0.05	0.402
$\boldsymbol{\phi}$	0.193	p<0.05
$\boldsymbol{\alpha_{min} \times \xi_{min} \times \xi_{max}}$	p<0.05	p<0.05

To analyze the results, a multi-way ANOVA is conducted using the R software, presenting p-values. These values indicate the significance of each parameter on solution quality (measured by the objective function) and computation time. In literature, Multi-way ANOVA, known for detecting significant parameters and their interactions, is deemed suitable (Cuervo et al., 2014).

Multi-way ANOVA is employed to analyze main and interactional effects, detecting correlations between variables and providing additional information in the analysis when variables are correlated. Significant parameters (P-value < 0.05) are highlighted in boldface type in Table 3-2.

The value of $\alpha_{max}$is determined as $125 \times \alpha_{min}$. The Table 3-2 highlights key factors influencing solution quality and computation time: number of iterations, minimum and maximum requests for removal, reaction factor for roulette wheel weight, and initial penalty.

Parameter β exclusively affects computation time, while the interaction of initial penalty values α_{min}, $\boldsymbol{\xi}_{min}$, and $\boldsymbol{\xi}_{max}$ significantly influences both solution quality and computation time. This underscores the importance of the ALNS heuristic's ability to explore the infeasible solution space and the percentage of nodes to be removed. Specifically, emphasis should be on enhancing the exploration power of the infeasible region rather than speeding up the transition between feasible and infeasible boundaries.

Figures 3-1(a-c) depict the interaction of ξ_{min}, ξ_{max}, and , α_{min} on solution quality. It is observed that, for a fixed value of $\xi_{min}$and α_{min}, an increase in the value of ξ_{max}contributes to improving solution quality. Consequently, better solutions are achieved. However, this positive trend reverses when ξ_{max} becomes more than 35%, leading to behavior resembling random restarts, diminishing the impact of large neighborhood search. Notably, increasing α_{min}, regardless of $\xi_{min}$and ξ_{max} values, results in lower solution quality, indicating reduced time for exploration.

Conversely, lower α_{min} values necessitate more exploration of the infeasible region. Ultimately, the metaheuristic performs better with ξ_{min} at 5% and ξ_{max} ranging from 30% to 35%.

Figure 3-2 illustrates the impact of ξ_{max} and α_{min} at a fixed ξ_{min}at 5% on average computing time. Clearly, as ξ_{max} increases, the algorithm demands more computing time to explore the solution space and escape local optima. Conversely, higher values of α_{min} result in lower computing time, as the algorithm allocates less time to search in infeasible segments. Consequently, lower values of α_{min}, as depicted in Figure 3-

2, exhibit better performances, enhancing the capability to explore the infeasible area more effectively.

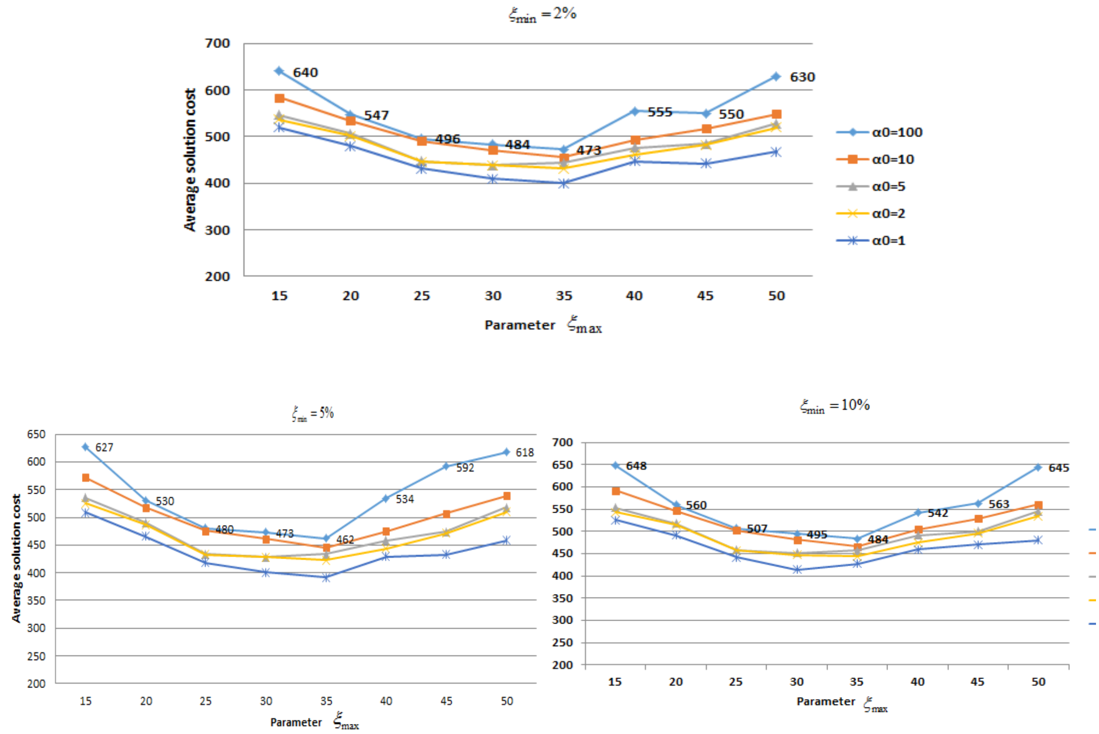


Figure (3-1) Influence of the Parameters ξ_{min}, ξ_{max} and the Initial Penalty $\alpha_{min}(\alpha_0)$ on the Average cost of the Best Solutions Found.

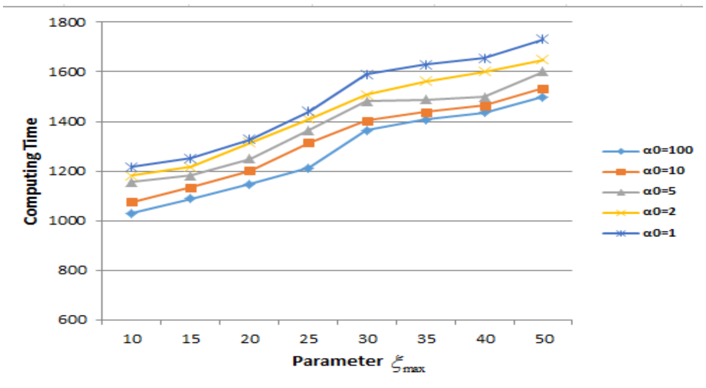


Figure (3-2) Influence of the Parameters ξ_{max} and the Initial Penalty $\alpha_{min}(\alpha_0)$ on the Average Computing Time, ($\xi_{min} = 5\%$)

3-3-1-2- **Analysis of the LNS operators**

In the ALNS metaheuristic, the critical question is identifying the optimal combination of removal and insertion heuristics. While some heuristics may exhibit weaker performance, disregarding them can degrade the overall solution. Although certain neighborhoods may not directly enhance the solution, they can play a crucial role in escaping local optima for other operators in future iterations.

This emphasizes the effectiveness of simultaneously using multiple neighborhoods to empower the solution through diversification or intensification. Achieving a suitable combination of operators, balancing intensification and diversification, yields promising outcomes. However, the drawback lies in the increased complexity and computing time when incorporating a large number of insertion and removal heuristics. Moreover, the selection or design of heuristics should align with the problem structure, emphasizing the need for problem-specific knowledge. These advantages and disadvantages underscore the importance of a meticulous pilot study in the preliminary stages.

To identify the optimal combination of ALNS operators, as discussed in Section 3.3.1.1, a full factorial experiment is deemed necessary. Treatments include scenarios where all neighborhoods (both removal and insertion operators) are deactivated, relying solely on the initial solution. It's crucial to note that all other metaheuristic parameters are set based on the values in Table 3-1. Ten runs are conducted over 10 representative instances from small, medium, and large sets, and the average results for each level are incorporated into the experiment. Figure 3-3 presents key findings that facilitated the definition of our removal and insertion operators. The vertical axis depicts the average percentage gap of results from the best-known solutions, while the x-axis indicates the activation or deactivation of each operator.

This analysis highlights that Shaw removal (based on distance), Worst removal, Random removal, Basic insertion, Deep greedy insertion, and Regret-2 insertion significantly impact solution quality. Removal operators, specifically Shaw and Worst, contribute to intensification, while Random removal adds a diversification mechanism. Similarly, in the insertion mechanism, Basic and Deep greedy operate on an intensification basis, while Regret-2 explores using a diversification mechanism. Notably, Shaw removal (based on demand) and second-best insertion have a lesser effect on solution quality compared to other operators, although they provide a modest improvement when activated. Consequently, excluding these two operators doesn't significantly impact the solution, and their contribution to enhancing solutions appears limited.

Remarkably, among all combinations in our experiment, Shaw (based on distance), Worst and Random removals, and Basic, Deep greedy, and Regret-2 insertions exhibit minimal deviation from the best-known solutions. Consequently, this combination is selected for further analyses. While all considered destroy and repair algorithms contribute positively to solution quality to some extent, our objective is to strike a balance between solution quality and computing time. Therefore, deactivating operators with a lower or weaker impact on solution quality is highly desirable.

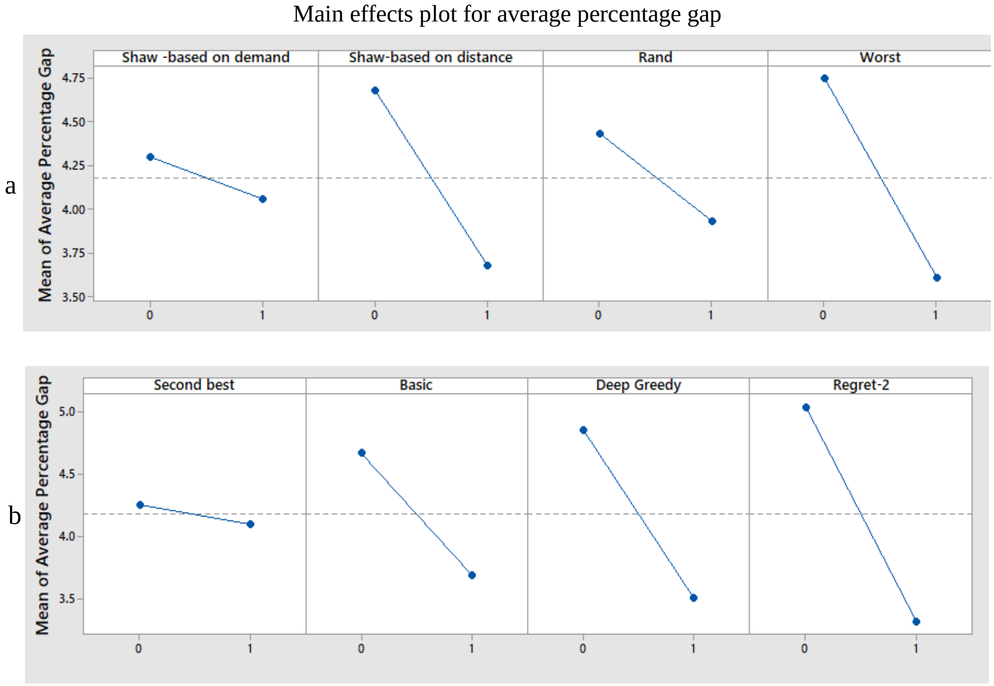


Figure (3-3) Plot of Average Gap from the Best- known Solution Values for Given Heuristic Setting (a: Removal Heuristics, b: Insertion Heuristics)

3-4- Results

Having determined the optimal parameter settings for each solution approach, two stages of experiments are conducted to assess the performance of each heuristic. In both stages, five simple LNS heuristic configurations, each consisting of one removal and one insertion heuristic ("Shaw removal (based on distance) with Regret-2," "Shaw removal (based on distance) with Deep greedy," "Worst removal with Regret-2," "Shaw removal (based on distance) with Basic greedy," "Rand removal with Regret-2"), and the full adaptive LNS with dynamic weight-adjusted (ALNS) are investigated.

3-4-1- Comparison with the best-known solutions

The experiments aim to comprehend algorithm behavior and determine whether the oscillation strategy is is utilized or not. This analysis specifically explores algorithm performance under two scenarios related to bus capacity constraints. The first scenario relaxes bus capacity constraints through the oscillation strategy, while the second maintains tight capacity constraints. The objective is to identify which scenario yields better performance. For precision in tracking the analysis, each scenario is separately investigated in this section.

3-4-1-1- Considering oscillation strategy

In the first stage, each heuristic configuration is evaluated based on the best gap and average gap compared to the results of the best-known solutions presented by Schittekat et al. (2013). This assessment involves tests conducted on a set of SBRP benchmark problems, utilizing 104 instances from the benchmark instances proposed by Schittekat et al. (2013).

Specifically, the %Best Gap represents the percentage gap between the best solutions calculated after 10 runs and the best-known solutions (both proposed heuristic and MIP in Schittekat et al. (2013)) from SBRP instances. This metric relates to the algorithm's ability to find superior solutions. The %Avg Gap calculates the percentage gap between the average cost of solutions after 10 runs and the best-known solutions (both proposed heuristic and MIP in Schittekat et al. (2013)) from SBRP instances, providing insights into the robustness of the algorithm. Detailed experiment information is reported in Appendix 3.

The best-known solutions results are divided into two categories. The first is the best-known solutions from the proposed heuristic (referred to as BKS_{MH}), and the second is the best-known solutions obtained from solving the MIP model (referred to as BKS_{exact}). The results are compared against either of these two benchmarks. Tables 3-3 show the percentage gap from BKS_{MH}, while Tables 3-4 present the percentage gap from BKS_{exact}.

The aggregated findings, representing the average across small, medium, and large instances, are presented in Table 3-3. For each metaheuristic, this table illustrates the gap percentage between the best solutions found after 10 runs and the best-known solutions (BKS_{MH}) from Schittekat et al. (2013).

In Table 3-3, columns two to four represent the removal heuristics, while columns five to seven indicate the insertion heuristic used. Rows differentiate between configurations classified as LNS or ALNS.

Comparing the first configuration among simple LNS configurations, it is evident that, on average, the Shaw removal (based on distance) with Regret-2 insertion heuristic achieves a smaller best gap from the best-known solutions, with a percentage gap of 1.96% for the entire set considered. The Shaw removal (based on distance) with Deep greedy insertions is ranked second.

From the perspective of the average gap, Shaw removal (based on distance) with Deep greedy insertion exhibits superior performance among other LNS frameworks. However, the combination of Rand insertion with Regret-2 performs poorly. In terms of both best gap and average gap, Shaw removal (based on distance) outperforms other removal heuristics.

Comparing the first and second configurations for both the best and average gaps, the results indicate that the ALNS heuristic is capable of finding better solutions than all simpler LNS heuristics. This enhanced performance is attributed to the fact that, on average, ALNS yields satisfactory results for all instances, while individual LNS heuristics may produce suboptimal solutions in some instances. Additionally, recognizing the problem structure appropriately emerges as a critical element, an aspect that has received less attention in the literature.

Indeed, problem-specific knowledge plays a crucial role in designing the structure of metaheuristics efficiently, leading to the development of more effective algorithms. For further insights, readers can refer to Arnold and Sörensen (2017), where the author provides an example of how incorporating problem-specific knowledge contributes to the design of more efficient heuristics. The author emphasizes a strong relationship between the problem's structure and the quality of the heuristics employed.

Table (3-3) The Results Obtained from Solving the Instances Placed in S, M, and L Sets (First Scenario)

(a). Best gap from best-known solutions (BKS_{MH})

Configuration	Removal heuristics			Insertion heuristics			Average over all instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	3.14%
LNS-2			•			•	2.25%
LNS-3	•					•	1.96%
LNS-4	•				•		2.05%
LNS-5	•			•			2.65%
ALNS	•	•	•	•	•	•	1.64%

(b). Average gap from best-known solutions (BKS_{MH})

Configuration	Removal heuristics			Insertion heuristics			Average over all instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	5.11%
LNS-2			•			•	3.75%
LNS-3	•					•	3.13%
LNS-4	•				•		3.06%
LNS-5	•			•			3.91%
ALNS	•	•	•	•	•	•	2.77%

Similar comparisons are conducted concerning BKS_{exact} (refer to Table 3-4). In the case of the best gap, superior performance is observed for ALNS, with LNS-3 heuristic ranking second. Regarding the average gap, ALNS consistently outperforms other configurations.

Table (3-4) The Results Obtained from Solving the Instances Placed in S by Instance 43 (First Scenario)

(a). Best gap from best-known solutions (BKS_{exact})

Configuration	Removal heuristics			Insertion heuristics			Average over instances(percentage)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret - 2	
LNS-1		•				•	2.70%
LNS-2			•			•	2.00%
LNS-3	•					•	1.51%
LNS-4	•				•		1.68%
LNS-5	•			•			2.19%
ALNS	•	•	•	•	•	•	1.42%

(b). Average gap from best-known solutions (BKS_{exact})

Configuration	Removal heuristics			Insertion heuristics			Average over instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret - 2	
LNS-1		•				•	4.61%
LNS-2			•			•	3.66%
LNS-3	•					•	2.71%
LNS-4	•				•		2.67%
LNS-5	•			•			3.63%
ALNS	•	•	•	•	•	•	2.43%

The average computing time for all metaheuristics across all 104 instances is provided in Table 3-5. Notably, among LNS configurations, Shaw removal (based on distance) with any heuristic insertion exhibits significantly longer computing times than others, while Rand removal with Regret-2 demonstrates faster performance. Overall, the ALNS metaheuristic secures the second rank in terms of computing time. This behavior can be attributed to the compound nature of ALNS, incorporating various combinations of removal and insertion heuristics, ultimately requiring a shorter computing time.

Table (3-5) Total Average Computing Time in SET S, M and L in Seconds (First Scenario)

Configuration	Removal heuristics			Insertion heuristics			Total Computing time over sets.
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	54,239
LNS-2			•			•	58,497
LNS-3	•					•	61824
LNS-4	•				•		63095
LNS-5	•			•			59,479
ALNS	•	•	•	•	•	•	56,670

The number of instances matching the best-known solutions (both proposed heuristic and MIP in Schittekat et al. (2013)), along with better solutions found for each metaheuristic, is summarized in Table 3-6. Each instance is executed 10 times.

For the ALNS heuristic, 35 out of 104 instances match with the best-known solutions (BKS_{MH}), and 4 better solutions are also observed. Additionally, Shaw removal with Regret-2 ranks second in performance.

Table (3-6) Number of Matched Solutions with Best-known Solutions (Both Proposed Heuristic and MIP in Schittekat et al. (2013)) and Number of Better Solutions

Metaheuristic configurations	Number of matched solutions (with exact method)	Number of matched solutions (with heuristic method)	Number of better solutions
Rand removal with regret-2 (LNS-1)	8	8	1
Worst removal with regret-2 (LNS-2)	10	14	2
Shaw removal (based on distance) with regret-2 (LNS-3)	14	23	2
Shaw removal (based on distance) with deep greedy (LNS-4)	11	20	2
Shaw removal (based on distance) with basic greedy (LNS-5)	8	12	1
ALNS	19	35	4

3-4-1-2- Consideing tight capacity constriants

Similar to the first scenario, six proposed heuristics (LNS-1, LNS-2, LNS-3, LNS-4, LNS-5, LNS-6, and ALNS) are executed for 104 instances taken from benchmark instances, considering tight bus capacity constraints. The results are then compared to the best-known solutions (both proposed heuristic and MIP in Schittekat et al. (2013)). Each heuristic is run 10 times, and the average solution, best solution, and average computing time of the 10 runs are calculated. The average solution gap, the best solution gap to the best-known solutions, and computing time are reported. Detailed experiment information is available in Appendix 4. The aggregated findings are presented in Tables 3-7, 8, and 9, respectively.

Table 3-7 indicates the gap from BKS_{MH}, while Table 3-8 represents the percentage gap from BKS_{exact}.

Table (3-7) The Results Obtained from Solving the Instances Placed in S, M, And L Sets (Second Scenario)

(a). Best gap from best-known solutions (BKS_{MH})

Configuration	Removal heuristics			Insertion heuristics			Average over all instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	3.20%
LNS-2			•			•	2.26%
LNS-3	•					•	2.22%
LNS-4	•				•		2.50%
LNS-5	•			•			2.68%
ALNS	•	•	•	•	•	•	1.95%

(b). Average gap from best-known solutions (BKS_{MH})

Configuration	Removal heuristics			Insertion heuristics			Average over all instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	5.28%
LNS-2			•			•	3.73%
LNS-3	•					•	3.80%
LNS-4	•				•		4.11%
LNS-5	•			•			4.31%
ALNS	•	•	•	•	•	•	3.18%

Table 3-8 presents the results of the comparison with BKS_{exct}. In terms of the best gap criterion, LNS-3 achieves a smaller gap, with ALNS ranked as the second. Regarding the average gap criterion, ALNS achieves the smallest gap.

Table (3-8) The Results Obtained from Solving the Instances Placed in S Set (Second Scenario)

(a). Best gap from best-known solutions (BKS_{exct})

Configuration	Removal heuristics			Insertion heuristics			Average over all instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	2.85%
LNS-2			•			•	1.97%
LNS-3	•					•	1.58%
LNS-4	•				•		2.19%
LNS-5	•			•			2.53%
ALNS	•	•	•	•	•	•	1.70%

(b). Average gap from best-known solutions (BKS_{exct})

Configuration	Removal heuristics			Insertion heuristics			Average over all instances (Percent)
	Shaw (Based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	4.91%
LNS-2			•			•	3.38%
LNS-3	•					•	3.24%
LNS-4	•				•		3.96%
LNS-5	•			•			4.36%
ALNS	•	•	•	•	•	•	3.04%

Table (3-9) Total Average Computing Time in SET S, M And L in Seconds (Second Scenario)

Configuration	Removal heuristics			Insertion heuristics			Total computing time over sets.
	Shaw (based on distance)	Rand	Worst	Basic greedy	Deep greedy	Regret -2	
LNS-1		•				•	55,107
LNS-2			•			•	60,158
LNS-3	•					•	63,298
LNS-4	•				•		65,155
LNS-5	•			•			61,204
ALNS	•	•	•	•	•	•	57,796

When comparing two scenarios based on the best gap with best-known solutions (BKS_{MH}), LNS-1 and LNS-2 yield similar results. Conversely, concerning the best gap with best-known solutions (BKS_{exact}), LNS-2 and LNS-3 show comparable outcomes. On average, the proposed heuristics produce lower best gap values for the first scenario. Examining total computing time, the first scenario, utilizing a strategic oscillation, demonstrates shorter computing times. In conclusion, the first scenario emerges as the recommended algorithm for the problem.

Another analysis has been conducted to assess the percentage difference between ALNS and LNS metaheuristics, both with and without strategic oscillation. The results indicate that, with the adoption of strategic oscillation, the deviation of ALNS from the worst and best LNS metaheuristic configurations is 1.5% and 0.34%, respectively. In an alternative scenario where, tight capacity constraints are applied, the deviation of ALNS from the best and worst metaheuristic configurations is 1.25% and 0.29%, respectively. While the observed deviation may not be very pronounced, it underscores that the presence of the strategic oscillation has led to increased differentiation between ALNS and each of the LNS approaches.

3-5- Further analysis

In this stage, additional analysis is performed on a set of 24 instances, comprising 10 from the small set, 8 from the medium set, and 6 from the large set. For simplicity, the methods "Shaw removal (based on distance) with Deep greedy," "Shaw removal (based on distance) with Regret-2," "Shaw removal (based on distance) with basic greedy," "Worst removal with Regret-2," "Rand removal with Regret-2," and Adaptive Large Neighborhood Search are referred to as LNS-1, LNS-2, LNS-3, LNS-4, LNS-5, and ALNS, respectively.

In the first experiment, we analyze how different acceptance criteria impact the average solution cost for considered algorithms. We focus on three types: Random Walk (RW), Better Acceptance (BA), and Simulated Annealing (SA). The goal is to assess how well acceptance criteria align with the specific problem at hand. Results are illustrated in Figure 3-4, depicting the influence of acceptance criteria on average solution costs.

For all heuristics, results show that, on average (across 24 instances), Better Acceptance (BA) outperforms Random Walk (RW) for small and medium sets. However, for large instances, RW proves more efficient than BA. In summary, focusing solely on extreme diversification (RW) or strong intensification (BA) is not universally effective. Simulated Annealing serves as a middle-ground option, although not necessarily the best. On average, for small, medium, and large instances, Simulated Annealing (SA) with a 2.55% Best Gap outperforms BA (2.74% Best Gap) and RW (2.70% Best Gap) across all heuristics. Additionally, LNS-5 shows dissatisfactory performance for all sets.

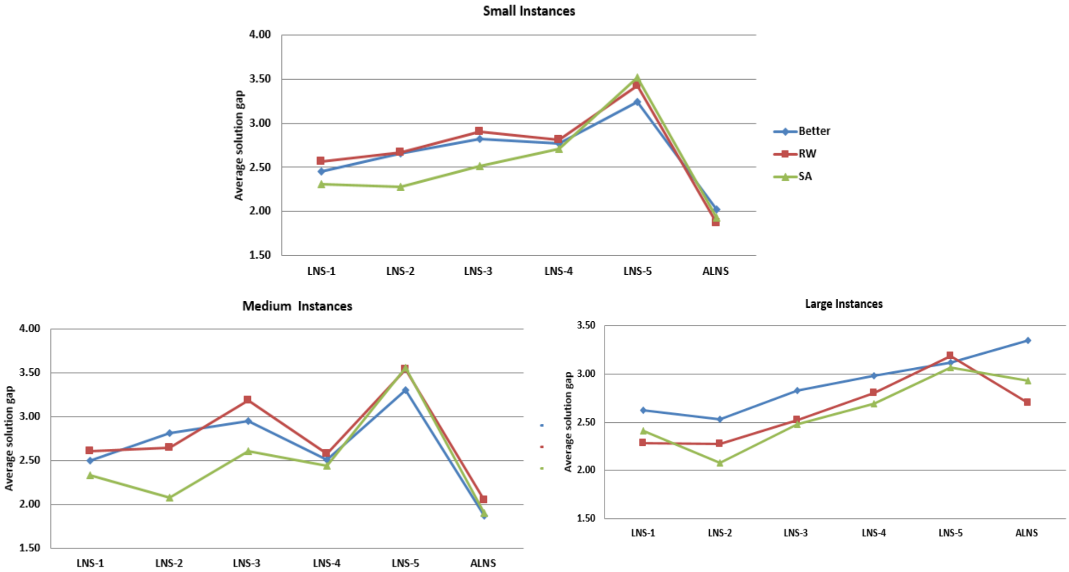


Figure (3-4) Influence of Acceptance Criteria Method on The Average Solution Gap

The second experiment exclusively focuses on the ALNS configuration. It examines the impact of each heuristic embedded in the ALNS construction to enhance solution quality. The weights assigned to heuristics are dynamically calculated based on their recorded performance, with higher weights given to better-performing heuristics. This dynamic weighting system influences the probability of selecting a heuristic during the search, thereby impacting solution results.

Results, based on the average weight of each heuristic across 24 data cases, are depicted in Figure 3-5. Notably, regret-2 heuristic attains the highest weight among insertion heuristics, while Shaw removal (distance-based) receives the highest weight among removal heuristics. This suggests that the contribution of Shaw removal heuristic (distance-based) surpasses that of other heuristics throughout the search process.

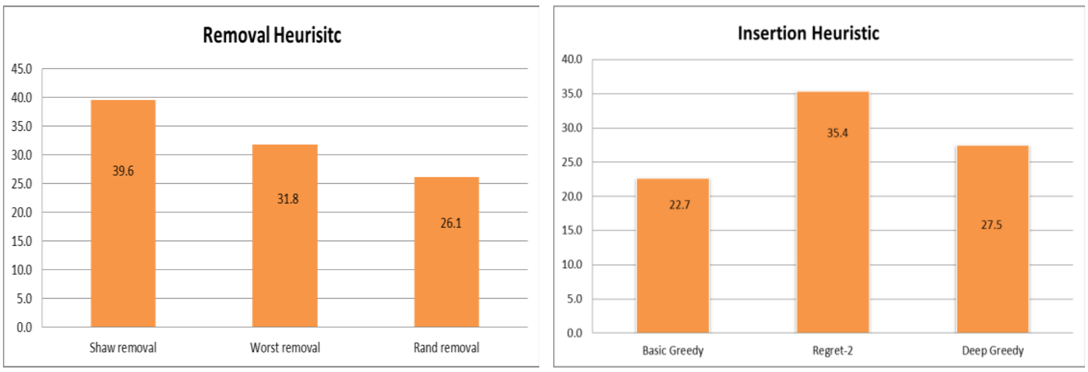


Figure (3-5) Weights of the Heuristics

3-6- Conclusion

In this chapter, various LNS and ALNS heuristics based on an oscillation strategy are introduced to address the SBRP, a problem presented by Schittekat et al. (2013) as a challenging variant of the VRP. The chapter concentrates on the SBRP, integrating bus stop selection and route generation into a unified optimization approach. Specifically, students are assigned to potential bus stops based on walking distance from their homes. Simultaneously, these stops are incorporated into the bus route to efficiently transport students to school while minimizing travel distances.

Simple LNS and ALNS metaheuristics are proposed to address small, medium, and large instances within a reasonable time frame. Each metaheuristic undergoes statistical analysis to determine the optimal heuristic parameter settings.

After determining the optimal parameter settings for each solution, a comparison between both metaheuristics in different configurations and the best-known solutions is conducted. This evaluation considers solution quality, robustness, and computing time across all instances. To provide a more nuanced understanding of the proposed algorithm, two scenarios are formulated, facilitating a clearer comprehension of algorithm behavior with and without the oscillation strategy.

In terms of the percentage of the best gap to the best-known solutions (BKS_{MH}), LNS-1 and LNS-2 exhibit nearly comparable results in both scenarios. However, other heuristics perform poorly in the second scenario. Regarding computing time, heuristics in the first scenario achieve results in less time. In summary, the first scenario is deemed more reliable for delivering superior performance in solution quality and computing time.

In the first scenario, computational experiments indicate that the ALNS proves highly competitive compared to the best metaheuristic (BKS_{MH}).introduced by Schittekat et al. (2013). This competitiveness stems from the ALNS algorithm's structured exploration of large portions of the solution space, making it robust and adaptable to various cases while avoiding frequent entrapment in local optima. Across 10 runs, the ALNS metaheuristic analysis yields results ranging from a 1.64% Best Gap to a 2.77% Average Gap concerning the existing best-known solutions (BKS_{MH}). The computational results reveal that in terms of finding optimal solutions, the ALNS outperforms other configurations. It matches the best-known solutions (BKS_{MH}), in 35 instances, and surpasses them in 4 cases. The Shaw removal with Regret-2 ranks second, matching the (BKS_{MH}), in 23 instances. Comparing with (BKS_{exact}) in the first scenario, the ALNS demonstrates superior performance, matching the best-known solutions in 19 instances. This comparison highlights that the ALNS provides a lower percentage gap in the first scenario, with LNS-3 securing the second position.

The primary contribution of this research is the recognition of ALNS as the best metaheuristic among all proposed solution algorithms. On average, the first scenario exhibits a lower percentage gap concerning both (BKS_{MH}), and (BKS_{exact}). In terms of computing time, ALNS and Rand removal with Regret-2 outperform other algorithms.

In proposing future research, two suggested topics are presented. Regarding the features, the current study reveals insufficient emphasis on realistic characteristics. To enhance proximity to reality in our upcoming work, we aim to integrate features such as the effects of mixed-loads (picking up students from different schools on the same bus), constraints on the maximum route length, consideration of multiple schools, and determination of the maximum number of allowed students for each bus stop.

The second topic involves enhancing specific aspects of the metaheuristic to improve its performance and reduce computing time. This can be approached through multiple branches. One extension to boost the oscillation strategy method's performance involves constructing a memory list to retain infeasible solutions that lead to global best solutions. These solutions can then be considered to return to a feasible state. Another interesting idea is to introduce variable values for α_{min} and α_{max} during the search space, allowing these boundaries to adapt based on solution performance. Moreover, a promising avenue for research is to incorporate simple yet effective problem-specific heuristics in the improvement phase of the algorithm.

Chapter 4:

An Adaptive large neighborhood search metaheuristic for the school bus routing problem with mixed-load consideration

4-1- Introduction

Tehran, the capital of Iran, ranked among the world's top 40 cities in 2018, boasting a population of approximately 8.9 million[1]. More than 1 million students in Tehran, constituting 10.8% of the total students in Iran, rely on daily public transport system.

Tehran's substantial student population poses a significant challenge for educational authorities in managing school transportation services. Yet, specific factors in Tehran's student transportation—such as safety concerns, security restrictions, multiple routes per bus, and inconsistent loading and unloading times—create complex school bus routing issues. Addressing these challenges demands extra efforts to plan and organize cost-effective transport services.

Authorities strive to establish an efficient transport system, mindful of limited resources to save costs. Simultaneously, ensuring student convenience is crucial in transportation plan design. In situations where dedicating a single bus to one school is impractical due to resource constraints, a "mixed-load approach" becomes beneficial (see e.g., Park and Kim, (2010)).

This approach, different from the "single-load approach," enables the transportation of students from different schools on the same bus, fostering resource-sharing among schools and enhancing school-bus system efficiency. However, this strategy introduces complexities, leading to overcrowded buses and extended routes. Designing bus routes based on a mixed-load plan could prove effective, requiring careful consideration of objectives, assumptions, and constraints.

This study is pioneering in simultaneously addressing bus stop selection and route generation while accounting for both mixed-loading and load balancing effects. Specifically, the chapter builds upon the groundwork laid by Schittekat et al. (2013) and endeavors to adapt their findings to real-world features, incorporating factors like mixed-load effects and school time windows.

Each student is required to walk to the bus stop from their home, considering the maximum walking distance. The bus then initiates its route from the garage, picking up students at designated bus stops while adhering to specified constraints, such as bus capacity. The plan prioritizes student convenience, factoring in considerations like the student's maximum walking distance to the bus stop and the permissible number of students at each stop.

Our study makes several key contributions:

1) Formulating a novel mathematical representation of the School Bus Routing Problem (SBRP) that incorporates the defined objective and assumptions.

[1] https://worldpopulationreview.com/world-cities/tehran-population

2) Proposing an Adaptive Large Neighborhood Search (ALNS) metaheuristic distinct from the one presented in Chapter 3, and subsequently comparing its performance with existing benchmarks.
3) Conducting a pair comparison of each removal and insertion operator, instead of an independent comparison.
4) Assessing the impact of both mixed-load and single-load approaches on reducing the number of buses, total traveled distance, average weighted riding distance of students, and bus occupancy.

4-2- Problem description and mathematical model

In this research, we focus on multiple schools, one type of students, potential bus stops, a set of garages, and identical buses, each with the same capacity. The study arises from the necessity to formulate a daily transportation plan for transporting students from their homes to their respective schools.

The process involves assigning each student to an approved bus stop, considering a defined walking distance. Subsequently, each bus commences its route from the garage (starting location), collects students from designated bus stops, transports them to school, and ultimately returns to the garage (ending location). Each student has to be delivered to his/her respective school.

As the problem selects bus stops and generates routes with consideration for the mixed-load effect, students from different schools can be assigned to the same bus. The starting and ending locations of a bus are not required to be the same, potentially avoiding long return trips to the same garage. However, to prevent bus crowding at a same garage, the model incorporates an allowable number of parking spaces for each garage. To align with real-world scenarios, our model introduces two-time constraints: each bus must reach its associated school before a defined latest arrival time, and each stop cannot be visited by the bus before a specified earliest time.

Let G, P^{+} and P^{-} respectively define starting and ending locations of a bus, potential bus stops, and potential schools. P is the union of potential schools and bus stops ($P = P^{-} \cup P^{+}$), and N is the set of all nodes ($N = P \cup G$).

The travel time from node i to node j is computed as the distance between the two nodes divided by the speed of the bus. For simplicity, all buses are assumed to move at the same speed.

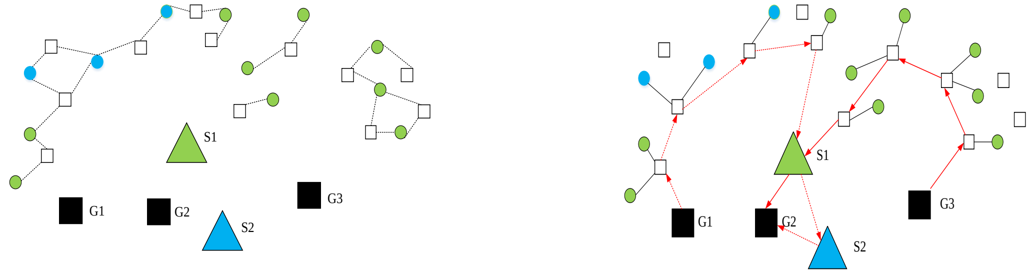
The objective function aims to minimize the total travel time of all buses. In this problem, school data includes the location of each school and only the latest possible times for bus arrival. Specifically, the study seeks to pick up primary and secondary school students, allowing each school to have a different time window.

The most important constraints in our problem include:

1) Every student must walk from their home to one of the potential bus stops within a specified maximum walking distance.
2) Each bus starts from a garage and concludes its route at the garage closest to the last school it visited.
3) The maximum allowable number of students for each bus stop should not exceed the limit, denoted as ms.
4) The number of buses returning to a garage must not surpass the available parking spaces, denoted as P_g.
5) Each bus should arrive at its designated school (denoted as $i \in P^-$) before the latest time b_i, establishing an upper bound on the time for delivering students to their respective schools.
6) The service time of each stop must occur after the earliest time a_i, and if the bus arrives at the i^{th} stop before a_i, it must wait.
7) The load of each bus along its path should not exceed its given capacity.

Constraints (1) and (3) prioritize student convenience in our model. Table 4-1 details the symbols used in the model, and Figure 4-1(a) provides an illustrative example of the problem. Students are represented by circles, potential bus stops by small squares, garages by large black squares, and schools by triangles. The color-coding matches students with their respective schools. Dotted lines indicate the reachable stops for each student.

Figure 4-1(b) illustrates a feasible (though not necessarily optimal) solution with two routes denoted by red lines. Each route starts from a garage, picks up students from various stops, transports them to the associated school, and then returns to the garage. The problem incorporates the following assumptions: (1) buses may carry students from different schools simultaneously; (2) each student must be picked up before being delivered to their respective school; (3) each school can be visited by more than one bus, but each bus must visit each school once.



(a) Instance with possible allocation of students to the potential bus stops

(b) Possible feasible solution

Figure (4-1) Example School Bus Routing Problem with Mixed Load Effect

Table (4-1) Indices, Sets, Parameters, And Decision Variables Used in Mathematical Model

Indices	
k	Bus index
i, j	Node indices
l	Student index
Sets	
G	Set of starting and ending depot locations (garage locations)
K	Set of buses
S	Set of students
P^+	Set of potential pickup locations (bus stop locations)
P^-	Set of delivery locations (school locations)
$P = P^- \cup P^+$	Set of stops and schools
$N = P \cup G$	Set of nodes
Parameters	
c	Bus capacity
$big\ M$	Large constant
a_i	Earliest arrival times to stop $i \in P^+$
b_i	Latest arrival times to school $i \in P^-$
ap	Average pickup time at pickup points for each student
ad	Average delivery time at delivery points for each student
C_{ij}	Travel distance from node i to node j $(i, j \in N)$
t_{ij}	Travel time from node i to j $(i, j \in N)$
s_{il}	A parameter equal to 1 if student l can reach stop $i \in P^+$, and 0 otherwise
q_{il}	A parameter equal to 1 if student l is related to the school $i \in P^-$, and 0 otherwise
P_g	The number of parking spaces at the garage g
ms	The maximum number of allowable students for each stop
$O_i = \{S \mid s_{il} = 1\}$	The set of students that can be assigned to stop i
$W_i = \{S \mid q_{il} = 1\}$	The set of students that should be delivered to school i
Decision variables	
X_{ijk}	1 if bus k traverses the arc from node $i$to j $(\forall i, j \in N)$, and 0 otherwise
y_{ik}	1 if the bus k visits stop i, 0 otherwise
z_{il}^k	1 if student l is picked up by bus k from stop i, and 0 otherwise
T_{ik}	Arrival time of bus k to node i $(\forall i \in N)$
L_{ik}	The load of bus $k$after leaving node i $(\forall i \in P)$
h_{ik}	1 If bus $k$visits school $i \in P^-$, and 0 otherwise
D_{jl}^k	1 if student l is delivered by bus $k$to school j, and 0 otherwise

The mathematical programming formulation of the school bus routing problem is as follows:

$$Min \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} t_{ij} X_{ijk} \qquad (4\text{-}1)$$

S.t.

$$\sum_{j \in N} X_{jik} = \sum_{j \in (P^+ \cup P^-)} X_{ijk} = y_{ik} \qquad \forall i \in P^+, k \in K \qquad (4\text{-}2)$$

$$\sum_{j \in (P^+ \cup P^-)} X_{jik} = \sum_{j \in N} X_{ijk} = h_{ik} \qquad \forall i \in P^-, k \in K \qquad (4\text{-}3)$$

$$\sum_{i \in G} \sum_{j \in P^+} X_{ijk} \le 1 \qquad \forall k \in K \qquad (4\text{-}4)$$

$$\sum_{j \in P^-} \sum_{i \in G} X_{jik} \le 1 \qquad \forall k \in K \qquad (4\text{-}5)$$

$$\sum_{i \in G} \sum_{j \in G} X_{ijk} = 0 \qquad \forall k \in K \qquad (4\text{-}6)$$

$$\sum_{k \in K} y_{ik} \le 1 \qquad \forall i \in P^+ \qquad (4\text{-}7)$$

$$\sum_{k \in K} Z_{il}^k \le s_{il} \qquad \forall l \in S, i \in P^+ \qquad (4\text{-}8)$$

$$Z_{il}^k \le y_{ik} \qquad \forall l \in O_i, i \in P^+, k \in K \qquad (4\text{-}9)$$

$$y_{ik} \le \sum_{l \in S} Z_{il}^k \qquad \forall i \in P^+, k \in K \qquad (4\text{-}10)$$

$$\sum_k D_{jl}^k \le q_{jl} \qquad \forall l \in S, j \in P^- \qquad (4\text{-}11)$$

$$D_{jl}^k \le h_j^k \qquad \forall l \in W_j, j \in P^-, k \in K \qquad (4\text{-}12)$$

$$h_{jk} \le \sum_{l \in S} D_{jl}^k \qquad \forall j \in P^-, k \in K \qquad (4\text{-}13)$$

$$\sum_{i \in P^+} Z_{il}^k = \sum_{j \in P^-} D_{jl}^k \qquad \forall l \in S, k \in K \qquad (4\text{-}14)$$

$$\sum_{i \in P^+} \sum_{k \in K} Z_{il}^k = 1 \qquad \forall l \in S \qquad (4\text{-}15)$$

$$\sum_{l \in S} Z_{il}^k \le ms \qquad \forall i \in P^+, k \in K \qquad (4\text{-}16)$$

$$L_{ik} = 0 \qquad \forall i \in G, k \in k \qquad (4\text{-}17\text{-}a)$$

$$L_{ik} + \sum_{l \in S} Z_{jl}^k \le L_{jk} + bigM(1 - X_{ijk}) \qquad \forall i \in P, j \in P^+, k \in k \qquad (4\text{-}17\text{-}b)$$

$$L_{ik} - \sum_{l \in S} D_{jl}^k \le L_{jk} + bigM(1 - X_{ijk}) \qquad \forall i \in P, j \in P^-, k \in k \qquad (4\text{-}17\text{-}c)$$

$$\sum_{l \in S} Z_{il}^k \le L_{ik} \le C \qquad \forall i \in P^+, k \in k \qquad (4\text{-}17\text{-}d)$$

$$T_{ik} + ap.\sum_{l \in S} Z_{il}^{k} + ad.\sum_{l \in S} D_{il}^{k} + t_{ij} \le T_{jk} + bigM\,(1 - X_{ijk}) \qquad \forall i \in P, j \in P, k \in K, i \ne j \qquad (4\text{-}18)$$

$$T_{ik} + t_{ij} \le T_{jk} + bigM\,(1 - X_{ijk}) \qquad \forall i \in G, j \in P^{+}, k \in K \qquad (4\text{-}19)$$

$$T_{ik} \le T_{jk} + bigM\,(1 - Z_{il}^{k}) \qquad \forall i \in P^{+}, j \in P^{-}, l \in S, k \in K \qquad (4\text{-}20)$$

$$T_{ik} \ge a_i - (1 - y_{ik})bigM \qquad \forall i \in P^{+}, k \in K \qquad (4\text{-}21)$$

$$T_{ik} \le b_i + (1 - h_{ik})bigM \qquad \forall i \in P^{-}, k \in K \qquad (4\text{-}22)$$

$$\sum_{i \in P^{-}} \sum_{k \in K} X_{ijk} \le Pg \qquad \forall j \in G \qquad (4\text{-}23)$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in P^{+}, k \in K \qquad (4\text{-}24)$$

$$X_{ijk} \in \{0,1\} \qquad \forall i, j \in N, i \ne j, k \in K \qquad (4\text{-}25)$$

$$Z_{il}^{k} \in \{0,1\} \qquad \forall i \in P^{+}, l \in S, k \in K \qquad (4\text{-}26)$$

$$D_{jl}^{k} \in \{0,1\} \qquad \forall j \in P^{-}, l \in S, k \in K \qquad (4\text{-}27)$$

$$h_{ik} \in \{0,1\} \qquad \forall i \in P^{-}, k \in K \qquad (4\text{-}28)$$

$$r_{lk} \in \{0,1\} \qquad \forall l \in S, k \in K \qquad (4\text{-}29)$$

The objective function (4-1) aims to minimize the total travel time of all buses. Constraints (4-2) ensure that a bus entering a stop node must also leave it, and the equivalent constraints for school nodes are expressed in Equation (4-3). Constraints (4-4) state that a bus cannot start from its home location (garage) more than once. Similarly, constraints (4-5) ensure that a bus cannot reach its final location (garage) more than once, allowing for the possibility that some buses may remain unused. Constraints (4-6) prohibit direct transfers from one garage to another garage. Lastly, constraints (4-7) dictate that each stop should be visited no more than once.

Constraints (4-8) mandate that each student must be picked up from the stop they walk to. Constraints (4-9) specify that picking up a student from a non-visited stop by bus k is not allowed. Constraints (4-10) ensure that stops are not visited unnecessarily. Constraints (4-11) guarantee that each student is delivered to their respective school, and constraints (4-12) ensure that whenever a student is assigned to a bus, the bus also visits the school associated with that student. Constraints (4-13) prohibit unnecessary visits to schools.

Constraints (4-14) state that the number of pickup and delivery students in each route must be equal. Constraints (4-15) ensure that each student is picked up exactly once. Constraints (4-16) impose that the number of allocated students to each allowable stop must not exceed m_s.

The next four sets of constraints, (4-17-a), (4-17-b), (4-17-c), and (4-17-d), pertain to load constraints. Specifically, (4-17-b) states that when a node i is followed by a pickup node $j \in P^{+}$, the total number of students after visiting node j should be greater than or equal to the sum of the number of students after servicing node i and the number of students picked up at node j.

Similar to constraint (4-17-b), inequality (4-17-c) asserts that when node i is followed by a delivery node, $j \in P^-$, the total number of students after visiting node j should be greater than or equal to the number of students after visiting node i minus the number of students delivered to node j. In practical terms, (4-17-b) and (4-17-c) govern the load on a bus immediately after leaving each node on its route. Additionally, (4-17-d) ensures the capacity of buses.

Constraints (4-18) to (4-22) focus on time-related considerations. (4-18) calculates the arrival time of each bus at a node in p. Similar to (4-18), (4-19) pertains to routes from the garage to the stop. (4-20) ensures that student pickups occur before deliveries. Time windows for stops and schools are determined by (4-21) and (4-22) respectively. (4-23) restricts parking places in each garage. Finally, variable and their types are presented in (4-24) to (4-29).

4-3- Solution strategy

SBRP is a generalization of the Vehicle Routing Problem (VRP), a known NP-hard problem. While exact methods from the literature can provide optimal solutions, they are practical only for scenarios with a relatively small number of stops or students. This limitation falls short of real-world cases involving hundreds of stops or students.

To address this, heuristic approaches become essential to handle large instances and obtain near-optimal solutions within a reasonable timeframe. Various heuristic variants, grounded in local search contexts, have been applied to the VRP. Local search operators create regular moves that slightly alter the current solution. These moves could change the requests within one or two different routes simultaneously.

This type of operator enables rapid exploration of a large parts of the solution space, causing minor changes with each iteration. While this approach offers rapid exploration, it comes with limitations. For instance, imposing tight constraints on the problem and employing local search operators may not yield significant benefits (Ropke and Pisinger, 2006). Consequently, smoothly transitioning from one promising area to another becomes challenging.

To address this issue, alternative strategies can be employed, such as opting for larger standard moves instead of incorporating smaller ones. Using this approach is time-consuming but yields better results than the standard move strategy. Instead of making small changes, it's more effective to implement large moves for increased exploration in the solution space. Thus, there's a case for favoring extensive exploration through large neighborhood search (LNS) (refer to Chapter 3 for details).

Algorithm 4-1 depicts the ALNS algorithm's mechanism. This metaheuristic has two stages: the construction stage and the improvement stage. In the first stage, a student allocation problem is solved for each stop. Following the implementation of the student allocation heuristic, a modified version of the nearest neighborhood constructive heuristic generates a feasible initial solution. This initial solution serves as input for the second stage, i.e., the improvement stage (refer to Line 7 in the pseudo-code).

The improvement stage has two levels executed sequentially over multiple iterations. Specifically, the algorithm enhances the solution using adaptive large neighborhood heuristics at the primary level (refer to Line 8 in the pseudo-code). In each iteration, the removal heuristic disconnects a set of q stops, placing them in the stop pool named the U bank list. Then, using the insertion heuristic, it inserts the stops from U bank into the solution. It's important to note that during removal and insertion operations, students from different schools might be inserted in the same route. In such cases, while maintaining feasibility, the school associated with the student also needs to be inserted at the most cost-effective position within the current route.

The value of q is a critical parameter defining the scope of our solution approach, essentially indicating the neighborhood size. If q is zero, no search occurs in the solution space. Conversely, when q equals the cardinality of P^+, the algorithm functions like a multi-start, solving the problem from scratch. Additionally, this value can depend on the solution's behavior at each iteration. To strike a balance between diversification and intensification mechanisms, the following procedure updates the value of q.

Initially, q is set to q_{min} and systematically adjusted during the adaptive large neighborhood search algorithm. Specifically, q 's value changes based on the solution from the previous iteration. For instance, if an acceptable solution is consistently obtained over several iterations, q should stay at the low level, q_{min}, to emphasize intensification. Conversely, if a worse solution appears for several iterations, q must be increased to explore the solution space more effectively. Consequently, a large number of q are removed and then re-inserted, aiding in achieving an appropriate diversification strategy during the search.

To select removal and insertion heuristics, a roulette wheel mechanism is employed, considering their past successful behavior. Additionally, the Meta-destroy operator is implemented when no improvement occurs in the best solution for δ consecutive iterations. This operator executes two destroy operators sequentially, enhancing diversification. Importantly, this procedure operates independently of performing removal and insertion steps in each segment. Specifically, in each segment, the count of successive non-improving solutions is tracked from the beginning of the improvement stage. If this count exceeds δ (where δ is less than the number of iterations in each segment), the Meta-destroy operator is activated.

After applying the removal and insertion operators, the Redistribution operator is invoked at the second level whenever a new best solution is found (refer to Line 15 in the pseudo-code). This is because the combined application of removal and insertion heuristics reconstructs a substantial part of the solution, leading to a dispersed distribution of students between routes. To address this, the redistribution operator optimizes the current load distribution by transferring students between routes while maintaining feasibility. In practice, this operator minimizes the deviation of the loading values of the routes by achieving a desirable distribution of students among them.

The ALNS algorithm demonstrates excellent performance for large-scale optimization problems, particularly demonstrating success in vehicle routing problems. The positive outcomes in VRP applications have motivated this study to apply ALNS to the school bus routing problem with a mixed-load plan.

The ALNS algorithm outlined in this chapter differs from the one discussed in Chapter 3 in several key aspects, including:

- Introducing problem-specific removal and insertion heuristics;
- Employing a pairwise selection of removal and insertion mechanisms to ensure comprehensive metaheuristic performance;
- Implementing a time-saving strategy, namely local and global insertions, in the improvement phase;
- Introducing a meta-destroy operator if no improvement occurs in the best solution after a delta consecutive iterations;
- Executing distribution operators, post-application of removal and insertion operators, to balance the current occupied capacity among the routes;
- Constructing the initial solution based on the nearest neighborhood with a greedy randomized selection mechanism;
- Operating the algorithm under tight capacity constraints;

Algorithm (4-1) Adaptive Large Neighborhood Search Metaheuristic

Input: U: set of all potential stops, G: set of all garages, P^{-} :set of all schools, S: set of all students
R (set of Removal heuristics), I (set of Insertion heuristics), q (number of stops/ requests to be removed $q \in \{1,...n\}$), q_{max} (maximum number of stops to be removed), P^{+}(List of stops to which students are allocated), π(initial score of heuristic ($I \cup R$)), w (initial weight of removal and insertion heuristic ($I \cup R$)),
ρ (The number of iterations), η (parameter to set q_{min}), δ (no of consecutive iterations without improvement)

// Stage 1: Construction phase

P^{+}=all student allocated to the bus stop // Allocating using student allocation heuristic

x_o =Route generation (P^{+}, s, G, P^{-}) // Generating route using NNg heuristic

$x_{best} = x_o$

$f_{best} = f(x_0)$

$x_{act} = x_o$

// Stage 2: Improvement phase

/// 2.1 Set of removal and insertion heuristics in the first level

$q = q_{min}$, initialize the roulette wheel; initialize the adaptive parameters (π, w)

While *S*topping criterion ρ is not met do

Roulette wheel mechanism: Select one Removal heuristic $h_{rem} \in R$ and one Insertion heuristic $h_{ins} \in I$or two Destroy operators (if x_{best} has not been improved in last consecutive δ iterations)

Remove q requests from solution x_{act} using h_{rem}, creating a partial solution

Insert q customers into the partial solution using h_{ins}, creating a solution x^{*}_{act}

If accept (x_{act}, x^{*}_{act}) then

/// 2.2 Redistribution operator in the second level

$x^{**}_{act} = Redistribution(x^{*}_{act})$ // Applying Redistribution heuristic to x^{*}_{act}

$x_{act} = x^{**}_{act}$

$q = q_{\min}$

Else

If $q < q_{max}$ then

$q = q + 1\%$

Else

$q = (\frac{q_{\max}}{\eta})$

End if

End if

If $f(x^{**}_{act}) < f_{best}$ then

$x_{best} = x^{**}_{act}$

End if

Update the roulette wheel (π, w)

End while

Output: x_{best}

4-3-1- Constructing an initial solution

The construction stage's main concept is to create a feasible initial solution for the SBRP. This involves three steps: assigning each student to an allowable stop, grouping allowable stops based on the closest garage, and generating routes for allowable stops of each garage. The process unfolds in three sequential stages.

First, the student allocation heuristic assigns each student to an allowable stop. After allocating all existing students to the possible bus stops, the potential stops with their respective students are identified (for more detail see Chapter 1). In the second step, each identified stop is assigned to the closest garage, determining the distribution of each stop to a given garage in advance.

Next, the modified nearest neighborhood with the greedy randomized adaptive procedure (NNgr) is utilized to generate a route, introducing a balanced approach between greediness and randomness. Unlike a simple greedy nearest neighborhood heuristic, our version incorporates modifications in constructing routes in two ways.

Firstly, for each route (i.e., bus) originating from a given garage, the next node (i.e., stop) is randomly selected from the restricted candidate list (RCL). This RCL includes α first closest non-visited stops assigned to that garage. The size of the RCL, denoted as the value of α, is a parameter controlling the balance between greediness and randomness. A small α leads to an extremely greedy construction, while a large α (equivalent to the number of non-visited stops in the solution) results in a completely random construction.

Secondly, a feasibility check is conducted considering both the allowable bus capacity and school time window constraints. If a feasible solution is generated without violating these constraints, the candidate stop is added to the route. Conversely, if constraints are breached, the generated route returns to the associated school to deliver students and, ultimately, returns to the closest garage. This return to the closest garage prevents long trips that might occur if the bus returned to the garage it started from. If the capacity of the nearest garage is already filled, the next closest garage is selected. Here, the question arises: Why does the bus move to the closest and next closest garage at the end of its operation? Consider these reasons: 1) Reducing total travel time for the bus in student journeys. 2) Making the bus available for other services, enhancing efficiency. 3) Contributing to CO2 emission reduction[1], crucial in Tehran's critical air pollution scenario. These pieces of evidence show that, in practice, this action provides several advantages.

https://ifpnews.com/tehran-critical-air-pollution-crisis [1]

The two approaches are iteratively applied until all non-visited stops are considered for each garage. As each bus can pick up students from different schools, all associated schools must be inserted at the end of the considered route in the cheapest way possible. To achieve efficient time savings, a data structure is employed and updated throughout the operation of the NNgr heuristic. This data structure includes information related to the load and travel time of a bus traversing along route k.

The example in Figure 4-2 illustrates the functionality of the data structure. Throughout the operation of NNgr for each stop, the following information is updated: visited stop (S), number of students allocated to the stop (AS), load of bus after visiting the current stop (LS), load of bus after visiting the next stop (LN), remaining capacity after visiting the current stop (R), remaining capacity after visiting the next stop (RC), arrival time at the current stop (AT), and allowable remaining time to reach the respective school (RT).

This straightforward data structure procedure enables efficient checking of both capacity and school time window constraints before selecting any stop. In this example, assuming the bus capacity is 6, the solid line represents the generated route. Suppose stop B is a candidate for insertion in the route. As indicated in Table 4-2, the values of LA and LB are 5 and 7, respectively. This suggests that inserting stop B in the route would lead to an infeasible solution. Consequently, instead of visiting stop B, the bus must return to the respective schools.

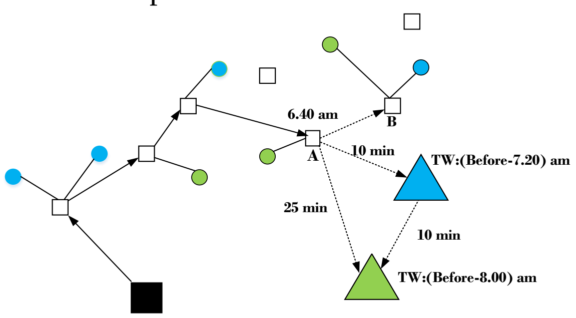


Figure (4-2) Example for Constructing an Initial Solution

Table (4-2) Data Structure

Stop(S)	AS	LS	LN	R	RC	AT	(RT) Travel time to school 1	(RT) Travel time to school 2
A	1	5	7	1	-1	6.40	10 min	25 min

4-3-2- Adaptive large neigborhood search (ALNS)

In essence, the ALNS heuristic is an iterative process comprising destruction and insertion operators. During each iteration, a removal heuristic is used to eliminate a set number of stops (i.e., requests) from the current solution. Subsequently, an insertion heuristic is applied to reinsert these stops into the current solution, constructing a new solution (for detailed information, refer to Ropke and Pisinger, 2006). Removal heuristics are detailed in Section 4.3.2.1, and insertion heuristics are addressed in Section 4.3.2.2.

4-3-2-1- Removal operators

The removal operator serves as the algorithm's foundation, removing q stops at each iteration and adding them to the list U. These operators must be chosen to efficiently explore the entire search space or, at the very least, its interesting segments. It is crucial to consider both diversification and intensification operators in a structured manner, rather than solely focusing on specific types of destroy operators. This study incorporates various removal heuristics, enabling both diversification and intensification strategies. Three removal heuristics draw inspiration from Ropke, while others are novel and adapted based on the considerations of the problem (SBRP).

- **Shaw removal**

The basic concept of Shaw removal is presented here similarly to Chapter 3, with the only difference being related to the mechanism for selecting one request from the list U.

When one request is randomly chosen from the list U (if it is not empty), the following procedure is applied to the selected candidate stop. Instead of choosing a stop randomly, as proposed by Shaw et al. (2006), we opt for the stop that offers a higher likelihood of cost savings. To achieve this, one route is randomly selected in advance. Subsequently, for each stop, i, included in the considered route, the saving value is calculated as follows:

$s(i) = dist(prev(i), i) + dist(i, next(i)) - dist(next(i) - prev(i))$, where $prev(i)$ and $next(i)$ are respectively the predecessor and successor of a node i in the considered route.

Nodes candidates are sorted in decreasing order. This prioritizes requests with greater saving potential to enhance the solution quickly if inserted in another position. Subsequently, the stop with the maximum saving is chosen and transferred to list U.

In the second step, the degree of similarity between the selected stop in list U and the other $q-1$ stops not removed from the current solution is calculated. The Shaw removal process, detailed in Algorithm 4-2, introduces a random parameter y between $[0, 1]$, with p defining the degree of randomness to the selected request. Lower p values compel the heuristic to choose more similar stops, while higher values allow selecting less similar requests.

Algorithm (4-2) Shaw Removal (Inspired by Ropke)

Function Shaw removal $\{x_{act} \in solution, p \in R^{+}, q \in P^{+}\}$

Request: r = selected request from x_{act} using the saving method;

Set of Request: $U = \{r\}$;

While $U \leq q$ **do**

r = selected request from U;

Array: L = an array containing all request from X_{act} not in U;

Sort L such that $i \leq j \rightarrow R(r, L[i]) < R(r, L[j])$;

Choose a random number y from the interval [0,1);

$U = U \cup \{L([y^{p} |L|)\}$;

End while

Remove the requests in U from X_{act};

◻ Shaw removal based on similarity of school of the student

This operator incorporates concepts from the Shaw removal heuristic, differing in its consideration of similarity based on schools rather than a relatedness measurement $R(i, j)$ between two stops determined by distance. Formally, the heuristic aims to remove a set of stops that are similar in terms of the variety of schools with respect to their assigned students. This approach is expected to facilitate a rational reshuffling of these stops, thereby avoiding the unnecessary inclusion of a school.

In practice, the degree of similarity indicates how much two stops are alike based on the schools of their students. Similar to Shaw removal, the procedure initially selects a random stop to remove. In subsequent iterations, it selects stops that are similar to the already removed requests based on the schools of their students. The degree of similarity between the two stops is given by:

$$d(S_1, S_2) = \sum_{k} |y_{s1}^{k} - y_{s2}^{k}| \quad (4\text{-}30)$$

Let y_{s1}^{k} determines whether stop s_1 has any student of the school k. Therefore y_{s1}^{k} takes value of 1 if stop $s1$ has a student of school k, and 0 otherwise. Using y_{s}^{k}, we can easily obtain the degree of similarity between two requests. The lower $d(S_1, S_2)$ indicate that two stops ($s1$ and $s2$) are more related. This procedure proceeds until q stops are selected and transferred to the U bank.

◻ Worst removal

The Worst removal heuristic presented here is similar to Chapter 3 (refer to Chapter 3.4.2 for more details).

Algorithm (4-3) Worst Removal (Inspired by Ropke)

Function Worst Removal $\{x_{act} \in solution S, p \in R^{+}, q \in P^{+}\}$;

While $q > 0$ **do**

Array: L = All planned requests i, sorted by descending $cost(i, x_{act}) = \Delta f_i$;

Choose a random number y from the interval [0,1);

Request: $r = L([y^p |L|)$

remove r from solution S;

$q = q - 1$;

End while

- **Random removal**

The Random removal heuristic presented here is similar to Chapter 3 (refer to Chapter 3.2.2.1 for more details).

The Random removal operator fosters diversification by randomly selecting and inserting stops into the list U. This introduces a degree of randomization in the solution space, aiding in overcoming local optima.

- **Least load bus removal (LUB)**

The Least load bus removal operator focuses on removing the bus or route with the smallest load, where the load is defined as the number of picked-up students in the route. Essentially, it selects the route with the least occupied capacity and removes all stops contained in that route, aiming to completely destroy the route.

- **Single load route removal (SLR)**

The Single-load route removal heuristic randomly picks a route that exclusively contains students from a single school (i.e., a single-load route) and endeavors to remove the q stops included in the route. This heuristic follows a strategy similar to the least used bus removal. In both the Single-load route and the Least used bus removal heuristics, if the number of requests i in the candidate route k is less than q, another route is selected. It's important to note that this operator aims to decrease the number of single-load routes.

4-3-2-2- Insertion operators

These heuristics aim to construct a partially destroyed solution by inserting requests from the list U into the existing route when possible. Two critical considerations must guide the insertion procedure. Firstly, the request should be inserted at any feasible position, ensuring adherence to capacity constraints and school time windows during the insertion process. Secondly, during the insertion for the stop in the candidate route, the heuristic must check whether the related school is already part of the new route or not.

If the related school is not already in the route, it needs to be inserted in the considered route at the cheapest possible position. Importantly, during the insertion operation, when the candidate stop to be inserted has students from other schools, the cost of insertion is the summation of the cost created by inserting that stop and the respective school(s).

Although these procedures increase the time complexity of the algorithm, they promote diversification. To achieve this, our insertion heuristic follows two strategies: local and global insertions. In the local strategy, a stop from the list U is only allowed to be inserted into routes where its related school is located. Conversely, in the global insertion strategy, regardless of the existence or absence of a related school, unplanned stops can be inserted at any optimal position in the existing route.

In this study, basic greedy, regret 2, and regret 3 operators are employed based on the global insertion method. Basic greedy based on the largest demand and the second-best insertion are executed using the local insertion strategy.

The basic greedy, second-best insertion and regret-k heuristics employed here are similar to the one explained in the previous chapter, so we omit its explanation.

- **Basic greedy based on the largest demand insertion**

This heuristic only deviates from the basic greedy in the selection of a request from the U-bank. While the basic greedy heuristic simply picks the first stop from the list U and inserts it in the cheapest position, the basic greedy based on the largest demand aims to select unplanned stops from the list U based on the number of demands. Consequently, the request with the largest demand in the list U is the first to be inserted.

4-3-3- Redistribution operator

After employing a set of large neighborhood search heuristics in the previous stage, the distribution of students between routes may lose its balance. This implies that some routes contain a large number of students, while others include smaller numbers. To address this situation, a redistribution operator is employed to balance the current capacity in the current solution. Initially, a list of routes is created in decreasing order based on the occupied capacity.

Subsequently, for the first β routes in the list, an attempt is made to move students to another allowable stop in another route, if possible. In this study, this value is set to 25%, determined through a pilot study. The only exception is when the number of routes generated in the incumbent solution is less than 4. In this case, the redistribution operator is deactivated because a lower number of routes reduces the efficiency of the redistribution operator in transferring students. Transferring students to any new route is possible when two conditions are met: first, ensuring that an allowable stop exists for the candidate student; second, making sure that a respective school is considered as well. The latter means that once there is the possibility to transfer a student, a related school must also be present at the end of the route. If this is not fulfilled, an associated school needs to be inserted.

4-3-4- Adaptive search engine

Adaptive weight adjustment assesses the importance of each removal and insertion based on their performance in generating a profitable solution. In each iteration of the ALNS heuristic, one removal and one insertion operator need to be selected. Choosing different removal and insertion operators at each iteration offers several advantages. Firstly, it encourages efficient diversification in the search. Secondly, it guides the algorithm to discover better results.

The combination of one insertion with one removal may perform well in some instances, while in other cases, different removal and insertion operators might yield better results. This alternation between various removal and insertion heuristics results in an experimentally robust heuristic. Additionally, it aids in achieving a good balance between computing time and solution quality, as implementing a number of insertion or removal operators individually can be time-consuming. The question is how the algorithm selects removal and insertion operators.

The selection of removal and insertion is governed by a roulette-wheel mechanism, where each operator is assigned a weight. The probability of selecting each heuristic depends on its past performance in previous iterations. Specifically, each operator is assigned a score, and the operator that yields a better solution has a higher probability of being selected again. This means that even an operator with poor performance still has a small chance of being chosen. In our study, the selection of removal and insertion heuristics at each iteration is carried out through a pairwise selection mechanism.

4-3-5- Adaptive weight adjustment

In this section, we detail the methods for choosing removal and insertion heuristics using the pairwise selection approach. Most literature tends to independently select removal and insertion operators, potentially overlooking the chance to assess their joint performance on the metaheuristic's performance (see Ropke and Pisinger, 2006).

To address this, the algorithm considers the joint performance of a pair of operators and assigns weight ρ_{dr} to the operators based on their performance. At the beginning of a segment, all pairs have the same weight $\rho_{dr} = 1$, and all scores are set to 0. During each segment, every time a pair of removal and insertion is applied, its score is increased by the parameters $\sigma_1^*, \sigma_2^*, \sigma_3^*$ depending on its performance. If the pair finds the best solution, the score of the pair is increased by σ_1^* ; if the solution is improved but not better than the best solution, the score of the pair is increased by σ_2^*; and finally, if the solution worsens, the score of the pair is increased by σ_3^*. After each segment is completed, the weight is updated as follows:

$$\rho_{dr} = \gamma \frac{\Psi_{dr}}{\max(1, O^*_{ij})} + (1-\gamma)\rho_{dr} \tag{4-31}$$

Similar to the method proposed by Ropke and Pisinger (2006), the value of γ represents the reaction factor, O_{ij}^* specifies the number of times the pair of removal and insertion i is applied to segment j, and Ψ_{dr} represents the score of each pair of removal and insertion. Better results achieved by each pair are assigned greater weights, increasing the likelihood of the pair being selected. Let n_d and n_r be the number of destroy and repair operators, respectively. At each iteration, the roulette wheel mechanism is employed to choose one pair of removal and insertion operators with a probability.

$$\Phi_{dr} = \frac{\rho_{dr}}{\sum_{d=1}^{nd}\sum_{r=1}^{nr}\rho_{dr}} \tag{4-32}$$

4-3-6- Acceptance and stopping criteria

Another crucial aspect of an ALNS metaheuristic involves the solution acceptance rule. After generating a new solution through destroying and rebuilding operators, the acceptance criterion rule is employed to determine whether the new solution is accepted. Various acceptance methods exist. The better acceptance method only accepts a solution if it is superior to the previous one. While this straightforward acceptance rule encourages intensification, it tends to get stuck in local optima.

To achieve this, it seems reasonable to avoid restricting the algorithm to accepting only improving solutions, allowing for the exploration of alternatives to escape local optima. Striking a balance between intensification and diversification involves considering worse solutions occasionally. In this context, the decision to accept a new solution follows the simulated annealing method (Kirkpatrick, Gelatt, and Vecchi 1983).

If a new solution $f^{/}(s)$ is better than the previous one $f(s)$, search continues with a solution $f^{/}(s)$. Otherwise, the worse solution $f^{/}(s)$ is accepted with probability: $P = exp\left(\frac{f(s)-f(s')}{T}\right)$, where $f^{/}(s)$ and $f(s)$ respectively denote the objective function of the new and incumbent solutions, and $T > 0$ signifies the temperature parameter. The process starts from the initial temperature T_{start} and the temperature is gradually decreased by replacing $T = T \times C$ at each iteration, where $0 < c < 1$ and is used to represent a cooling factor parameter.

The decrease during the algorithm's operation indicates that a non-improving solution is less likely to be chosen in subsequent iterations. It's important to note that the appropriate value for T_{start} is directly dependent on the specific problem. Rather than treating T_{start} as a fixed parameter, its initial value is calculated using the results obtained in the initial solution, following the idea suggested by Dayarian et al. (2013). In practice, the initial temperature is set to $\frac{-wf(x_0)}{\ln(0.5)}$, allowing $w\%$ worse solutions (here set to 5%) to be accepted with a probability of 50%. The specific value for parameter w needs to be determined.

4-4- Experimental analysis

Our experimental computations consist of two main parts. In the initial stage (Sections 4.4.2 & 4.4.3), calibration is conducted to determine the optimal parameters and suitable operators that significantly impact the metaheuristic's performance. For both Sections 4.4.2 and 4.4.3, the testing set comprises 10 instances (4 instances from set S, 4 instances from set M, and 2 instances from set L). After identifying the best parameter settings, an analysis is performed to explore the impact of single-load and mixed-load structures on solution quality (minimizing total traveling time). Finally, a comparison is made with current literature in this area to discern the main effects of the mixed-load strategy.

4-4-1- Instance generation

Since the presented problem has not been considered earlier, no test instances are available in the literature. To address this, new data sets comprising 100 instances are generated for experiments. The problem size in this data set varies based on the number of garages (ranging from 1 to 4), the number of schools (ranging from 1 to 10), the number of stops (ranging from 10 to 100), and the walking distance (ranging from 5 to 25). Our data set includes three sets: small, medium, and large instances. Small instances have 10 to 20 stops, while medium and large instances have between 30 to 60 and 70 to 100 stops, respectively. To simplify instance generation, the number of students is calculated based on the number of stops, with the number of students related to each stop generated as random variables between 3 to 5.

To generate the data set, 6 parameters per instance should be defined in the primary stage: n_g (the number of garages), n_h (the number of schools), n_p (the number of stops), n_s (the number of students), and w_{max} (maximum walking distance for each student to reach a bus stop). All instances are generated and scattered in the Euclidean square between (0,0) and (x_{max}, y_{max}). In order to make data set similar to the real world, the values of (x_{max}, y_{max}) are set to (80 *80 km). Each school's coordinates are generated in the area of (60 km ×60 km) with respect to the center of Euclidian square. Correspondingly, the coordinates of each stop are generated in the interval of $(w, x_{max} - w), (w, y_{max} - w)$. For each generated stop, the coordinates of each student is obtained based on the angle $\alpha_j \in [0,2\pi]$ and walking distance w from the stop. Thus, the coordinates of each student are obtained by $x = x_s + w \cos \alpha_j$ and $y = y_s + wsin\alpha_j$.

In the final step of our calculations, the allocation of students to a school is addressed. To tackle this, the average number of students for each school is computed. Subsequently, an attempt is made to assign students to the closest school until the number of assigned students to each school reaches the average value. If this condition is met, the remaining students are assigned to the second closest school, and the process continues until all students are allocated properly. The departure time from the garage is set to 6:15 for all buses. The minimum and maximum arrival times at each stop and school are randomly generated within the time frames of (6:30 to 7:00) and (8:00 to 8:30) a.m., respectively.

4-4-2- Calibration of the metaheuristic parameters

The proposed metaheuristic involves crucial parameters that need to be set and fine-tuned. This stage encompasses statistical analyses to determine the optimal parameter configuration. A full factorial experimental design is employed for parameter analysis on a subset of instances. The parameters under consideration are summarized in Table 4-3 and include number of iterations (ρ), number of iterations without improvements (δ), minimum and maximum percentage of requests to be removed (ξ_{min}, ξ_{max}), parameter to control value of $q_{max}(\eta)$, weight adjustment in roulette wheel mechanism ($\sigma_1, \sigma_2, \sigma_3, \gamma$), size of restricted candidate list α, and randomness parameter in the removal procedure (p).

The analysis results are presented in Table 4-4. The Multi ANOVA output indicates that the number of iterations, the number of iterations without improvements, the minimum and maximum number of requests to be removed, and the reaction factor for the roulette wheel weight are all significant factors influencing both solution quality and computing time (P_value less than 0.05). Notably, α and σ_1 are identified as parameters significantly affecting solution quality. The optimal parameter setting for further analysis is provided in the last column of Table 4-3.

Table (4-3) Heuristic Parameters

Parameter	Description	Values	Selected value
ρ	Defines the number of iterations	350,450,550	450
δ	Define number of iterations without improvements	10,20	10
ξ_{min}	Introduces minimum percentage of request, stops, to be removed at each ALNS iteration	2%,5%,10%	5%
ξ_{max}	Introduces maximum percentage of requests, stops, to be removed at each ALNS iteration	15%,20%,25%,30%, 35%,40%,45%	25%
η	Introduces the parameter to control the value of q_{max}	2,3	2
p	Is responsible for randomness in the removal process	2,4,6	4
σ_1	Is the weight adjustment of algorithm in roulette wheel mechanism	40,50,60	50
σ_2	Is the weight adjustment of algorithm in roulette wheel mechanism	20,30,40	20
σ_3	Is the weight adjustment of algorithm in roulette wheel mechanism	1,5,10	5
γ	Is the reaction factor of the weights in roulette wheel mechanism	0.25,0.5, 0.75	0.5
α	Size of the restricted candidate list	1,2,3,4	2

Table (4-4) Best Parameter Setting

Parameters	Computing time	Average solution cost
ρ	**p<0.05**	**p<0.05**
δ	**p<0.05**	**p<0.05**
ξ_{min}	**p<0.05**	**p<0.05**
ξ_{max}	**p<0.05**	**p<0.05**
η	0.125	**p<0.05**
p	0.072	0.0794
σ_1	0.0846	**p<0.05**
σ_2	0.061	0.937
σ_3	0.110	0.407
γ	**p<0.05**	**p<0.05**
α	**0.137**	**p<0.05**

4-4-3- Heuristic calibration

As previously mentioned, a set of removal and insertion heuristics is considered for our problem. During the operation of the metaheuristic, there may be cases where certain removal and insertion operators do not directly improve the solution. However, they create opportunities to escape local optima for other operators in subsequent iterations, ultimately leading to better-quality solutions towards the end of the search.

In other words, even if an operator exhibits weaker performance, its inclusion can stimulate other operators to effectively navigate away from local optima. However, selecting a large number of removal and insertion operators increases computing time, extends the exploration of the solution space, and results in higher computational complexity.

These findings indicate that the selection of appropriate removal and insertion operators is not straightforward and requires in-depth analysis. Striking a balance between computing time and solution quality is crucial. To address this, similar to Section 4.4.2, a full factorial experimental design is conducted with the levels presented in Table 4-3. It is important to note that other heuristic parameters are held constant at this stage, as determined in Section 4.4.2. Graphical representations of the results are displayed in Figures 4-3 and 4-4.

The analysis of variance (ANOVA) indicates that among the removal and insertion operators, Shaw removal, Worst removal, and Random removal heuristics with both basic greedy and Regret k-heuristics have a significant impact on the quality of the solution. Moreover, the SLR and Least used bus removal, Basic greedy based on largest demand, and Second-best insertion operators slightly improve the solution and display poorer performance than other considered operators. As a result, the combination of Shaw removal, Worst removal, and Random removal heuristics with both Basic greedy and Regret k-heuristic as insertion heuristics is recommended for further analysis (Section 4.4.4).

Table (4-5) Removal and Insertion Heuristics Setting

Heuristic	Value	No. of levels
Shaw removal (based on distance)	On –off	2
Shaw removal (based on demand)	On –off	2
Worst removal	On –off	2
Random removal	On –off	2
Least bus removal	On –off	2
SLR removal	On –off	2
Basic greedy	On –off	2
Basic greedy based largest demand	On –off	2
Second best insertion	On –off	2
Regret-k heuristic	On –off	2

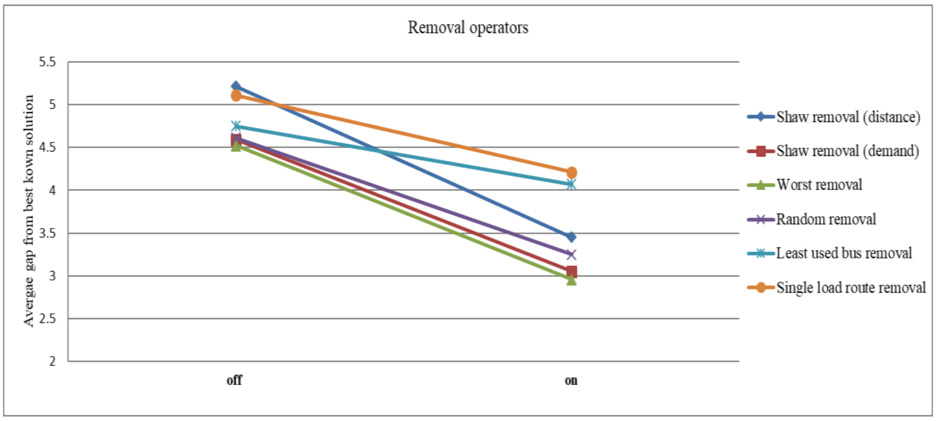


Figure (4-3) Removal Operators

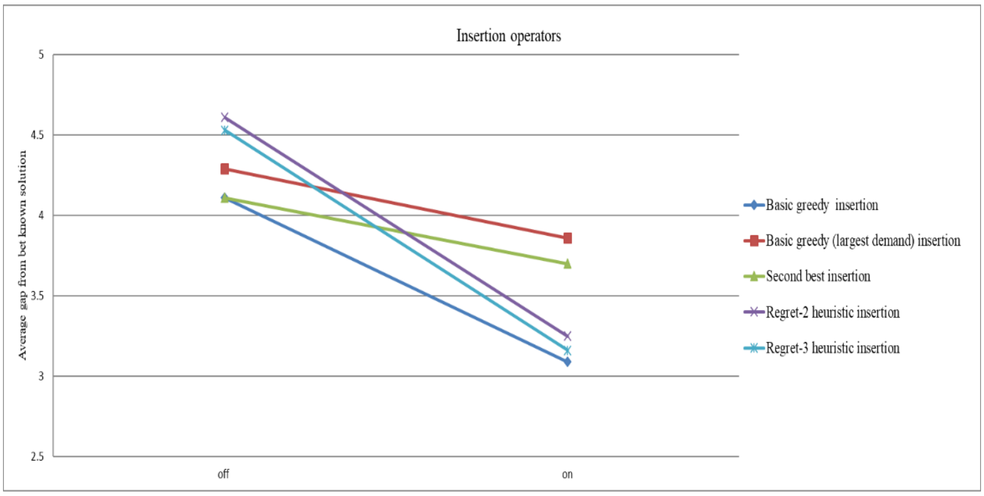


Figure (4-4) Insertion Operators

4-4-4- Computational experiments

The experiments are conducted in two categories: 1) analysis of metaheuristic configuration and 2) analysis of main characteristics.

4-4-4-1- Experiments on the configuration of metaheuristic

In this section, experiments are conducted to assess the impact of each pair of removal and insertion operators integrated into the ALNS metaheuristic on the solution's quality. At the end of each segment, the weight of each pair of removal and insertion is computed based on its achieved score. This weight correlates with the solution's quality. In practice, a pair of heuristics with a higher weight will be selected with a higher probability and has the potential to yield better solutions throughout the search.

Regarding this mechanism, the results suggest that Shaw removal with any insertion heuristic produces the highest weight. This interestingly demonstrates that what is more important is the similarity idea, whether based on demand or distance. The random removal is ranked second. Therefore, it can be said that the Shaw removal heuristic orients the intensification stage, and the random removal justifies the diversification. These findings further support the idea of using the ALNS metaheuristic. In practice, the ALNS enjoys a set of intensification and diversification heuristics that, in case some heuristics produce weak performance, while others can help to escape local optima properly.

Table (4-6) Weight Values for The Pairs of Removal and Insertion Heuristics

Pair of removal and insertion heuristics	Weight	Pair of removal and insertion heuristics	Weight
Shaw removal (distance) and basic greedy	46.37	Random removal and basic greedy	42.27
Shaw removal (distance) with regret-2	53.19	Random removal and regret-2	41.10
Shaw removal (distance) with regret-3	41.18	Random removal and regret-3	36.19
Shaw removal (demand) and basic greedy	41.13	Worst removal and basic greedy	19.45
Shaw removal (demand) with regret-2	44.28	Worst removal and regret-2	23.65
Shaw removal (demand) with regret-3	37.17	Worst removal and regret-3	26.17

4-4-4-2- Experiments on main characteristics

In this section, we aim to compare the performance of the proposed metaheuristic against the solution given by the CPLEX solver. The characteristics of the test problem sizes and the results of the metaheuristic and exact solutions are summarized in Appendix 5. The exact solution is reported as long as the optimal solution is found within 45 minutes. Since the combination of removal and insertion heuristics can suggest better performance (as shown in the results of Section 4.4.3), we only consider it in this section. For the proposed approach (ALNS metaheuristic), each instance is run 10 times, and finally, the amount of gap is presented.

Two percentage gaps are reported: the average gap, representing the percentage gap between the average costs of solutions calculated after 10 runs and the exact solution, and the best gap, indicating the percentage gap between the best solutions calculated after 10 runs and the exact solution. As expected, as the problem size increases, typically after instance 20, the exact method struggles to find a feasible solution within the allotted time. Consequently, the exact method can optimally solve instances up to 20 within a reasonable computing time.

Compared to the exact method, ALNS yields solutions with an average percentage gap lower than 1.5% and achieves optimal solutions in 6 instances.

Figures 4-5 present a comparison of key characteristics (total travel time and total number of buses) between using a single and mixed-load strategy. It's noteworthy that while our defined objective function focuses on minimizing total travel time, the mixed-load strategy effectively reduces the number of buses. Hence, we analyze the behavior of the total number of buses in our study. Our experiments clearly demonstrate that the mixed-load effect significantly reduces both the number of buses and total travel time.

This reduction is particularly pronounced in larger instances, especially in the case of total travel time. This is because, as the problem size increases, there is a greater tendency to utilize a smaller number of routes, resulting in more significant savings in travel time. Conversely, small instances exhibit the lowest deviation, attributed to the limited number of schools, making the metaheuristic's consideration of the mixed-load effect less impactful.

In terms of the number of routes, a noticeable deviation is observed for small instances. Overall, there is a 10.77% reduction in the number of routes and a 13.90% reduction in total travel time achieved through the utilization of the mixed-load effect.

Figure (4-6) further illustrates that the average weighted riding time and route length are considerably smaller than those of the mixed-load method, with reductions of 7.8% and 8.43%, respectively.

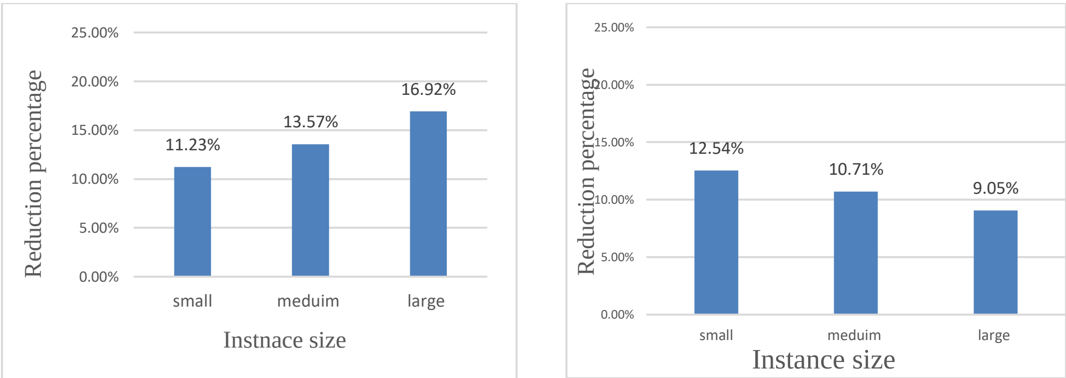


Figure (4-5) (a) Reduction Percentage in The Number of Routes (Right Side) and (b) Reduction Percentage in Total Travel Time (Left Side) while Considering Mixed Load Effect

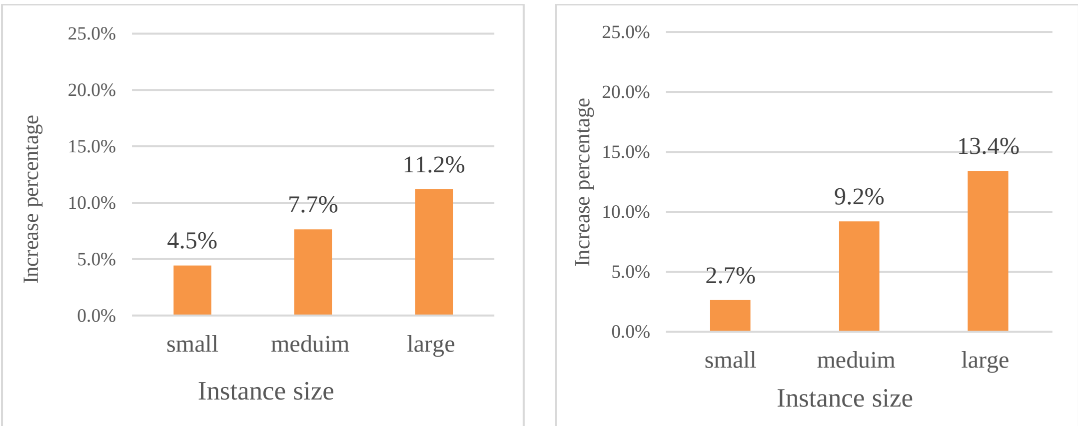


Figure (4-6) (a) Increase Percentage in Average Riding Time (Left Side) and (b) Total Route Length (Right Side) while Considering Mixed Load Effect

In our analysis of bus occupancy rates (Figure 4-7), we consider the mixed-load effect. The results show that the rate of bus occupancy experiences the least deviation from the single-load for small instances. Conversely, an improvement in bus occupancy is observed as the problem size increases.

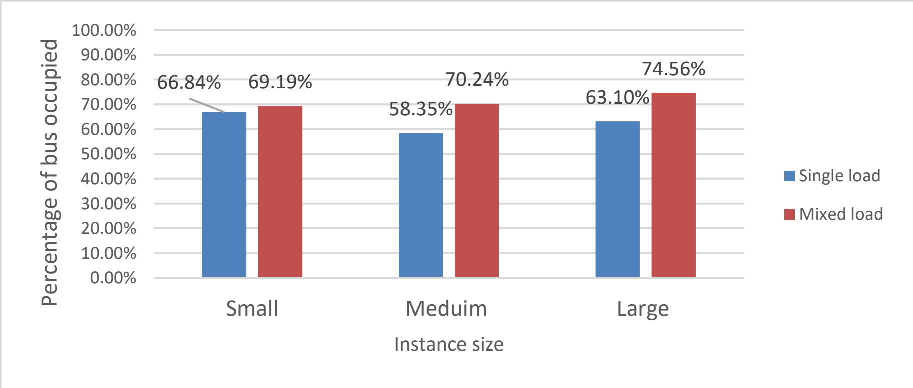


Figure (4-7) Bus Occupation Percentage while Considering Mixed Load Effect

In conclusion, the impact of the mixed-load effect on total travel time savings depends on several factors. The distance between schools is a crucial determinant; when schools are close, significant savings can occur as a bus efficiently picks up students from multiple schools and delivers them in a single route. The proximity of stops also plays a role, but the distribution of students at each stop is more critical. If stops with students from the same school are close, the benefit of mixed-load diminishes. However, when stops are farther apart, with students from different schools at each stop, the mixed-load offers better opportunities for reducing total travel time.

4-4-4-3- Comparison with best-known solutions and previous studies

To assess the efficiency and effectiveness of the proposed metaheuristics, we have conducted a comparison with a prior study in this field. Various configurations of the proposed metaheuristic, such as the simple Large Neighborhood Search (LNS) with a single removal and insertion operator, as well as the full adaptive configuration (ALNS), are evaluated against the best-known solutions in the literature. The five configurations of the simple LNS heuristic include Shaw removal (based on demand) with Basic greedy, Shaw removal (based on distance) with Regret-2, Random removal with Regret-2, Worst removal with basic greedy, and Worst removal with Regret-3.

For simplicity, we abbreviate the aforementioned LNS configurations as LNS-1 to LNS-5. Given the introduction of a new version of the school bus routing problem in this chapter, some adaptations are necessary in our assumptions.

To ensure a fair comparison with Lima et al.'s study (2016), we consider the students' locations at stops, designating the student's home as a potential bus stop. The routing cost values ($) are determined by multiplying unitary travel distance costs ($1.00) with the travel distance.

The routing cost values ($) generated by different metaheuristics (LNS and ALNS) are compared with the best-known solutions from Souza Lima's study. Table 4-7 presents the routing costs of the proposed metaheuristics alongside the algorithm proposed by Souza Lima (best-known solutions).

ALNS consistently yields the best values among the proposed configurations, with minimal deviation from Lima et al.'s study (2016). For four instances, superior results are achieved, and in other cases, the deviation is negligible (average deviation percentage around 2%).

Table (4-7) Comparison of Different Kinds of Metaheuristic (LNS and ALNS) with Best Known Solutions

Instance	P(student)	H(school)	ILS (best- known solution)	LNS-1	LNS-2	LNS-3	LNS-4	LNS-5	ALNS
1	250	6	7,024.4	7,383.0	7,353.4	7,382.7	7,179.2	7,160.5	7,189.9
2	250	12	10,575.0	11,233.8	10,867.1	10,968.0	10,886.1	10,921.0	10,575.0
3	500	12	19,368.2	20,583.1	19,588.8	20,846.3	20,193.8	20,236.5	19,329.5
4	500	25	27,066.3	29,311.5	27,974.0	29,400.6	28,909.3	28,134.4	26,714.4
5	1,000	25	52,622.6	52,807.7	56,502.1	56,894.0	55,989.8	55,533.1	54,396.8
6	1,000	50	65,139.5	67,564.8	65,682.4	68,770.3	71,426.0	68,734.2	68,982.8
7	2,000	100	89,398.6	95,099.6	95,302.9	102,410.2	93,275.2	98,253.4	92,536.5
8	2,000	100	105,215.4	117,571.1	107,700.3	109,132.8	115,472.7	108,935.1	108,161.5
9	250	6	7,930.6	8,521.5	8,031.8	8,547.5	8,695.2	8,275.9	8,097.2
10	250	12	12,224.9	13,051.4	13,032.0	12,259.7	12,889.2	12,427.4	12,399.4
11	500	12	17,681.6	19,128.6	18,689.2	19,888.9	18,452.4	18,092.8	18,450.7
12	500	25	23,037.7	24,780.5	23,231.7	24,855.9	24,046.5	24,211.4	23,751.9
13	1,000	25	50,627.1	52,713.0	50,913.0	51,867.3	53,202.9	53,092.0	51,690.2
14	100	50	66,585.9	68,397.8	68,397.8	68,605.9	68,149.6	70,491.1	66,590.4
15	2,000	50	94,661.1	101,640.7	101,276.8	103,986.3	99,529.6	96,433.1	97,747.0
16	2,000	100	88,846.8	96,216.6	93,008.0	96,509.3	91,253.7	93,507.2	89,898.2
17	250	6	10,812.2	11,820.6	11,318.5	11,856.6	11,071.6	11,285.7	11,162.5
18	250	12	14,645.7	16,190.4	14,645.3	16,239.7	15,162.9	15,254.0	15,004.5
19	500	12	21,840.4	23,912.0	22,538.0	23,984.8	22,321.6	22,587.1	21,842.7
20	500	25	24,723.6	25,073.7	24,881.5	28,159.1	25,801.4	24,984.4	24,691.4

Another noteworthy result is that, on average, all the examined LNS heuristics exhibit poorer performance compared to the Iterated Local Search (ILS) algorithm in the literature. This underscores that, for this particular problem with the specified characteristics, relying solely on a removal-insertion pair is not reliable, and an effective outcome requires a suitable combination of operators.

In plain terms, the Random removal heuristic employs a diversification strategy. Conversely, both Shaw and Worst removal operators focus on a limited portion of the solution space, yielding better results in early iterations compared to other removal heuristics. However, as the solution approaches a high-quality level, the likelihood of getting stuck in local optima increases. In cases where the aforementioned heuristics cannot operate, there's a significant risk of being trapped in a local optimum.

This illustrates that relying solely on one set of the above removal operators doesn't mitigate the risk of getting stuck in local optima. Crucially, certain operators may enhance the solution in the initial stage, while others prove beneficial towards the end of the process. Therefore, a strategic combination of diversification and intensification operators can be advantageous in the search process, where one operator compensates for the shortcomings of another. The ALNS takes advantage of this situation and as a result finds better results.

4-5- Conclusion

This study aims to introduce a new mathematical formulation and solution methodologies for the urban school bus routing problem, taking into account the mixed-load effect. The obtained results affirm the effectiveness of the proposed framework, as it yields cost savings compared to the single-load framework. Key characteristics of the School Bus Routing Problem (SBRP) in this study include homogeneous buses, maximum allowable students at each stop, school arrival time considerations, and the presence of multiple garages.

In the initial stage, the formulated instances are solved using the CPLEX solver in GAMS. Given the CPLEX solver's capability to handle 20 instances within a reasonable computing time, the ALNS with a different configuration is then proposed to solve all generated instances efficiently.

To efficiently investigate the algorithm, four lines of experiments are proposed. Firstly, a comparison between single-load and mixed-load is conducted, examining various outputs such as total travel time and the number of buses. Secondly, an analysis of different configurations of metaheuristics is performed, including each pair of removal and insertion operators embedded in the ALNS metaheuristic, to assess their impact on solution quality. In the third line of experiments, the performance of the proposed algorithm is compared with the solution provided by GAMS/CPLEX.
Finally, a fair comparison with a different study is conducted.

In the earlier case, it is demonstrated that considering the mixed-load effect leads to better solutions.

In the second analysis, among different combinations, the Shaw removal with Regret-2 gains more weight.

In the third line of experiments, the solution of the proposed metaheuristic is compared against the solution of GAMS/CPLEX. The results show that the average percentage gap from GAMS/CPLEX is lower than 1.5%, and optimal solutions are found in 7 instances.

In the final comparison with the best study, a promising result is observed when considering the ALNS metaheuristic. However, the results of other LNS configurations are not very encouraging.

The current study suggests several promising directions for further research. In the first direction, researchers may consider incorporating additional constraints and features into the proposed model to make it closer to reality. Specifically, accounting for simultaneous morning and afternoon delivery and pickup of students could be explored.

A second research line involves employing specialized neighborhoods to address the characteristics of the problem and thereby reduce the complexity of problem-solving

Chapter 5:

A new metaheuristic using adaptive neighborhood selection for school bus routing problem: considering mixed load and multi-shift load conditions

5-1- Introduction

Municipalities aim to develop efficient operational strategies for managing school bus systems. These strategies should align with annual student transportation budgets, encompassing operational costs, driver employment expenses, vehicle costs, bus stop equipment, and cost-effective solutions.

In Tehran, despite the growing number of transported students, the budget has either remained unchanged or experienced marginal growth compared to previous years. This highlights the municipality's necessity to adopt a cost-effective approach for the efficient management of the approved budget. The policymaker must establish an effective mechanism to control, and to some extent, optimize the overall transport cost, encompassing both fixed and variable costs.

In this context, introducing the integrated method applicable for both morning and afternoon shifts can serve as a valuable tool for cost optimization or, at the very least, cost control. Although any optimization or cost reduction solutions come with initial expenses, they can ultimately lead to more significant reductions in operating costs throughout the year.

In Tehran, a significant and growing challenge is the lack of integrity in picking up and dropping off students during morning and afternoon shifts. The congestion and the higher number of students during morning hours, attributed to some schools operating exclusively in the morning shift, further complicate the issue. This complexity makes it difficult for municipalities to establish a robust framework for integrating the two shifts.

Moreover, the substantial gap between the start time of the afternoon shift and the end time of the morning shift poses a major obstacle to successful integration. These issues have led to independent scheduling and separate service distribution for morning and afternoon shifts, sometimes outsourcing these tasks to the private sector.

Another challenge arises when students from the same bus stop follow different school time windows. In practice, at each stop, there may be primary and elementary school students with different time windows. If a bus accommodates students from different schools simultaneously, there's a higher likelihood of missing the chance to pick up eligible students at subsequent stops, resulting in increased computation time and costs.

Unlike the classical version of routing problems, where each customer is visited only once, allowing split loading permits multiple visits to serve students from the same stop. This approach has the potential to yield significant savings in travel costs and fleet numbers. The Split Delivery Vehicle Routing Problem (SDVRP) can be addressed within the framework of the general Vehicle Routing Problem (VRP), as discussed by Archetti and Speranza (2012) and Irnich et al. (2014). By employing split pick-up and drop-off methods, each stop can be serviced through multiple visits, proving particularly beneficial in scenarios where a candidate stop hosts students from different schools (more than 2 schools) or encounters tight capacity constraints

To optimize student transportation, careful attention should be given to the diverse time windows, various trip types (morning and afternoon), students' riding time, and the split mechanism for picking up students. To create an efficient model and attain desired outcomes, it is crucial to integrate morning and afternoon shifts, incorporating split pickups at each stop.

The comprehensive schedule for student transportation spans from 6:45 AM to 6:15 PM, encompassing three distinct phases throughout the day.

The initial phase (morning) involves transporting students from their homes to school, spanning from 6:45 AM to 8 AM. The second phase encompasses gathering morning-shift students from schools and transporting afternoon-shift students from their locations to their respective schools, taking place from 11 AM to 1:45 PM. The final phase focuses on transporting afternoon-shift students from schools to their homes, occurring between 5 PM and 6:15 PM. These three phases can be coordinated with forward, simultaneously forward-backward, and backward schemes for the school bus routing problem, respectively.

While many studies recommend addressing morning and afternoon issues separately due to their complexity, the crucial aspect for the public manager is to contemplate both shifts simultaneously. This approach aids in creating a cost-effective model for school transportation, aligning with the objectives of the current study.

This chapter makes a following contribution. From a mathematical perspective, we address morning and afternoon students through a mixed-load scheme (between 11:15 AM and 2:00 PM) in the following manner.

Each bus departs from its parking space, sequentially serves the set of bus stops, picks up afternoon students, and transports them to their respective schools—this represents the forward path. Conversely, in the backward case, the bus picks up morning students from their schools and drops them off at their respective bus stops. In the forward path, bus stops are the pick-up points, and schools are the drop-off points, while the reverse is true for the backward path.

Our primary focus lies in developing an efficient and innovative framework for assigning a time window constraint for each demand (i.e., student) rather than each node (i.e., bus stop and school). This implies that students from the same stop may have different time windows. As a result, our model can utilize the split-load scheme to pick up students with tighter (i.e., earlier) time window restrictions. This aligns with the concept of incorporating the split load effect.

Regarding the solution methodology, we aim to implement constructive heuristic strategies, including the selection of local search operators based on their performance within the search space. This approach establishes an efficient mechanism that reduces computing time, enables the use of appropriate local search operators, and enhances diversification.

To the best of our knowledge, the mixed-load School Bus Routing Problem (SBRP) has garnered significant attention in the past seven years and is likely to remain a focus of research in the near future. Integrating the concept of morning and afternoon trips brings the SBRP closer to real-world conditions. Although split loading has gained increased attention, it remains largely overlooked in the SBRP field. Notably, no major study to date has examined the impact of mixed-load and split-load school bus routing during both morning and afternoon trips. This research delves into these three characteristics.

This study makes the following contributions:

1) Introducing an innovative School Bus Routing Problem (SBRP) scheme that incorporates morning and afternoon features, considering the mixed-load effect (transporting students from different schools on the same bus), multi-shift load effect (accommodating students from both morning and afternoon shifts simultaneously), and split load.

2) Creating specific neighborhoods tailored to the SBRP.

3) Designing an iterated local search metaheuristic with an adaptive mechanism.

4) Implementing multi-shift loading to effectively reduce operational costs.

5-2- Problem definition

Overall, the problem definition is categorized as follows

1) The bus departs from the garage to pick up students for the first trip from their respective stops. After picking them up, the bus transports them to their respective schools, resembling the forward mechanism.

2) Upon dropping off students from the first trip at their schools, the bus proceeds to pick up morning-shift students from their schools, transporting them to their designated stops. This process is part of the backward trip.

3) Finally, the bus returns to the nearest garage.

Innovatively, we introduce a multi-shift loading approach with a mixed-load concept, which is addressed using an Iterated Local Search (ILS)-based heuristic. The scenario involves multiple schools, a single student type, numerous bus stops, and a set of garages.

The problem assumes the advance knowledge of the locations of garages, schools, stops, and the assigned time windows for students. It encompasses both morning and afternoon students. The process involves each bus commencing its journey from the garage, picking up a subset of students from their designated bus stops (i.e., the origin point), and transporting them to their respective schools. Subsequently, the bus picks up students who have completed their school day and transports them to their designated stops.

Our model introduces two key innovations. Firstly, instead of imposing time constraints on nodes (i.e., stops and schools), the constraints are applied on the demand side, focusing on students. Each student is assigned lower and upper time-window bounds. In the first shift, the lower bound determines the time a student is available at the stop to be collected, while the upper bound specifies the maximum time for the student to be dropped off at the associated school. Conversely, in the second shift, the lower bound indicates the time a student is available at the school to be picked up, and the upper bound represents the maximum time for the student to be dropped off at the bus stop (home).

The model incorporates constraints to closely mirror real-world scenarios:

1)Time window constraints are applied to each student in both shifts, as opposed to schools and stops;

2) The bus capacity constraints must not be violated, and this capacity is the same for all busses;

3) A maximum riding time constraint is enforced to enhance student convenience;

4) The model allows for the simultaneous handling of morning and afternoon students (multi-shift loading) and the transportation of students from different schools (mixed-loading);

5) Split-loading is permitted at each stop, providing flexibility in the loading process;

Table 5-1 provides a summary of the parameters and variables incorporated in the model, while Figure 5-1 illustrates an example of the problem. In the figure, a student is represented by a circle, a bus stop by a small square, a garage by a large black square, and a school by a triangle. Students share the same color as their respective schools. Bold lines indicate forward trips, and dashed lines represent backward trips. The number of students ready for the return trip is visualized under each school.

Figure 5-1 presents a feasible (though not optimal) solution, where each bus must initiate from a garage, pick up students from various stops, and transport them to their respective schools.

The problem is based on the following assumptions:

1)Buses have the flexibility to transport students from different schools simultaneously;

2) In both forward and backward trips, students are picked up before being dropped off;

3) Each school and stop may be visited by more than one bus;

4) Each student is assigned to either the morning or the afternoon shift;

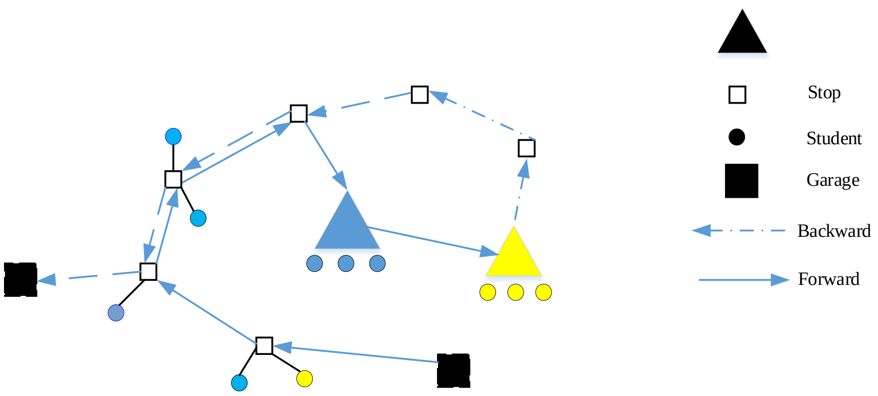


Figure (5-1) An Example of School Bus Routing Problem

Table 5-1 presents the sets, parameters, and decision-variables used in the model.

Table (5-1) Indices, Sets, Parameters, and Decision Variables Used in the Mathematical Model

Sets	
Q	Set of students (q)
K	Set of school buses (k)
$I = \{G\} \cup S \cup C \cup S'$	Set of nodes (i, j), where G is the set of the garages, S is the set of bus stops, C is the set of schools, and S' is the set of duplicated bus stops
Parameters	
$M1 - M5$	Five large numbers
C_k	Capacity of bus k
s_q	Time for student q to get on/off a bus
G_q	Origin of student q
N_q	Destination of student q
G_i'	Students who have node i as their origin
N_i'	Students who have node i as their destination
lb_q	Lower bound of time window for student q
ub_q	Upper bound of time window for student q
RT_q	Maximum riding time of student q
$t_{i,j,k}$	Transportation time from node i to node j by bus k
Decision variables	
$X_{i,j,k}$	A binary variable that is 1 if the bus k moves from node i to node j, and 0 otherwise
$Y_{i,k}$	A binary variable that is 1 if node i is visited by the bus k, and 0 otherwise
$P_{q,k}$	A binary variable that is 1 if student q is picked up by bus k, and 0 otherwise
$D_{q,k}$	A binary variable that is 1 if student q is delivered by bus k, and 0 otherwise
$L_{i,k}$	A nonnegative variable showing the number of students in bus k when it leaves node i
$T_{i,k}$	A nonnegative variable showing the time the bus k visits node i

The mixed integer linear programming (MILP) formulation for the SBRP defined above is as follows:

$$Min \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} t_{i,j,k} * X_{i,j,k} \quad (5\text{-}1)$$

$s.t.$

$$\sum_{j\in I} X_{1,j,k} \leq 1 \qquad \forall\, k \in K \qquad (5\text{-}2)$$

$$\sum_{i\in I} X_{i,1,k} \leq 1 \qquad \forall\, k \in K \qquad (5\text{-}3)$$

$$\sum_{i\in I} X_{i,j,k} = \sum_{i\in I} X_{j,i,k} \qquad \forall j \in I, k \in K \qquad (5\text{-}4)$$

$$\sum_{j\in I} X_{i,j,k} = Y_{i,k} \qquad \forall i \in I, k \in K \qquad (5\text{-}5)$$

$$\sum_{k\in K} Y_{i,k} \geq 1 \qquad \forall i \in I \qquad (5\text{-}6)$$

$$\sum_{k\in K} X_{i,i,k} = 0 \qquad \forall i \in I \qquad (5\text{-}7)$$

$$\sum_{i\in S}\sum_{i\in C} X_{i,j,k} \leq 1 \qquad \forall k \in K \qquad (5\text{-}8)$$

$$\sum_{i\in C}\sum_{i\in S'} X_{i,j,k} \leq 1 \qquad \forall k \in K \qquad (5\text{-}9)$$

$$P_{q,k} \leq Y_{i,k} \qquad \forall\, i \in I - \{\mathrm{G}\}, k \in K, q \in G_i' \qquad (5\text{-}10)$$

$$\sum_{k\in K} P_{q,k} = 1 \qquad \forall q \in Q \qquad (5\text{-}11)$$

$$D_{q,k} \leq Y_{i,k} \qquad \forall\, i \in I - \{\mathrm{G}\}, k \in K, q \in N_i' \qquad (5\text{-}12)$$

$$P_{q,k} = D_{q,k} \qquad \forall k \in K, q \in Q \qquad (5\text{-}13)$$

$$\sum_{q\in G_i'} P_{q,k} + \sum_{q\in N_i'} D_{q,k} \geq Y_{i,k} \qquad \forall i \in I - \{\mathrm{G}\}, k \in K \qquad (5\text{-}14)$$

$$L_{i,k} + \sum_{q\in G_j'} P_{q,k} - \sum_{q\in N_j'} D_{q,k} \leq L_{j,k} + \left(1 - X_{i,j,k}\right) M1 \qquad \forall i \in I, j \in I - \{\mathrm{G}\}, i \neq j, k \in K \qquad (5\text{-}15)$$

$$L_{i,k} \leq C_k \qquad \forall i \in I, k \in K \qquad (5\text{-}16)$$

$$T_{i,k} + s_q\Big(\sum_{q\in G_i'} P_{q,k} + \sum_{q\in N_i'} D_{q,k}\Big) + t_{i,j,k} \leq T_{j,k} + \left(1 - X_{i,j,k}\right)M2 \qquad \forall i \in I, j \in I - \{\mathrm{G}\}, i \neq j, k \in K \qquad (5\text{-}17)$$

$$T_{j,k} - T_{i,k} \leq RT_q + \left(1 - P_{q,k}\right)M3 \qquad \forall k \in K, q \in Q, i = G_q, j = N_q \qquad (5\text{-}18)$$

$$T_{j,k} \geq T_{i,k} - \left(1 - P_{q,k}\right)M4 \qquad \forall k \in K, q \in Q, i = G_q, \quad j = N_q \qquad (5\text{-}19)$$

$$T_{i,k} \geq lb_q P_{q,k} \qquad \forall q \in Q, i = G_q, k \in K \qquad (5\text{-}20)$$

$$T_{i,k} \leq ub_q + \left(1 - D_{q,k}\right)M5 \qquad \forall q \in Q, i = N_q, k \in K \qquad (5\text{-}21)$$

$$X_{i,j,k}, Y_{i,k}, P_{q,k}, D_{q,k} \in \{0,1\};\ L_{i,k}, T_{i,k} \geq 0 \qquad (5\text{-}22)$$

The objective function (5-1) serves to minimize the total travel time of buses. Constraints (5-2) and (5-3) are degree constraints. Constraint set (5-4) is flow conservation, and constraints (5-5) determine whether each node is visited by a bus or not. Constraints (5-6) ensure that each node is visited at least once. Constraints (5-7) prevent loops in nodes. Constraints (5-8) ensure that a bus travels from a given set of stops to a set of schools at most once, while constraints (5-9) guarantee that a bus travels from the set of schools to the set of stops at most once. Constraints (5-10) state that if a node is visited by a bus, the students on that node can be picked up. Constraints

(5-11) ensure that all students must be picked up by the bus. Constraints (5-12) state a condition similar to (5-10) for the delivery of students. Constraints (5-13) ensure that if a bus picks up a student, it also take him/her to the related destination. Constraints (5-14) prevent unnecessary node visits to buses.

Constraints (5-15) calculate the number of students in a bus after leaving the nodes, while constraints (5-16) limit the bus load.

Constraints (5-17) are time-related constrains. Constraints (5-18) ensure that the maximum riding time is not exceeded. Constraints (5-19) guarantee that a student is first picked up and only then dropped off. Constraints (5-20) and (5-21) specify the time window of each student. Finally, variables and their types are expressed in constrains (5-22).

5-3- Solution methodology

The School Bus Routing Problem (SBRP) is recognized as an NP-hard problem, making it impractical to solve in polynomial time. For large-scale instances, heuristic approaches prove more effective. The challenge lies in designing a heuristic that aligns with the problem's characteristics, finding a balance between computing time and solution quality, and managing the trade-off between intensification and diversification. Neglecting these considerations hampers the efficiency of the heuristic.

Researchers adopt varied approaches based on the problem's peculiarities and scope. In this context, we propose an Iterated Local Search (ILS) heuristic for the SBRP, incorporating an adaptive mechanism. The structure and details of the iterated local search have been extensively presented by Lourenço, Martin, and Stützle (2003).

The Iterated Local Search (ILS) is a straightforward, easily implementable, and robust metaheuristic, with its effectiveness hinging on the iterative implementation of embedded local search, perturbation, and acceptance criteria. The ILS consists of three key phases: constructive, improvement, and perturbation. Widely acknowledged in the literature, the ILS metaheuristic is a well-established methodology for the Vehicle Routing Problem (VRP) and its variants.

The initial solution is generated using a constructive heuristic based on predefined criteria like maximum riding time, demands time window, and capacity constraints. Subsequently, the solution undergoes local search for further enhancement until a local optimum is reached. To avoid local optima and establish a favorable starting point for the improvement phase, a diversification mechanism is applied to the solution, employing the destroy and repair method.

The ILS heuristic incorporates a notable adaptation by introducing an adaptive mechanism in the local search block. Instead of adhering to the traditional fixed method of ordering neighborhoods, each neighborhood is selected based on its outcome in the preceding operation. This entails assigning a higher weight to neighborhoods with superior performance, thereby increasing their probability of selection. Simultaneously, neighborhoods with lower levels of effectiveness also have a chance, albeit a lower one, of being chosen. The general scheme is illustrated in Algorithm 5-1.

Algorithm (5-1) The Proposed Algorithm (ILS with an Embedded Adaptive Mechanism)

Input: U (set of all potential stops), G (set of all garages), P^{-} (set of all schools), S (set of all students), I (Set of operators), q (percentage of route to be destroyed), P^{+}(List of stops to which students are allocated), μ (initial score of operator (H)), w (initial weight of operator (H)), nit (number of iterations without improvement), It (total number of iterations that contains no of segments), and *n (*number of iterations in each segment), nit_{max} (maximum number of iterations without improvement);

Stage 1: Construction phase

P^{+}=all students allocated to the respective stop // Student allocation

$x_o = route\ generation(\ p^{+},\ p^{-}$,S,G,) // Generating route using constructive heuristic

$x_{best} = x_o$

$f_{best} = f(x_o)$

$x_{act} = x_o$

Stage 2: Improvement phase

Set of heuristics

Initialize the roulette wheel; initialize the adaptive parameters (μ, w)

While Stopping criterion It is not met,

  For *seg←1 to n*

    Roulette wheel mechanism: Select **one** operator $\boldsymbol{H} \in \boldsymbol{I}$ through the adaptive mechanism

    x_{act}^{*}= H (x_{act}) // improve the solution by applying the selected improvement

    If accept (x_{act}^{*}, x_{act}),

      $x_{act} = x_{act}^{*}$

    End if

    If $f(x_{act}) < f(x_{best})$

      $x_{best} = x_{act}$

    End if

    Update the number of iterations without improvement (nit), update the collected score of operator

    If max number of iterations without improvement reached

Stage 3: Perturbation phase

      Update the parameter q

      x_{act}= Perturb (x_{act}, q) by applying the perturbation neighborhood

      $nIt = 0$

    End if

  End for

  Update the roulette wheel parameter (μ, w)

End while

Output: x_{best}

5-3-1- Construction phase

The constructive heuristic introduced here is combined with a Greedy Randomized mechanism known as GRASP. The GRASP configuration has garnered particular attention in the realm of the Vehicle Routing Problem (VRP) and its variants. The goal is to sequentially develop an initial solution for two types of trips: forward and backward.

During the forward trip, the objective is to generate routes where the bus initiates from the current garage, serves a set of stops, picks up afternoon students, and ultimately drops them off at their respective schools. It's important to note that, in the first trip, stops are assigned to the closest garage, as detailed in Chapter 4.

In contrast, during the backward trip, the bus picks up morning students from their schools, leaves the school premises, transports them to their designated stops, and finally returns to the garage. Given the assumption of a multi-load scheme in the problem, it's possible for students from both the first and second trips to be on the same bus simultaneously.

The construction process of our problem is divided into four phases: phases 1 and 2 occur during the forward trip, while phases 3 and 4 accompany the backward trip. Here's a breakdown of the forward trip:

In phase1, the heuristic selects a set of bus stops to generate a route.

Phase 2 prioritizes visiting those candidate schools whose students have already been picked up by the bus.

In contrast, during the backward trip:

phase 3 identifies which schools can be chosen (i.e., schools that the bus can visit) to pick up students.

Phase 4 involves transporting the students from the visited schools to their residence locations.

To construct phase 1, two tasks need to be undertaken, namely tasks 1 and 2. Task 1 determines the mechanism for selecting the next stop to be visited, and task 2 determines which students in the selected stop can be collected by the bus. Since this study assumes a split load situation, not all demands (i.e., students) of a candidate stop might be collected in the first visit. In this structure, considering the first task, total $t_{score1} = \frac{s_{1+}s_2}{s_3}$ is calculated for all non-visited stops and added to the list U.

In this relation, s_1 represents the number of students in the candidate stop that have a common school with students who have already been picked up by the bus, s_2 indicates the number of allowable students at the candidate stop with respect to time constraints (both maximum riding time of students and time window of the school); and s_3 shows the distance between the last stop visited by the bus and next candidate stop.

This formula prioritizes stops with shorter distances and students from more commonly associated schools. After calculating scores for all stops in list U, they're sorted in decreasing order of total score. A restricted candidate list (RCL) is formed with the first α non-visited stops. A random stop from RCL becomes the current node. Upon selecting a candidate stop to add to the current route, it's crucial to identify its allowable students. Notably, while split demands are allowed, determining the priority is essential. The criteria include giving precedence to students whose schools have already been chosen by the bus.

If the candidate stop includes students whose schools have not been considered yet, those students are randomly selected based on predefined constraints. Once all students from a candidate stop are taken by bus, the stop is removed from list U, which is then updated. If some students at the stop remain unassigned, the candidate stop stays available for further processing.

When adding a stop to the current route, the respective school for each selected student needs to be located in the cheapest feasible position. This process continues, and a non-visited stop is added to the current route as long as one of the following situations arises: 1. The list U becomes empty (no eligible nodes to add); 2. Candidate stops cannot meet both maximum riding time and time window constraints simultaneously; 3. The bus capacity constraints are no longer satisfied.

If any of these situations arises, we cease operations on the current route in the first shift. Consequently, the current route is returned to the respective school(s), and the heuristic endeavors to continue in the second shift. Specifically, in the initial step, the heuristic aims to create a new route for the available stops, maintaining the current route as long as a feasible solution is identified. If none of these situations occur, the bus must decide to return to its respective school to drop off the assigned students. In this second phase, it is crucial to consider the priority of schools to be served based on their distance, while adhering to the constraints under consideration.

In phase 1, before the heuristic considers additional candidate stops, we need to check and update the status of routed stops, those already visited by the bus, based on predefined constraints. To save time, an efficient data structure is employed to manage the allowable riding time of students to reach their respective schools and their school time windows in advance during phases 1and 2.

The return trip commences from the school, serving its students within their designated time windows and transporting them to their residences. In our problem, with a multi-shift loading scenario, dropping off first-shift students at their respective schools and picking up second-shift students can occur simultaneously. This allows the bus to pick up second-shift students from their school while still transporting first-shift students.

Developing backward trips mirrors forward trips with minor differences. After dropping off first-shift students, the bus heads to a school to pick up second-shift students, which could be the same location or a different one. Phase 3 addresses this issue. Initially, the first available schools within the time window are filtered for the current bus. Subsequently, the total score for each school is calculated as the sum of two partial scores, combined using the formula: $t_{score2} = \delta s_1 + \beta s_2$.. Here, s1 reflects the similarity between candidate schools' stops and those visited by the bus in the morning trip. When the school is selected, the locations of stops where students are to be dropped off are examined, calculating the number of stops visited by the bus in both morning and afternoon shifts. Additionally, s_2 calculates the distance of the bus from each candidate school. The coefficients δ and β can be assigned values between 0 and 1, based on the constraint $\delta + \beta = 1$.

The optimal values for both coefficients are provided in Table 5-3. Subsequently, eligible schools are arranged in ascending order of total score. The heuristic initiates school selection from the top of the list, exploring sequentially until the bus capacity constraints are reached. Ultimately, in phase 4, a decision is made to plan the return trip. Similar to phase 2, this is executed based on minimum distance. The sole difference in constructing routes between two shifts is the exclusion of the split load framework from the return trip. To enhance the heuristic's efficiency, an effective data structure is defined and updated when new candidate stops are considered.

Table 5-2 illustrates the sample data structure. The parameters in the data structure encompass the next visited stop (NVS), the number of available students at each stop (AS), the cumulative load of the bus upon arrival at the candidate stop (CL), the load of the bus after visiting the candidate stop (CLA), the remaining capacity before visiting the candidate stop (R), the remaining capacity after visiting the candidate stop (RC), the arrival time at the candidate stop (AT), and the time to reach the respective school (RT). A similar structure is designed for the backward trip. This straightforward yet effective data structure enables the solution to manage both capacity and time constraints before taking any action, resulting in more efficiency.

As depicted in Figure 5-2, when the candidate bus reaches stop 3, there are three available students to be picked up. Taking all students at this stop leads to a capacity violation. Consequently, it must be determined which students to serve to satisfy all constraints. Opting for student b would result in a time window violation. This situation prompts the use of the split load framework, allowing the bus to take students a and c, while student b remains unserved. This approach not only avoids time window violations but also reduces travel time. All relevant values are checked and updated when a new school or stop is introduced into the problem.

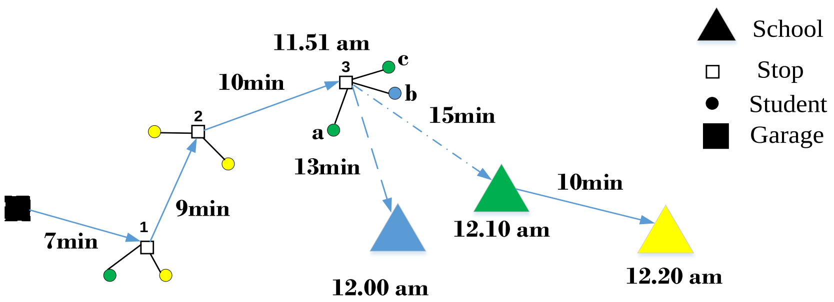


Figure (5-2) Example for Constructing an Initial Solution

Table (5-2) Data Structure

NVS	AS	CL	CLA	R	RC	AT	(RT) Travel time to school 1	(RT) Travel time to school 2
3	3	4	7	2	-1	11.51	13 min	15 min

5-3-2- Improvement phase

As previously mentioned, the local search operator aims to iteratively modify the current solution, proven effective in solving VRPs. Before entering the improvement block, two crucial questions must be addressed: 1) What kind of neighborhood is specific to SBRP; and 2) Which set of neighborhoods strikes a proper balance between computational effort and cost. Thus, it is imperative to develop operators adaptable to the problem at hand. Larger-sized neighborhoods can provide higher-quality solutions, but they sacrifice computational time efficiency. Conversely, small local search operators have quick execution times but may not yield substantial improvement.

Therefore, it's crucial to identify neighborhoods that explore more efficiently within a desirable computing time. To achieve this, the following strategies are considered. The first strategy involves assessing whether the neighborhood can be divided into two or more smaller neighborhoods. The rationale is that when a candidate neighborhood operates with a smaller size, it incurs lower time complexity. The second strategy is built on heuristic pruning, effectively implementing the neighborhood based on its performance. Another noteworthy strategy is to introduce SBRP-specific operators. Designing a neighborhood compatible with problem-specific knowledge proves helpful, especially as the problem size expands.

This approach focuses on creating a suitable mechanism for exploration. Specifically, in some cases, simultaneous exploration within one route or between two routes may not lead to further improvement due to tightly applied constraints. Hence, it becomes essential to explore three or more routes.

These considerations underscore the importance of incorporating the aforementioned strategies in the improvement phase. We introduce two types of neighborhoods, each tailored to SBRP's specific requirements. The common neighborhood aligns with the well-established VRP framework, while the other leverages insights derived from problem characteristics like single-load, mixed-load, and split load scenarios. For heuristic pruning, we implement a neighborhood selection mechanism, which will be elaborated on in Section 5-3.3.

5-3-2-1- Common neighborhood

In literature, various neighborhoods have been used, but it's crucial to select ones that balance computing time and solution quality. In this study, we employ cross-exchange and ejection chains as these operators can create one or more small neighborhoods. Cross-exchange operates on a smaller neighborhood size (e.g., swap or remove-insert), requiring less computing time, while the larger version can yield high-quality solutions.

☐ Cross-exchange operator

This operator exchanges pairs of consecutive stops between two routes simultaneously. An example is provided in Figure 5-3. If one sequence is empty (contains no stops), but another sequence has only one stop, the operator functions as a remove-insert mechanism. Therefore, when both sequences contain only one stop, the operator proceeds with its swapping task. This flexibility allows the cross-exchange operator to act as a cross-exchange, remove-insert, or swap operator. This justification demonstrates that while the cross-exchange behaves like a remove-insert operation, the solution benefits from shorter computing time within the search space.

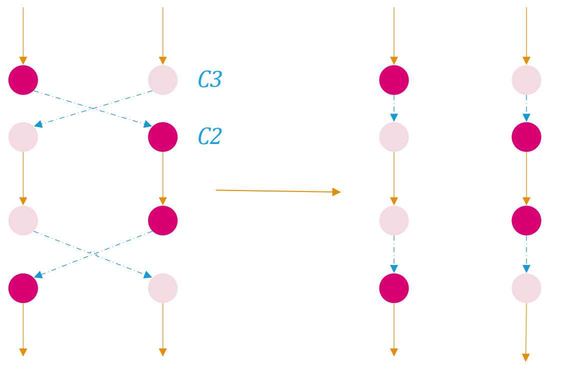


Figure (5-3) Cross-exchange Mechanism Between Sequences of Two Stops

☐ Ejection Chain

The ejection chain concept, introduced by Rego (2001), tackles simultaneous exploration in multiple routes. It involves relocating a stop from route A to route B, followed by the relocation of a different stop from route B to route C. This chain relocation between routes enhances diversification in the solution space to a certain extent. The following figure illustrates the ejection mechanism.

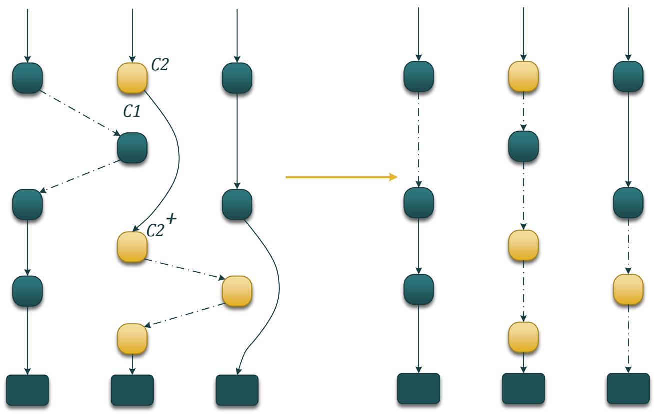


Figure (5-4) Ejection Mechanism with Two Relocations

Due to the high computational complexity of both aforementioned neighborhoods, particularly for large instances, we restrict their exploration to specific parts of the solution space. Consequently, these neighborhoods are designed to facilitate exchanges within a defined area of the solution, aiming to reduce computational effort. Specifically, when a stop is a candidate for the operation, the solution focuses on finding the closest stop.

5-3-2-2- Special neighborhood

The special neighborhood is implemented based on the nature of the problem.

- **Inter-route swap of stops with students of the same school(s)**

This operator swaps the positions of two stops between two routes when they have students from the same school(s). This maneuver aids in preventing the insertion of additional schools into the route.

- **Convert single to mixed-load effect**

To minimize the number of single-load routes, this operator aims to remove a stop from a single-load route and insert it into a mixed-load route.

- **Convert mixed-load to single-load effect**

This operator seeks to eliminate a stop from a mixed-load route and insert it into another route, thereby removing the stop whose student has a far distance relative to other existing school(s).

- **Split load**

Two operators related to split load are run in order to examine the characteristics of the problem more precisely.

- **Swap (1, 1)**

The idea of split swap is proposed by Boudia et al. (2007). This operator provides some modification based on the amount of demand of each stop. Suppose that two stops, nodes i and j, are candidate for exchange. Node i belongs to route 1 and node j belongs to route 2. Depending the amount of demand, at point $d_i > d_j$ node j is added to route 1, either before or after node i, in the best (i.e., cheapest) position, with demand j and node i is set with new demand $d_{i1} = d_i - d_j$. Also, a copy of node i is replaced in route 2 with a demand equal to $d_{i2} = d_j$. If $d_i < d_j$, the inverse behavior is observed. In case $d_i = d_j$, this operator is not taken into account and the move is discarded.

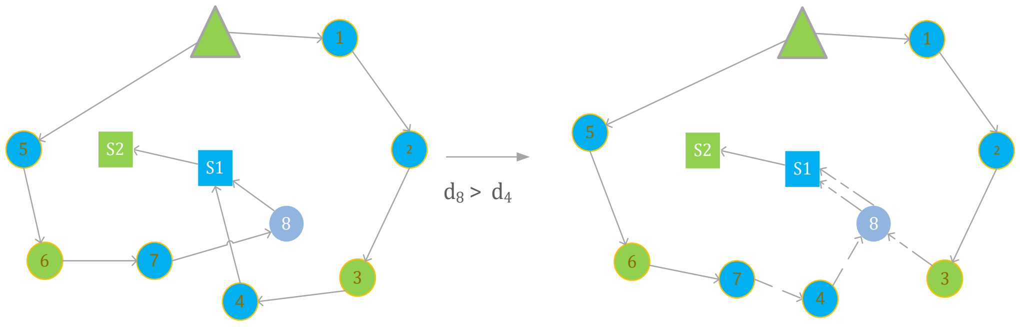


Figure (5-5) Swap (1,1)

◻ Swap (2, 1)

It follows the idea of Swap (1, 1) yet with the following adaption. We consider two nodes i and j for route 1 and node k for route 2 . To this end, we follow two scenarios: in the first one, when $d_i + d_j \geq d_k$ and $d_k > d_i$, node i and the copy of j are replaced in route 2 with d_i and $d_{j1} = d_k - d_i$. Additionally, node k is inserted in route 1 before and after node j in the cheapest position with d_k. Remaining demand d_{j2} is replaced in route 1 with $d_{j2} = d_j - (d_i + d_k)$. In the second case, if $d_i + d_j < d_k$, the insertion position for i and j can be either before or after k with demands d_i, d_j and $d_{k1} = d_k - (d_i + d_j)$ in route r_2. Additionally, the copy of node k is replaced in route r_1 with a demand equal to $d_k = (d_i + d_j)$.

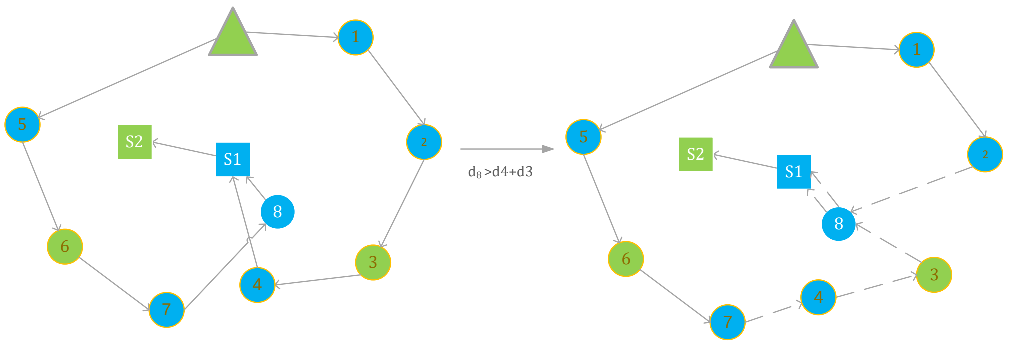


Figure (5-6) Swap (2,1)

To decrease complexity in the solution space, it's essential to check both aforementioned constraints and costs before implementing any move. The move is executed only if it achieves a desirable outcome for both issues (positive saving and a feasible solution); otherwise, it's discarded.

In summary, special neighborhoods are introduced and applied that incorporate the split load concept in SBRP. Furthermore, for an effective trade-off between complexity and solution quality, the cross-exchange and ejection chain operators are recommended and implemented.

Algorithm (5-2) Local Search Algorithm with an Embedded Adaptive Mechanism

Input:
Generate a feasible initial solution (x_0) , μ (initial score of operator), w (initial weight of operator), n (number of iterations in each segment), I (set of operators $\{H_1, H_2, \dots, H_I\}$, nit (number of iterations without improvement), nit_{max} (maximum number of iterations without improvement);

$(x_{act}) = (x_o)$
$\boldsymbol{f_{best}} = f(x_o)$
// Improvement Stage
Initialize the roulette wheel; initialize the adaptive parameters (π, w)
For *seg←1 to n*
Roulette wheel mechanism: Select operator $H \in I$ based on the obtained scores $\{\mu\}$ and weights $\{w\}$
Apply operator H to solution x_{act} , and create solution x_{act}^*
If accept (x_{act}, x_{act}^*)
$x_{act} = x_{act}^*$
End if
If $\boldsymbol{f}(x_{act}) < \boldsymbol{f_{best}}$
$x_{best} = x_{act}$
End if
Update the number of non-improved iterations (nit)
Update the collected scores of operators H
End For
Update weights w of operator H
Set scores $\mu = 0$ of operator H
Output: Report the best solution x_{act} found

5-3-3- Neighborhood selection mechanism

To evaluate the heuristic's performance more accurately, we consider two scenarios. In the first scenario, the neighborhood is chosen through the traditional VND mechanism, and the operation commences with a small case. Once small neighborhoods prove insufficient for improvement, the algorithm transitions to a larger neighborhood size to better explore the heuristic's performance in the solution space. In the second scenario, we employ an adaptive mechanism to understand the heuristic's performance and select neighborhoods based on their previous effectiveness.

An adaptive mechanism is employed to increase the likelihood of selecting a neighborhood with better performance during the search process. Conversely, a neighborhood with poor performance has a reduced chance of being selected. This structure integrates the local search operator into the adaptive mechanism. The search space is further divided into several segments. Initially, within each segment, all neighborhoods carry the same weight, set to one, and the score is initialized to zero. This score is updated iteratively in the following manner.

The scores indicate how well the neighborhood has performed in the current segment. In each iteration, if the considered neighborhood yields a new best solution so far, the score is increased by σ_1. Conversely, if a solution better than the incumbent solution is discovered, the score is increased by σ_2. If a solution worse than the incumbent solution is reached, the score is increased by σ_3. At the end of each segment, the weight is updated based on the following formula:

$$w_{i,j+1} = (1-\gamma)w_{i,j} + \gamma\frac{\mu_{ij}}{\varphi_{ij}} \qquad (5\text{-}23)$$

where $w_{i,j}$ represents the weight of neighborhood i in segment j, μ_{ij} shows the score of neighborhood i in the last segment, and φ_{ij} refers to the number of times neighborhood i is repeated in the last segment.

More importantly, the value of $0 < \gamma$<1 is the parameter for controlling the behaviors of the adaptive mechanism in the proposed algorithm. This value reflects the extent to which weights influence the effectiveness of the neighborhood. When a segment terminates, the new weights are calculated based on the acquired score and all scores are set to zero for the next segment. The probability of selecting the next neighborhood for the operation is determined using the roulette-wheel mechanism as follows: $\frac{w_{ij}}{\sum_{i\in I} w_{ij}}$.

In our algorithm, the segment length is set to 20 iterations, and the optimal values for the mentioned parameters are outlined in Table 5-3.

Compared to the traditional VND, our adaptive model is innovative in four ways: 1) Assigning weights to each neighborhood based on its performance, ensuring that considered operators do not have equal weight; 2) more compatibility with the size of the problem; 3) Introducing diversity to the search space, making it unpredictable which operator will be selected for the next operation; 4) Providing a chance for even bad moves to be selected.

5-3-4- Diversification mechanism

Diversification strategies are employed to escape from local optima by exploring various areas of the search space. They assist in creating a new promising starting point for the improvement phase. The perturbation mechanism is crucial for the metaheuristic's performance. In a basic scenario, the diversification strategy perturbs the current solution by randomly selecting a route, removing stops from it, and reinserting them into a newly generated route, consequently increasing the total travel distance. The diversification mechanism operates through destroy and repair operators.

5-3-4-1- **Destroy operator**

This operator aims to randomly destroy a portion of the solution space. Specifically, a random percentage of routes is selected, and all stops belonging to these routes are removed and inserted into the list M. The percentage is calculated as $\rho \times q$, where ρ is the number of routes generated in local search, and q is the perturbation size. The perturbation size is a critical factor in the algorithm. If the size is small, there's minimal change in the incumbent solution. Conversely, with a large perturbation size, the solution behaves like a random restart, potentially leading to the loss of advantages gained from local optima.

Dynamically setting the perturbation size offers several advantages. Firstly, it can be adjusted dynamically based on the current solution's status within the defined problem size. Secondly, it allows for reduced computing time when a smaller perturbation size is sufficient. Therefore, in this study, significant attention is given to updating the value of q in line with the solutions' performance. The perturbation size follows algorithm 5-3. Initially, the value of q is set to q_{min} and gradually updated based on the successful performance of the destroy-repair operation. If the improvement phase reaches the global best solution so far, q is adjusted to q_{min}, emphasizing the intensification strategy. Conversely, if the improvement phase fails to achieve a global best solution, exploration must continue, and q should be increased. If q reaches q_{max}, it is reset to q_{min}, to avoid reaching an infinite value.

5-3-4-2- **Repair operator**

In the repair phase, new routes are generated from all stops in the list M until it is emptied. This is done using the constructive heuristic described in Section 5.3.1. After applying the destroy-and-repair operator, the new solution is retained and delivered to the improvement phase for further enhancement. It's important to note that the perturbation mechanism aims to generate a new initial solution by diversifying the solution space.

Algorithm (5-3) Perturbation Mechanism

Input:
nit(Number of iterations without improvement), q_{min}(minimum percentage of routes to be removed), q_{max}(maximum percentage of routes to be removed), $f(\boldsymbol{x_{act}^*})$: cost found in the improvement phase, f_{best}: best cost found so far,
ρ : number of routes generated in the last improvement phase, M: list of non-visited stops= {}; $h_{destroy}$: heuristic is used to destroy part of the solution, h_{repair}: heuristic is used to repair part of the solution;

If $\boldsymbol{f(x_{act}^*) = f_{best}}$
$q = q_{min}$
Else
If $q < q_{max}$
$q = q + 10\%$
Else
$q = q_{min}$
End if
End if
Destroy $[\rho * q]$ routes from solution (x_{act}) using $h_{destroy}$ and insert nodes in list M
While $M = 0$
Insert nodes from the list M into the partial solution using h_{repair} and create solution x_{act}
End While
$nit = 0$
Output: Report solution x_{act} found

5-4- Problem generation

As the problem under consideration has not been previously explored, there are no existing instances in the literature. The dataset consists of 77 instances, encompassing garages, schools, stops, and students. The problem involves two trips (shifts). In the first trip, the bus departs from the garage, picks up students, and transports them to their respective schools. In the second trip, students are initially picked up from their schools and then transported to their respective stops (their home locations). The two trips vary in terms of the location where students are picked up.

In the first shift, students are picked up from stops, while in the second shift, the pick-up location is their respective schools. The dataset in this study includes varying numbers of garages (ranging from 1 to 4), stops (from 10 to 110), and schools (from 1 to 11). To facilitate a comprehensive analysis of metaheuristic behavior, three classes of instances S, M, and L are employed. Specifically, set S comprises 21 instances (with stops ranging from 10 to 30), set M contains 28 instances (with stops ranging from 40 to 70), and set L consists of 28 instances (with stops ranging from 80 to 110). All instances are generated and distributed in the Euclidean square between (0,0) and (xmax, ymax), with both xmax and ymax set to 100. The school is dispersed in a rectangle area of (50, 50) pertain to the center of square. Stop discrete have area of (90, 90).

The number of students at each stop is generated and distributed as random variables, ranging from 3 to 5. Given that the problem involves a combination of two shifts simultaneously, some characteristics (e.g., the number of stops and schools) are the same for both shifts, while others are treated differently. These varying characteristics include allowable arrival time at each school in the first shift, allowable picking-up time of students at their schools in the second shift, allowable arrival time at each stop in both shifts, and the number of students.

To allocate students to their respective schools, a simple procedure is followed. Initially, the average number of students per school is calculated. Then, students are assigned to their closest school until the number of allocated students reaches the average level. Once this threshold is reached, students are assigned to the next closest school. This procedure continues until all students are allocated to an accessible school. For each student, time window constraints are considered at both the origin and the destination. To align the data with real-world situations, all students belonging to each school must share the same upper bound for the first shift and the same lower bound for the second shift (all time parameters are in minutes).

For students in the first shift, the lower bound time window is randomly generated within the range of 11:00 a.m. to 11:20 a.m., representing the available time for students at a stop to be picked up. Similarly, the upper bound for the first shift is developed between 12:30 p.m. and 1:00 p.m., indicating the maximum time by which students are expected to be dropped off at their associated schools.

In the second shift, the lower bound time window is randomly generated between 12:40 p.m. and 1:30 p.m., while the upper bound time window is randomly considered between 1:50 p.m. and 2:40 p.m. The former represents the allowable period when students can be picked up from their schools, and the latter indicates the maximum period when students must be dropped off at their respective stops.

It is important to note that students belonging to the first shift must be picked up from their stops after the earliest time, whereas students of the second shift must be dropped off before the latest time. The traveling time between two nodes (stops or schools) is assumed to be the Euclidean distance divided by the speed of the bus (which travels at a constant speed of 20 km/hour). Section 5.6.1 is executed on all instances, while Sections 5.6.2 and 5.6.3 deal with only 30 instances.

5-5- Calibration setting

Before conducting the experimental analysis, it is crucial to comprehend the effect of each parameter on the algorithm's performance. The objective is to test and fine-tune parameters that impact both solution quality and computing time. To efficiently examine this effect, 10 sample instances (discussed in the previous section) are selected and classified into three groups S, M, and L (5 instances from set S, 3 instances from set M, and 2 instances from set L). Tuning is carried out on the instances through a full factorial experiment.

The proposed solution approach relies primarily on three types of parameters, which are as follows. The adaptive layer parameters include $\sigma_1, \sigma_{2,} \sigma_3$, and γ. To control and fine-tune the size of perturbation, two parameters, q_{min} and q_{max}, are introduced. In the construction stage, the parameters are denoted as α, δ, and β.

Each sample is run 10 times, and the average of the results is considered for analysis. The degree of significance (P-value) of each parameter on the solution's performance is measured. Specifically, p-values lower than 0.05 indicate that the given parameter is significant.

The combination of proposed parameters and their respective ranges is presented in Table 5-3. Table 5-4 illustrates the significance of each parameter. It can be observed that both the minimum and maximum percentages of routes supposed to be destroyed, the reaction factor, the scoring parameter (σ_1), and the number of non-improved solutions significantly affect both the quality of the solution and computing time. On the other hand, the parameters σ_2 and $\boldsymbol{\beta}$ only affect the quality of the solution.

Table (5-3) Heuristic Parameters and Best Parameter Setting

Parameter	Description	Value	#	Best value
nit	Number of non-improved solution	10,20,30	3	20
q_{min}	Defines minimum percentage of routes to be removed at perturbation phase	1%,5%,15%	3	5%
q_{max}	Defines maximum percentage of routes to be removed at perturbation phase	20%,25%,30% ,35%,40%,45 50%	7	30%
σ_1	Is scoring parameter in adaptive mechanism	45,55,65	3	55
σ_2	Is scoring parameter in adaptive mechanism	25,35,45	3	35
σ_3	Is scoring parameter in adaptive mechanism	1,5,15	3	5
γ	Is the reaction in adaptive mechanism	0.30,0.4, 0.5,0.60	4	0.5
α	Size of the restricted candidate list	1,2,3,4	4	3
β	Is coefficients used to weigh the partial scores in constructive phase	0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1	11	0.70

Table (5-4) P-Values of the F-Tests Used to Determine the Significance of each Parameter

Parameter	Computing time	Average solution cost
nit	p<0.05	p<0.05
q_{min}	p<0.05	p<0.05
q_{max}	p<0.05	p<0.05
σ_1	p<0.05	p<0.05
σ_2	p>0.05	p<0.05
σ_3	p>0.05	p>0.05
γ	p<0.05	p<0.05
α	p>0.05	p<0.05
β	p>0.05	P<0.05

5-5-1- Effect of the number of iterations on the performance of algorithm

In this stage, we independently investigate the influence of the number of iterations on solution quality and computing time. It is observed that increasing the number of iterations results in a better quality of solution, yet it demands higher computing time. To address this issue, we conduct an analysis to make a trade-off between solution quality and computing time. As in the above section, 10 instances are used for analysis, and the solution is tested with different numbers (100, 200, 300, 500, 700, 900, 1200, and 1500) of iterations. Other parameters are fixed at their best value and are taken from Table 5-3.

Each instance is run 10 times with different numbers of iterations, and the average gap between the best solution found after 10 runs and the best-known solutions is measured. Since no similar study has been carried out on this problem so far, we cannot refer to the best solution from the literature, and therefore, we have relied on our experiments. Thus, the value of the objective function at the maximum number of iterations is set as the best-known solutions, called BKSR.

The obtained results are given in Table 5-5. The first column presents the number of iterations employed, and the best gap percentage is provided in the second column. Ultimately, the total computing time required to solve 10 instances is recorded in the third column. It is seen that while the number of iterations increased from 100 to 200, the value of computing time is augmented by 1.65 times.

This trend keeps continuing for a larger number of iterations, such that by changing iterations from 200 to 300 and 300 to 500, the required computation time rises by 1.33 and 1.51 times, respectively. In spite of the quality of the solution, it is seen that when the number of iterations rises to 500, the best gap percentage changes significantly. At iterations greater than 500, little improvement appears in the best gap percentage, besides the fact that computing time keeps rising. This strongly shows that the capability of the heuristic to find a better solution decreases at higher iterations. It can be inferred that 500 iterations suffice for making a tradeoff between execution time and the quality of the solution.

Table (5-5) Computational Results

Number of iterations	Best gap (%)	Time (millisecond)
100	4.65	858,063
200	3.39	1,415,116
300	2.20	1,881,676
500	1.05	2,840,961
700	0.95	4,552,414
900	0.87	7,008,514
1200	0.79	9,687,359

5-6- Experimental testing

Having obtained the parameters' setting in their best value, we perform an experiment to investigate the performance of the proposed metaheuristic. Our experimental analysis serves three purposes:

1)To investigate the performance of the algorithm on instances as a whole (small, medium, and large); determining the value of the objective function and the computing time.
2) To understand the behavior of different parts of the heuristic.
3)To recognize factors influencing multi-shift loading and mixed-load schemes.

It should be mentioned that 77 instances are considered (ranging from 10 to 110 stops) for the first part; however, for the second and third parts, we study a set of 30 instances (12, 10, and 8 instances from small, medium, and large sets, respectively).

5-6-1- Metaheuristic performance

To assess the proposed mathematical formulation for the SBRP, we carry out computational experiments, including both the exact (integer programming) and metaheuristic approaches. Since the proposed problem is new, we cannot make a comparison with other methods at this stage. SBRP is an NP-Hard problem; thus, by augmenting its size, computing time will increase exponentially. Indeed, we can solve only 16 instances through the exact method, which supports the validity of the proposed metaheuristic in small instances.

Furthermore, we consider two scenarios to explore the behavior of the metaheuristic more accurately. The two scenarios differ only in the structure and order of their employed neighborhoods. Practically, we enable a comparison between the traditional (first scenario) and adaptive methods (second scenario) in the local search block, thereby shedding light on the behavior of the local search algorithm in finding a better solution. Each test instance is run 10 times to avoid the effect of randomness in the results, and the best solution out of these runs is recorded

To make an appropriate comparison between the two scenarios, we set a fixed number of iterations (i.e., 500) for solving each instance. As a result, two percentage gaps are observed, including the percentage gap between the best solution calculated after 10 runs and the exact solution (called the best gap) and the percentage gap between the two kinds of scenarios (addressed as the gap between scenarios). The findings are reported in Appendix 6. The aggregated results for each subset of instances are presented in Tables 5-6 and 5-7. Table 5-6 depicts the best gap for both proposed scenarios. The results indicate that, on average, the second scenario provides a lower percentage gap from the exact solution.

Table (5-6) Results Obtained by Solving Instances in Set S

Metaheuristic	Set S	Set M	Set L
First scenario	2.88%	----	----
Second scenario	1.79%	----	----

Table (5-7) Results Obtained by Solving Instances in Sets S, M, And L

Metaheuristic	Set S	Set M	Set L
Percentage gap between two scenarios	2.05%	3.12%	4.35%

Table (5-8) Total Computing Time of Each Metaheuristic (in Millisecond)

Metaheuristic	Set S	Set M	Set L
First scenario	20,668.8	964,041.5	94,898,574.9
Second scenario	20,946.1	984,456.1	97,693,848.2

To emphasize scenario performance, we compare results across large, medium, and small instances (Table 5-7). In small cases, both scenarios exhibit nearly similar behavior (difference of 2.05%). However, in medium and large cases, the second scenario consistently outperforms the first. On average, the second scenario excels in solution quality, albeit with a slightly longer computing time, possibly attributed to the roulette wheel mechanism (Table 5-8).

5-6-2- Heuristic analysis

In the second stage, an additional investigation is conducted on a set of 30 instances (12 from the small set, 10 from the medium set, and 8 from the large set). This aims to evaluate the impact of the heuristic embedded into the local search on enhancing solution outcomes. At the conclusion of each segment, weights for each local search operator are calculated. For instance, an operator that contributes to a superior solution receives the highest weight, and the heuristic with a greater weight is more likely to be chosen.

Results are presented according to the average weight of each heuristic across all 30 instances. Figure 5-7 illustrates that the "convert m-s" heuristic bears the highest weight, substantially contributing to enhancing the solution space. This may be attributed to the effectiveness of leveraging neighborhood-based problem-specific knowledge as a promising strategy for generating better solutions. The cross-exchange heuristic is ranked third in terms of weight. In conclusion, it is crucial to choose a heuristic aligned with problem-specific knowledge and capable of converting into two or more small operator sizes.

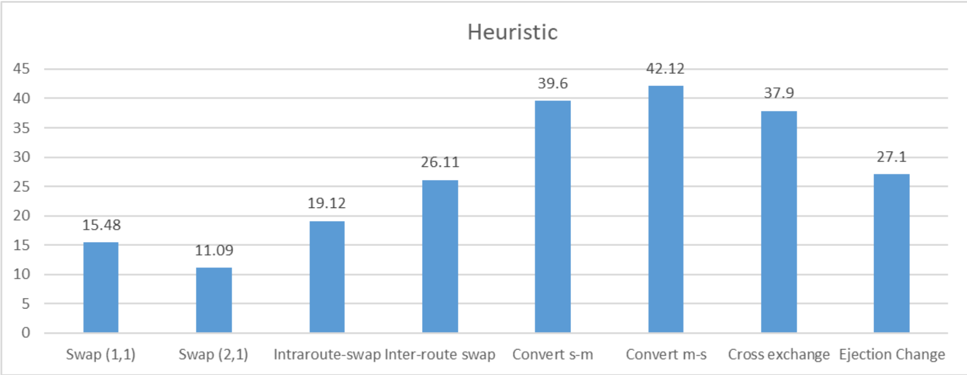


Figure (5-7) Weights of the Heuristics

5-6-3- Practical analysis

This study section explores the impact of parameter variations on the proposed model, focusing on practical implications. This analysis is specifically applicable to stakeholders within the school transportation sector. The goal is to offer a solution and strategy that can yield positive outcomes in terms of both cost reduction and urban traffic management. Challenges like increasing student population density, traffic congestion, and varying school time windows in certain districts have compelled municipalities to take essential steps to address these issues.

In this context, our objective function aims to minimize the total travel time for buses. Within the realm of mixed and multi-shift loadings, a significant concern revolves around minimizing the number of required buses. Thus, we incorporate a parameter for bus reduction percentage in our analysis. To align our model with real-life scenarios, we consider two scenarios: multi-shift loading and no multi-load shifting. This distinction ensures that, in multi-shift loading cases, the commute to and from school can occur simultaneously.

Our analysis is divided into several stages. Initially, the experiment illustrates the impact of time window fluctuations on average riding time constraints. Six scenarios, involving extended time windows, are considered for both with and without the multi-shift loading assumption. Figure 5-8 presents the results, with the vertical axis depicting the percentage of average riding time, and the horizontal axis featuring an extended range of time windows. Notably, as only the time window extends while other parameters remain constant, the average riding time increases.

The expanded time windows allow the bus more time for student loading while adhering to predefined constraints, resulting in an increase in average riding time. Beyond a certain point (after scenario 4), due to student-riding-time constraints, no further reduction occurs in riding time. Consequently, this riding time can increase by a maximum of 12.65% compared to its initial value.

Practically, this discovery assists the municipality in managing three parameters in specific situations: the allowable school time window, the riding time of students, and the total travel time of buses. In both cases, the effect of time window fluctuation on small instances is significantly less pronounced.

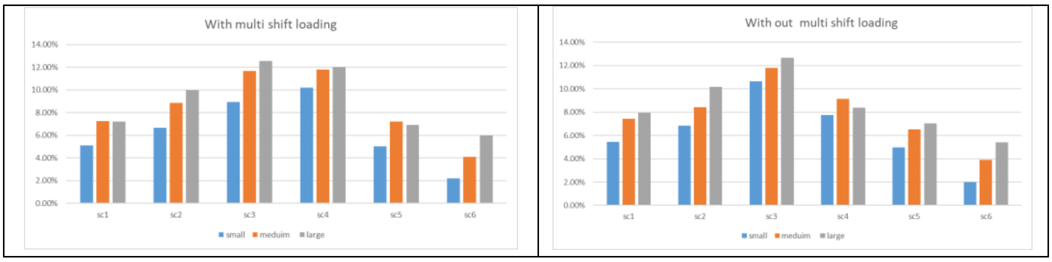


Figure (5-8) Impact of Time Window Fluctuation on Average Riding Time (With and without Multi-Shift Loading)

In the second part of this section, our goal is to examine the impact of the maximum riding point (in 5 scenarios) on both total travel time and the number of available buses. We systematically vary the maximum allowable time when students can be on the bus.

Figure 5-9 illustrates that as the maximum riding time increases, the number of required buses decreases, indicating that the bus has a greater opportunity to load more students. However, as the riding time increases, the total travel time also continues to rise. This conflict between these two parameters requires careful consideration by policymakers, who need to specify the extent to which they want to reduce the number of buses while taking into account the potential increase in total travel time.

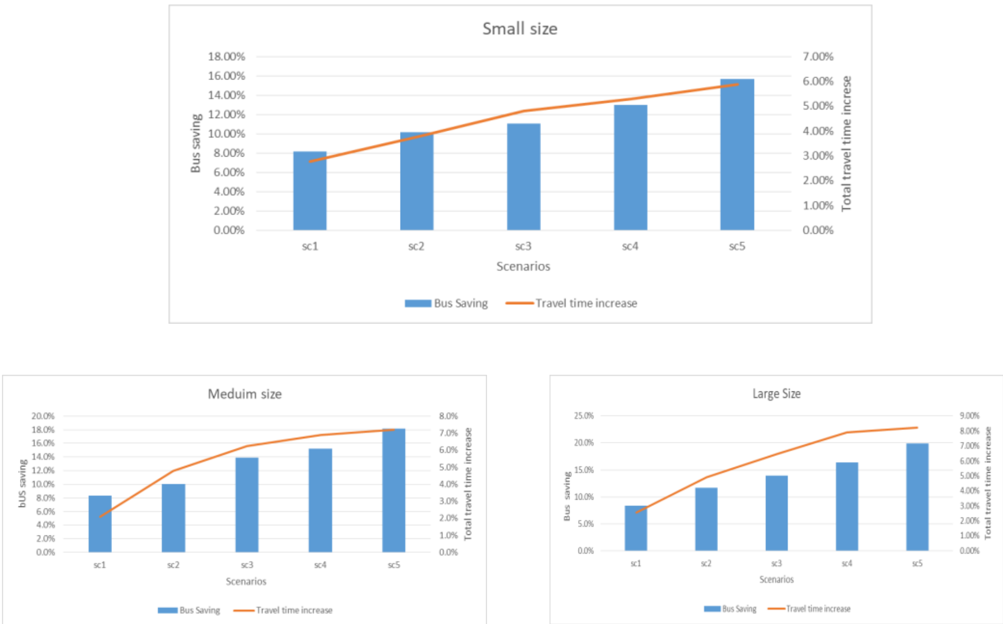


Figure (5-9) Impact of Maximum Riding Point on Total Travel Time by Bus and the Number of Available Buses (in Small, Medium, and Large Instances)

Another experiment (Figure 5-10) depicts the bus occupancy situation across different sample sizes. This finding aids municipalities in optimizing the capacity of their bus fleet (i.e., the number of students in a bus). A municipality seeks to comprehend how buses are occupied as the size of the problem increases, enabling it to raise the number of buses as far as possible.

In comparison, it can be noted that in small instances, there is no significant difference between buses operating with or without multi-shift loading. However, as the sample size increases, multi-shift loading shows improved bus capacity utilization.

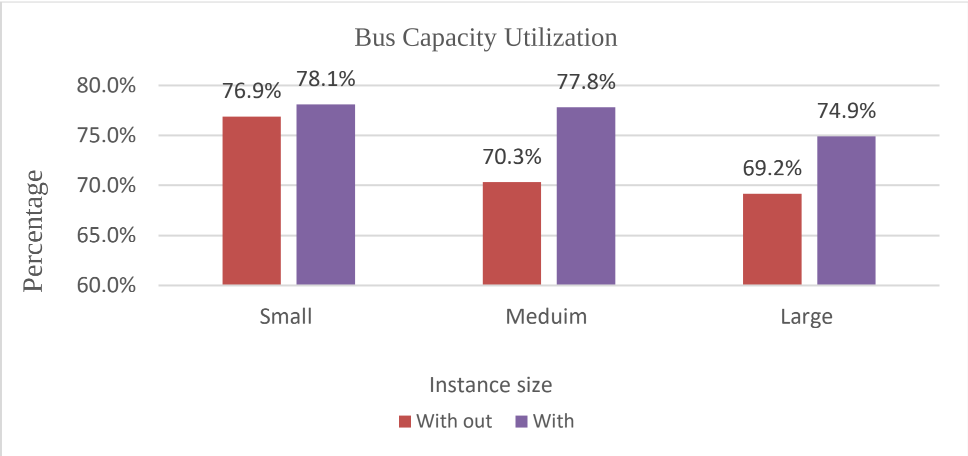


Figure (5-10) Bus Occupation Percentage under Multi-Shift Loading Conditions

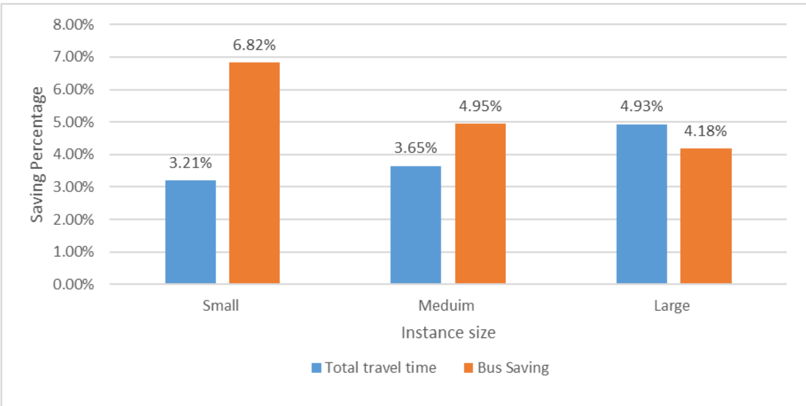


Figure (5-11) Percentage of Saving in the Number of Buses and Total Travel Time while Considering Multi Shift Loading

Our final analysis focuses on the impact of multi-shift loading on both the number of required buses and total travel time, as depicted in Figure 5-11. It is evident that employing a multi-shift loading mechanism leads to a significant reduction in total travel time, particularly in large instances. This indicates that our proposed algorithm enhances efficiency, specifically concerning total travel time, as the size of the problem increases. The key factor is that as the problem size grows, the algorithm can further decrease total travel time, especially when the distance between the dropping and picking points of afternoon students becomes closer. Conversely, the more substantial reduction in the number of buses is associated with small instances. Ultimately, based on the results, the maximum reduction in total travel time and the number of buses is 4.93% and 6.82%, respectively.

5-7- Real case

Tehran, Iran's largest city with 22 districts (refer to Figure 5-12), holds the 24th position in terms of area but claims the top spot in national population. As of 2019, the city boasted 96 municipalities, 8,686 schools, and accommodated 1,662,700 students utilizing either public or private services.

To address these points, we conduct experiments utilizing real data, concentrating specifically on downtown Tehran, encompassing districts 12 and 13, covering an area of 96.91 km² with 481 schools.

To enhance our comprehension of the issue, we perform a distinct analysis for each district, as detailed in Table 5-9. This examination encompasses all schools within both districts. Owing to limitations in data accessibility, our analysis is centered on 35 schools in district 12 and 35 schools in district 13, respectively.

The problem's time window spans from 11:00 a.m. to 2:00 p.m. To tackle the complexity of the issue, we employ a random assignment of bus stops within each of the two districts.

Within this section, our analysis encompasses the identification of which metaheuristic configuration yields superior results. Within this context, we explore two scenarios outlined in Chapter 5.3.3: the traditional and systematic selection of neighborhoods. It is evident that, for both districts (12 and 13), scenario 2 consistently yields, on average, a 3.41% lower solution cost. Additional detailed data is provided in Appendix 7 (districts 12 & 13).

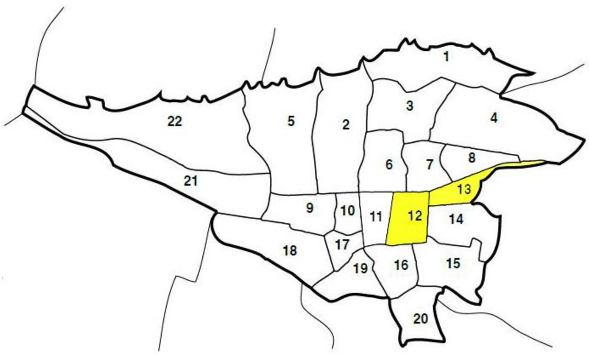


Figure (5-12) Map of Tehran City

Table (5-9) Characteristics of Districts 12 and 13, Tehran

District	Area	No. of school	No. of students (elementary school)	
			Morning shift	Afternoon shift
12	16.91 km^2	288	13,493	8,270
13	80 km^2	193	10,525	7,621

5-8- Conclusions and future research directions

The study makes significant contributions through its incorporation of forward, backward, and split pick-up operations, consideration of multiple garages and schools with varying time windows. Additionally, multi-shift loading is integrated into the model. The problem is not only focused on cost optimization but also considers average riding time and the fluctuation of time windows in the two shifts.

A total of 77 instances are generated for the problem, categorized into three subsets: small, medium, and large. As anticipated, when the problem size increases, the exact method struggles to find an optimal solution within a reasonable time frame after instance 16. To address this, we devised a metaheuristic approach.

The results are organized into three sections as follows:

In the first section, computational experiments are conducted along two lines. Initially, the proposed solution is compared with the exact method in 16 small instances, achieving a best gap below 2.12% (for the second scenario). Following this, two scenarios are developed to elucidate the configuration of the proposed metaheuristic. The concept revolves around comparing neighborhood selection traditionally (based on their size) versus systematically. It is observed that both scenarios exhibit similar behavior in small instances, with their differences becoming apparent as the size of the problem increases.

In the second section, additional analysis is undertaken to explore the impact of the metaheuristic elements on achieving an improved solution. The findings reveal that problem-specific knowledge heuristics play a crucial role in enhancing the solution. This suggests that focusing on neighborhoods that better reflect the problem characteristics proves to be more beneficial.

In the third section, a series of experiments are conducted to address practical concerns, offering valuable managerial insights. Notably, our analysis delves into the impact of imposing maximum riding time constraints on both total travel time and the number of required buses. The findings highlight that while maximum riding time constraints can reduce the number of buses, this reduction comes at the expense of longer total travel time. Therefore, policymakers should consider adopting a trade-off mechanism to effectively minimize both the number of buses and total travel time.

Several issues warrant further investigation in future research. The analysis of heuristics in the context of time characteristics deserves more attention. A potential avenue is the development of an adaptive mechanism to weigh and select each neighborhood in local search, taking into account both computing time and solution quality.

Another line of future research could prioritize realistic concerns, particularly in addressing students' safety in the context of the School Bus Routing Problem (SBRP). This could involve the development of models or strategies that enhance safety considerations in the routing process.

Chapter 6:

Taking a fresh look at safety and health issues in transportation the students to school

6-1- Introduction

Transporting over a million students in Tehran, a city with a population of 9.259[1] million, to and from schools necessitates daily scheduling. The municipality is committed to developing an efficient bus routing plan that considers the diverse locations of students across the city. Safety and health criteria are crucial, addressing key issues in the transportation system at every stage of any proposed plan.

In Tehran, the majority of students rely on public transportation for their school commutes. It is imperative to enhance awareness among students regarding the associated risks of each mode of public transportation. An interview conducted by researchers with urban planning experts reveals that, due to its stringent policies, bus transportation is safer than private cars. These policies should extend to school bus routing, encompassing factors such as the timing of bus stops for picking up or dropping off students, evacuation procedures in case of accidents, and the duration students spend on the bus.

These policies lay the groundwork for establishing rules to safely transport students to their schools. Technically, this falls under road traffic safety. School buses boast special safety features such as flashing red lights, cross-view mirrors, and stop-sign arms. They also adhere to high crush standards and have protective seating. The system proposed by the French National Council of Transport (CNT) considers multiple criteria for ensuring bus stop safety, including the route to the stop, pedestrian crossings, driver visibility, stop size and position, quality of the waiting area, and imposed maneuvers.

Another noteworthy approach in this field is the model introduced by the Swedish Transport Agency. This model assesses the risk of accidents and the level of insecurity for students waiting at bus stops or en route to them. It categorizes bus stops into four types based on design and waiting area.

Safety considerations are closely linked to factors such as the vehicle type (e.g., bus type), waiting/walking area, and local conditions (such as snow restrictions) (Chalkia et al., 2016).

In the school bus routing problem, a crucial aspect is the selection of capable and physically healthy drivers. These individuals should possess not only impeccable driving skills but also the ability to communicate safety tips to students effectively.

Additional complexities in the school bus routing problem arise from factors such as depreciation and technical defects in the bus fleet, demanding careful consideration. Parents express heightened worry over school bus accidents, particularly when

[1] https://worldpopulationreview.com/world-cities/tehran-population

technical issues are the cause. Parents will be promptly informed of any inconvenience through immediate notifications and written communication mechanisms.

A significant portion of Tehran's minibusses is worn out and fails to meet essential safety standards for student transportation. In the 2017-2018 period, the Deputy of Transportation and Traffic of Tehran Municipality mandated valid technical inspections for all school vehicles. Fleets failing this test are prohibited from providing services.

Currently, the enforcement of the rule mandates accreditation for all student transport fleets through technical inspection centers. An interview with an expert reveals that the majority of minibuses in Tehran are worn out and require specific technical inspections, highlighting their unsuitability for student transportation. To address this issue, the public-school transport system is collaborating with the private sector, such as the taxi organization, to introduce new types of vehicles. These aspects, emphasized in previous studies, contribute to alleviating parental concerns.

Crucially, to enhance efficiency in meeting students' safety and health requirements, it is essential for families and the school committee to give special attention to these issues. This collaboration results in a decrease in the number of accidents.

Families should instruct their children to wait in a safe place, like a sidewalk, and open the bus door only after ensuring the vehicle has stopped. It's important to remind children to listen to the driver's advice, stay seated while the vehicle is in motion, and avoid speaking loudly to prevent distractions. Adhering to these simple yet effective rules can reduce the risks in student transportation. Besides family efforts, if the municipality and relevant communities rigorously follow their established rules, including providing necessary resources, the occurrence of accidents can markedly decrease.

The transportation network in Tehran comprises three types of traffic zones:

Central restricted zone: Only public transport (bus, taxi, ambulance, etc.) is permitted in this zone.

The odd-even traffic scheme zone, which has been recently converted into a low emission zone (LEZ)[1]. This plan limits private vehicle access based on the last digit of their license plate, allowing them to move on alternate days. The maximum allowed entry for private vehicles is 20 working days per season, as per the newly approved bylaw aimed at reducing air pollution.

Free zone: Outside the central restricted and even-and-odd traffic plan areas, this zone allows the use of all kinds of vehicles. Implementing these restrictions helps control vehicle passage in crowded areas.

1 https://bimeh.com/mag/air-pollution-control-plan/

The widespread presence of the coronavirus demands heightened focus on students' health and safety. An alarming report from Iran showed a case fatality rate of 5% from the start of the pandemic until April 2021, the highest compared to other countries globally. This underscores the necessity of taking appropriate measures to guarantee students' safety as they return to school post-pandemic. It emphasizes the urgency of establishing an effective school transport system that prioritizes predefined safety and health considerations.

To achieve these objectives, it is crucial to create a safety map for school bus routing and develop appropriate guidelines.

To establish appropriate guidelines, it is crucial to thoroughly investigate items that could have a negative impact on safety and health. Taking action to control and mitigate these factors is consistent with the principles of risk assessment.

Properly identifying risks is essential, followed by implementing necessary actions to assess and mitigate them. In a meticulous risk analysis of student transportation, various factors, including traffic, population density, and health, are considered.

Risk assessment is a valuable method for evaluating safety levels in a complex and modern system. In practical terms, risk refers to the likelihood that something unfavorable will occur and the consequences of such an event[1].

Current studies predominantly focus on applying risk assessment in urban transportation, making it widely accepted in the industry. Despite this knowledge, risk analysis for student transportation is in its early stages and hasn't been fully integrated with the mixed-load and location-allocation concept in School Bus Routing Problem (SBRP). This chapter introduces a developed and implemented risk assessment model, considering various features. These features are derived from a combination of field observation and expert interviews.

The process involves identifying potential risks, followed by prioritization to determine their significance. The results are used as a basis for the proposed model.

In summary, this chapter makes four key contributions:

1)Introducing a risk assessment method to identify and score risks impacting students' health and safety negatively.

2) Prioritizing the more significant risks and incorporating them into the model.

3) Proposing a metaheuristic that explores diversification in the solution space.

4) Analyzing the overall transportation cost while relaxing risk constraints in the model.

[1] https://www.riskassess.com.au/docs/SISch2_5.00.pdf

6-2- Research method

This research method follows four sequential steps. First, it identifies and classifies main risks and sub-risks. In the second step, the score for each identified risk is calculated based on probability and impact level, followed by risk prioritization through pairwise comparisons. The third stage involves feeding the risks with higher weights into the model, prioritizing the more significant risks. Finally, the proposed model is solved using an exact method for small instances and metaheuristic configurations for all instances.

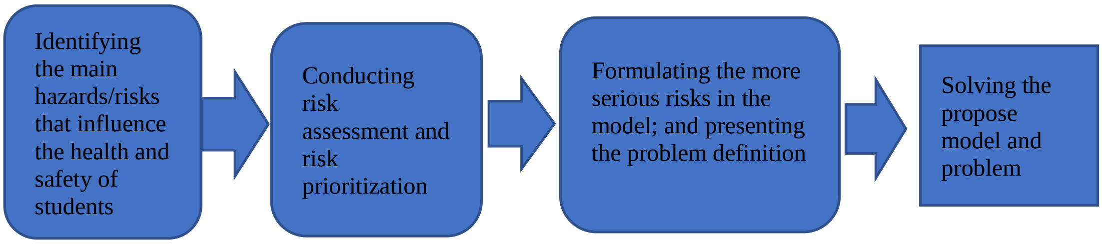


Figure (6-1) Research Methodology

In the initial stage, a method is developed to identify and classify risks. The focus is on recognizing hazardous conditions resulting from defects, interaction, and interference of factors deviating from desired objectives. Proper risk identification is vital in the risk assessment process, as its neglect can hinder performance in subsequent stages.

Risk identification can be accomplished systematically, experimentally, or innovatively. In this study, a combination of field observation and expert interviews is employed. Data for the survey is collected from 12 urban management experts (as key informed people) strategically distributed across different regions of the city. Tehran is segmented into three districts (north, south, and center), with four experts interviewed in each district, ensuring a comprehensive coverage of relevant expertise.

We are considering the implementation of these new district types (north, south, and center) to ensure adequate coverage of geographical divisions and proper distribution of students.

The researcher visited various city sections, such as bus stations, observed critical issues, and compiled the initial list of hazards. The final list is created by amalgamating experts' insights with the researcher's observations. A total of 12 risks are identified and further categorized into three groups for clarity: safety, health, and traffic.

To select the most significant risks, a risk assessment is necessary, involving scoring and trade-offs between identified risks to determine their respective weights (risk prioritization). Risks with higher weights are then chosen for further analysis. Risk prioritization is conducted using the Analytic Hierarchy Process (AHP) technique. AHP serves as a foundation for effective solutions in complex decision situations, streamlining and expediting the decision-making process.

AHP is a method designed to break down complex unstructured situations into simpler components, creating a hierarchical system problem. Developed by Saaty in 1987, the method has undergone extensive consideration and refinement. The AHP process involves three stages: 1) Structuring the matrix of judgment between each pair of criteria, 2) Making pairwise comparisons between the criteria and assigning priorities to each, and 3) Normalizing the paired comparison matrix to ensure uniform units across all criteria. Obtaining priorities for criteria involves performing pairwise comparisons for each criterion, determining the percentage importance of items (priority vector) for the Analytic Hierarchy Process (AHP).

To achieve this, 4 urban management experts are employed to score and weigh the risk factors, and the average weight is then inputted into the AHP for analysis.

More precisely, we employed 12 urban management experts to identify the initial risk, and subsequently narrowed it down to four experts from the initial group to prioritize the most critical risks.

Considering that the perspectives of a subset of experts, referred to as key informed people, can serve as a representation of the entire expert community, four experts are involved in this context.

Each analyst is tasked with answering two questions: 1) What is the score of each risk based on its impact and probability level? 2) Which criterion is more important and stronger in pairwise comparison?

For each identified risk, the score is calculated by multiplying the probability by the severity of the risk in each district. Finally, the average score is computed. The severity value for every identified hazard remains consistent across all districts, as it is based on the consensus of expert opinions.

The consolidated results of the risk assessment matrix and AHP are presented in the following. Appendix 8 displays the independent score and average score (across three districts) for each risk, while Table 6-1 illustrates the weight of each risk derived from pairwise comparison.

In the realm of safety, the findings highlight that the size of the bus stop holds the utmost importance. Concerning health, the results point to the highest weight being attributed to population density and the prevalence of coronavirus. Lastly, in terms of traffic, the highest weight is associated with traffic volume.

It is noteworthy, as observed in Appendix 8, that population density is higher in central and south district; traffic volume is greater in the center and south due to the concentration of businesses and government departments; and the location of bus stops is a common issue across most parts of Tehran.

The household income can influence the safety risk of student transport in several ways. Lower-income families may face challenges affording residences in neighborhoods with quality schools, resulting in longer commutes for their children. This extended travel duration raises the likelihood of transportation-related accidents or incidents. Moreover, lower-income households tend to reside in areas characterized by higher traffic volumes and less secure road infrastructure, with fewer sidewalks and crosswalks. This heightened exposure to traffic increases the risk of accidents and injuries for students walking or biking to school.

Table (6-1) Results of AHP

Row	sub-criteria	Weight
1	The route leading to bus stop	6.86%
2	Pedestrian crossing	8.59%
3	The location of bus stop	7.28%
4	Size of waiting area	15.70%
5	Quality of place in waiting area	8.38%
6	Density of population	9.37%
7	prevalence to corona virus	17.28%
8	Household Income	6.35%
9	Complex intersection	3.47%
10	High traffic volume	10.97%
11	Traffic speed	2.28%
12	Highway area	3.45%

6-3- Problem description and mathematical model

The SBRP explored in this chapter simplifies the well-known Vehicle Routing Problem (VRP). It involves a set of schools, one type of students, and a collection of garages and identical buses—each with a fixed capacity. Students are assigned to bus stops based on their maximum allowable distance. Subsequently, the bus departs from the garage, picking up the assigned students, and proceeds to the designated schools.

Given the mixed-load effect, the model facilitates transporting students from different schools in the same bus for practicality. To further enhance efficiency, the problem is confined to three regions (north, center, and south), and risk analysis is then carried out. Following this analysis, the model is customized to address the districts with a high-risk score.

The objective is to optimize the total travel time, and key assumptions for our problem include:

1)Each bus starts from the garage to pick up students from the stop and drops them off at their respective schools.

2) Students are allocated to their respective stops while ensuring the maximum walking distance is met.

3) Safety (size of bus stop), health factors (prevalence of coronavirus and population density), and traffic concerns (traffic volume) are integrated into the model.
Specifically, the problem involves considerations for traffic volume and population density on the arcs and health and safety issues (prevalence of coronavirus and size of bus stop) at the nodes, corresponding to each zone.
4)An upper time window is established for each school, ensuring that a bus arrives at the school and drops off the student before the latest time window.
5) A lower time window is defined for each stop, ensuring that a bus begins its service to pick up the student after the stop's earliest time.
6) The model enforces a maximum limit on the number of students allocated to each stop. This limit must not be exceeded.

Table (6-2) Indices, Sets, Parameters, and Decision Variables used in the Mathematical Model

Indices	
k	Bus index
i, j	Node indices
l	Student index
Sets	
G	Set of starting and ending depot locations (garage locations)
K	Set of buses
S	Set of students
P^{+}	Set of potential pickup locations (bus stop locations)
P^{-}	Set of delivery locations (school locations)
$P = P^{-} \cup P^{+}$	Set of stops and schools
$N = P \cup G$	Set of nodes
Parameters	
c	Bus capacity
$big\ M$	Large constant
a_i	Earliest arrival time at stop $i \in P^{+}$
b_i	Latest arrival time at school $i \in P^{-}$
ap	Average pickup time at pickup points for each student
ad	Average delivery time at delivery points for each student
C_{ij}	Travel distance from node i to node j $(i, j \in N)$
t_{ij}	Travel time from node i to j $(i, j \in N)$ The travel time from node i to node j is determined by dividing the travel distance between the two nodes by the speed of the bus.
s_{il}	A parameter equal to 1 if student l can reach stop $i \in P^{+}$, and 0 otherwise
q_{il}	A parameter equal to 1 if student l is related to school $i \in P^{-}$, and 0 otherwise
P_g	Number of parking spaces at garage g
ms	Maximum number of allowable students for each stop
$O_i = \{S \mid s_{il} = 1\}$	Set of students who can be assigned to stop i $(i \in P^{+})$
$W_i = \{S \mid q_{il} = 1\}$	Set of students who should be delivered to school i $(i \in P^{-})$
Tr	Risk threshold coefficient
Hr_i	Health risk parameter in node i
Pdr_{ij}	Density-of-population risk parameter from node i to node j
Tv_{ij}	Traffic volume risk parameter from node i to node j
Ar_{ij}	Total risk parameter from node i to node j: sumation of Pdr_{ij} and Tv_{ij}
Decision variables	
X_{ijk}	1 if bus k traverses the arc from node i to $j (\forall i, j \in N)$, and 0 otherwise
y_{ik}	1 if bus k visits stop i, 0 otherwise
Z_{il}^{k}	1 if student l is picked up by bus k from stop i, and 0 otherwise
T_{ik}	Arrival time of bus k to node $i (\forall i \in N)$
L_{ik}	Load of bus k after leaving node i $(\forall i \in P)$
R_{ik}	Value of the risk index for bus k when it arrives at node i $(\forall i \in N)$
h_{ik}	1 if bus k visits school $i \in P^{-}$, and 0 otherwise
D_{jl}^{k}	1 if student l is delivered by bus $k$to school j, and 0 otherwise

The mathematical programming formulation of the school bus routing problem is as follows:

$$Min \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} t_{ij}X_{ijk} \quad (6\text{-}1)$$

S.t.

$$\sum_{j\in N} X_{jik} = \sum_{j\in(P^+\cup P^-)} X_{ijk} = y_{ik} \qquad \forall i \in P^+, k \in K \quad (6\text{-}2)$$

$$\sum_{j\in(P^+\cup P^-)} X_{jik} = \sum_{j\in N} X_{ijk} = h_{ik} \qquad \forall i \in P^-, k \in K \quad (6\text{-}3)$$

$$\sum_{i\in G}\sum_{j\in P^+} X_{ijk} \le 1 \qquad \forall k \in K \quad (6\text{-}4)$$

$$\sum_{j\in P^-}\sum_{i\in G} X_{jik} \le 1 \qquad \forall k \in K \quad (6\text{-}5)$$

$$\sum_{i\in G}\sum_{j\in G} X_{ijk} = 0 \qquad \forall k \in K \quad (6\text{-}6)$$

$$\sum_{k\in K} y_{ik} \le 1 \qquad \forall i \in P^+ \quad (6\text{-}7)$$

$$\sum_{k\in K} Z_{il}^k \le s_{il} \qquad \forall l \in S, j \in P^+ \quad (6\text{-}8)$$

$$Z_{il}^k \le y_{ik} \qquad \forall l \in O_i, i \in P^+, k \in K \quad (6\text{-}9)$$

$$y_{ik} \le \sum_{l\in S} Z_{il}^k \qquad \forall i \in P^+, k \in K \quad (6\text{-}10)$$

$$\sum_{k} D_{jl}^k \le q_{jl} \qquad \forall l \in S, j \in P^- \quad (6\text{-}11)$$

$$D_{jl}^k \le h_j^k \qquad \forall l \in W_j, j \in P^-, k \in K \quad (6\text{-}12)$$

$$h_{jk} \le \sum_{l\in S} D_{jl}^k \qquad \forall j \in P^-, k \in K \quad (6\text{-}13)$$

$$\sum_{i\in P^+} Z_{il}^k = \sum_{j\in P^-} D_{jl}^k \qquad \forall l \in S, k \in K \quad (6\text{-}14)$$

$$\sum_{i\in P^+}\sum_{k\in K} Z_{il}^k = 1 \qquad \forall l \in S \quad (6\text{-}15)$$

$$\sum_{l\in S} Z_{il}^k \le ms \qquad \forall i \in P^+, k \in K \quad (6\text{-}16)$$

$$L_{ik} = 0 \qquad \forall i \in G, k \in k \quad (6\text{-}17)$$

$$L_{ik} + \sum_{l\in S} Z_{jl}^k Hr_i \le L_{jk} + bigM\left(1 - X_{ijk}\right) \qquad \forall i \in P, j \in P^+, k \in k \quad (6\text{-}18)$$

$$L_{ik} - \sum_{l\in S} D_{jl}^k \le L_{jk} + bigM\left(1 - X_{ijk}\right) \qquad \forall i \in P, j \in P^-, k \in k \quad (6\text{-}19)$$

$$\sum_{l\in S} Z_{il}^k Hr_i \le L_{ik} \le C \qquad \forall i \in P^+, k \in k \quad (6\text{-}20)$$

$$T_{ik} + ap.\sum_{l\in S} Z_{il}^k + ad.\sum_{l\in S} D_{il}^k + t_{ij} \le T_{jk} + bigM\left(1 - X_{ijk}\right) \qquad \forall i \in P, j \in P, k \in K, i \ne j \quad (6\text{-}21)$$

$$T_{ik} + t_{ij} \le T_{jk} + bigM\left(1 - X_{ijk}\right) \qquad \forall i \in G, j \in P^+, k \in K \quad (6\text{-}22)$$

$$T_{ik} \le T_{jk} + bigM\left(1 - Z_{il}^k\right) \qquad \forall i \in P^+, j \in P^-, l \in S, k \in K \quad (6\text{-}23)$$

$$T_{ik} \ge a_i - (1 - y_{ik})bigM \qquad \forall i \in P^+, k \in K \quad (6\text{-}24)$$

$$T_{ik} \le b_i + (1 - h_{ik})bigM \qquad \forall i \in P^-, k \in K \quad (6\text{-}25)$$

$$\sum_{i\in P^-}\sum_{k\in K} X_{ijk} \le Pg \qquad \forall j \in G \quad (6\text{-}26)$$

$$R_{ik} = 0 \qquad \forall i \in G, k \in k \quad (6\text{-}27)$$

$$R_{ik} + Ar_{ij}C_{ij}X_{ijk} \le R_{jk} + bigM\left(1 - X_{ijk}\right) \qquad \forall i \in G, j \in P^+, k \in k, i \ne j \quad (6\text{-}28)$$

$$R_{ik} + Ar_{ij}C_{ij}X_{ijk} \leq R_{jk} + bigM(1 - X_{ijk}) \quad \forall i \in P, j \in P, k \in k, i \neq j \quad (6\text{-}29)$$

$$R_{ik} \leq Tr \quad \forall i \in P, k \in k \quad (6\text{-}30)$$

$$y_{ik} \in \{0,1\} \quad \forall i \in P^{+}, k \in K \quad (6\text{-}31)$$

$$X_{ijk} \in \{0,1\} \quad \forall i, j \in N, i \neq j, k \in K \quad (6\text{-}32)$$

$$Z_{il}^{k} \in \{0,1\} \quad \forall i \in P^{+}, l \in S, k \in K \quad (6\text{-}33)$$

$$D_{jl}^{k} \in \{0,1\} \quad \forall j \in P^{-}, l \in S, k \in K \quad (6\text{-}34)$$

$$h_{ik} \in \{0,1\} \quad \forall i \in P^{-}, k \in K \quad (6\text{-}35)$$

$$r_{lk} \in \{0,1\} \quad \forall l \in S, k \in K \quad (6\text{-}36)$$

The objective function (6-1) minimizes the total travel time spent by all buses. Constraints (6-2) require that a bus entering the stop node should leave it as well. In constraints (6-3), the same constraints for school node are exposed. Constraints (6-4) ensure that a bus cannot start more than once from its home location (here, garage). Similarly, constraints (6-5) specify that it is not allowed for a bus to arrive at its final location (garage) more than once, which implies that some buses could remain unused. Constraints (6-6) specify that direct transfer from garage to garage is not possible. Constraints (6-7) enforce that each stop is visited no more than once. Constraints (6-8) enforce that each student is taken from the stop to which he/she walks. Constraints (6-9) represent that picking up a student from a non-visited stop by bus k is not possible. Constraints (6-10) guarantee that stops are not visited unnecessarily. Constraints (6-11) ensure that each student is delivered to his/her respective school. Constraints (6-12) guarantee that whenever a student is assigned to a bus, the school associated with this student is also visited by the same bus. Constraints (6-13) guarantee that schools are not visited unnecessarily. Constraints (6-14) impose that the number of pickup and delivery students in each route is equal. Constraints (6-15) state that each student should be picked up exactly once. Constraints (6-16) ensure that number of students allocated to each allowable stop must not be more than its range. The next four sets of constraints (6-17), (6-18), (6-19) and (6-20) are load constraints. Thus, constraints (6-18) state that when a node i is followed by a pickup node $j \in P^{+}$, the number of students after visiting node j is greater than or equal to the summation of the number of students after servicing node i and the number of students picked up in node j. Similar to constraints (6-18), inequality (6-19) proves that when node i is followed by delivery node $j \in P^{-}$, the number of students after visiting node j is greater than or equal to the number of students after visiting node i minus the number of students delivered to node j. In practice, constraints (6-18) and (6-19) determine the load on a bus only after it leaves each node on its route. Constraints (6-20) specify the capacity of buses. Constraints (6-21) -(6-25) are time-related constraints. The arrival time of each bus to a node in p is calculated according to constraints (6-21). Constraints (6-22) are similar to constraints (6-21), but they are designed for the routes from garage to stop. Constraints (6-23) ensure that students are picked up by a bus before they are delivered. Constraints (6-24) and (6-25) indicate the time window for stops and schools, respectively. Constraints (6-26) restrict the number of available parking places in each garage. Constraints (6-27) – (6-30) ensure that the global risk is at most equal to the risk threshold Tr. Finally, variables and their types are presented in constraints (6-31) -(6-36).

The health risk constraints are applicable to (6-18), (6-19), and (6-20). Ar risk constraints, which involve the combination of traffic volume risk and density of population risk constraints, are incorporated for constraints (6-27) and (6-30).

6-4- Solution methodology

Given that the School Bus Routing Problem (SBRP) is established as an NP-hard problem, a metaheuristic approach is necessary for handling large instances. It's crucial to elucidate the design of a metaheuristic capable of incorporating both diversification and intensification strategies and how to examine this behavior throughout the solution space.

In practice, achieving an efficient algorithm requires an appropriate trade-off between intensification and diversification mechanisms. Addressing concerns related to neighborhood structure and exploration mechanisms is crucial in this context. Undoubtedly, selecting the correct order of operators contributes positively to minimizing computing time on the path to reaching the optimal solution.

Hence, selecting the correct order of neighborhoods in the search space is crucial. More importantly, there is a need to strike a reasonable compromise between the solution's quality and the algorithm's computing time. Emphasizing neighbors that ensure high quality might be desirable but comes at the cost of increased computing time. Conversely, concentrating on smaller neighbors can impede the perturbation effect in the solution space, thereby diminishing the impact of exploration.

Currently, the adaptive mechanism of neighborhood selection stands out as a well-known approach in various routing problems. However, recent work by Turkeš et al. (2021) sheds light the extent to which the Adaptive Large Neighborhood Search (ALNS) can enhance heuristic performance. Interestingly, they demonstrate that the advantage of using the adaptive selection mechanism is not significant. Their recommendation is that researchers, when opting for ALNS, should follow a specific set of actions instead of blindly copying the algorithm from other studies. This underscores the importance of identifying elements that enhance the algorithm's performance during its design.

To achieve good performance, it is essential to carefully consider the following issues and implement them if feasible.

1) Rewarding the metaheuristic based on both its performance and execution time.

2) Paying careful attention to the difference made in the objective function during the execution of each operator.

3) Allowing the selection of operators with poor performance, even with a low probability.
4) Weighing the neighborhood-based degree of intensification and diversification in the solution space.
5) Considering the time interval between the execution of each operator in the solution space.

In numerous studies, operators are often weighted according to their past iteration performance. As per Turkeš's findings, relying solely on the adaptive mechanism may not ensure the discovery of good solutions, emphasizing the need for additional considerations. Therefore, the crucial task is to establish a well-defined structure for selecting both the order of local search operators and various approaches to exploration.

It is crucial to conduct a metaheuristic analysis of the diversification mechanism in the solution space, the type of neighborhoods (whether well-known or more specified), and the role of each neighborhood in the algorithm's performance (whether in time or cost).

To address this, regarding the diversification we propose different metaheuristic configurations:

- Multi-start structure (m)
- Perturb and improve structure (p)

Regarding the selection of operators in the improvement phase, two kinds of neighborhood structures are considered:

- Traditional order which follows a fixed order of neighborhoods.
- Systematic order (selection based on adaptive mechanism) which adheres to systematic order of neighborhoods.

In total, we suggest four kinds of metaheuristics:
1) Multi-start structure with a fixed order of neighborhoods (m-VND),
2) Multi-start structure with a systematic order of neighborhoods (m-HA),
3) Perturbation structure with a fixed order of neighborhoods (p-VND), and finally
4) Perturbation structure with a systematic order of neighborhoods (p-HA).

In both p-VND and p-HA metaheuristics, the initial solution is constructed during the first phase. Subsequently, the constructed solution undergoes the improvement phase (second phase) until a local optimum is reached.

Finally, the diversification heuristic is applied to escape the current solution (the stuck solution) from the local optimum. The diversification stage comes into play when the solution becomes trapped in a local optimum.

Simply put, this structure involves using the construction phase once and repeating the improvement phase from the perturbed solution for a specified number of iterations. On the contrary, in both m-VND and m-HA metaheuristics, the initial solution is constructed in the first phase and is carried over to the second phase for further improvement. If local optima are encountered, the algorithm restarts again from the initial solution to achieve more diversification. The algorithm iterates through both the construction phase and the improvement phase until reaching the maximum specified number of iterations.

The primary distinction between this VND and HA configuration lies in the way each selects neighborhoods: the HA metaheuristics organize neighborhoods based on their performance, whereas the VND metaheuristics adhere to a fixed order, progressing from small to large neighborhoods.

In VND metaheuristics, all operators are assessed using the first-improvement strategy, accepting all feasible movements that enhance the current solution. The neighborhoods are explored in order from small to larger (more complex) ones. If improvement occurs, the solution initiates intensification from the first operator. This process concludes when local search can no longer be enhanced by any of the operators.

Contrastingly, in HA metaheuristics (adaptive mechanism), operators are chosen based on their performance. Consequently, after executing a move, regardless of whether the solution improves or worsens, the next operator is selected using the roulette wheel mechanism.

For a more thorough exploration analysis, we introduce two diversification configurations termed multi-restart and perturbation mechanisms. The structures of multi- and perturbation heuristics, coupled with the adaptive layer, are outlined in Algorithm 6-1 and 6-2, respectively. The allocation of students to bus stops is executed in the construction phase, and the improvement phase, involving the student allocation sub-problem, is carried out to verify the feasibility of the solution (refer to Chapters 2 and 3 for further details).

Algorithm (6-1) The Proposed Algorithm (P_HA)

Input: U (set of all potential stops), G (set of all garages), P^{-} (set of all schools), s (set of all students), I (set of operators), q (percentage of routes to be destroyed), P^{+}(List of stops to which students are allocated), μ (initial score of operator (H)), w (initial weight of operator (H)), nit (number of iterations without improvement), It (total number of iterations), *n* (number of iterations in each segment), and nit_{max} (maximum number of iterations without improvement);ε (percentage of stops to be removed)

Stage 1: Construction phase

P^{+}= List of stops to which students are allocated // Student allocation
$x_o = route\ generation(\ p^{+},\ p^{-}$,s, G) // Generating routes via the constructive heuristic
$x_{best} = x_o$
$f_{best} = f(x_o)$
$x_{act} = x_o$
Stage 2: Improvement phase
Initialize the roulette wheel; initialize the adaptive parameters (μ, w)
  While Stopping criterion It is not met
    For *seg←1 to n*
      Roulette wheel mechanism: Select **one** operator $\boldsymbol{H} \in \boldsymbol{I}$ through the adaptive mechanism
      x^{*}_{act}= H (x_{act}) // improve the solution by applying the selected improvement
       If accept (x^{*}_{act}, x_{act})
         $x_{act} = x^{*}_{act}$
       End if
           If $f(x_{act}) < f(x_{best})$
             $x_{best} = x_{act}$
           End if
              Update the collected scores (μ) on operator ***H***
              Update the number of iterations without improvement (nit)
              If max number of iterations without improvement reached
Stage 3: Perturbation phase
                Update the parameter (q, ε)
                x_{act}= Perturb (x_{act}, q, ε) by applying the perturbation neighborhood
                $nit = 0$
              End if
    End for
          Update the weight (w) of operators
          Set $\mu = 0$ for all operators
  End while
Return x_{best};

Algorithm (6-2). The Proposed Algorithm (m_HA)

Input: P^+ (set of all potential stops), G (set of all garages), P^- (set of all schools), s (set of all students), I (set of operators), q (percentage of routes to be destroyed), P^+(List of stops to which students are allocated), μ (initial score of operator (H)), w (initial weight of operator (H)), nit (number of iterations without improvement), It (total number of iterations), *n* (number of iterations in each segment), and nit_{max} (maximum number of iterations without improvement);

While the stopping criterion It is not met
  Stage 1: Construction phase
    P^+= List of stops // Student allocation
    $x_o = route\ generation(\ p^+,\ p^-$, s,G) // Generating route via the constructive heuristic
    $x_{best} = x_o$
    $f_{best} = f(x_o)$
    $x_{act} = x_o$
  Stage 2: Improvement phase
    While stopping criterion nit_{max} $is\ not$ reached
    Initialize the roulette wheel; initialize the adaptive parameters (μ, w)
      For *seg←1 to n*
        Roulette wheel mechanism: Select **one** operator $\boldsymbol{H} \in \boldsymbol{I}$ through the adaptive mechanism
        x_{act}^*= H (x_{act}) // improve the solution by applying the selected improvement
        If accept (x_{act}^*, x_{act})
          $x_{act} = x_{act}^*$
        End if
          If $f(x_{act}) < f(x_{best})$
            $x_{best} = x_{act}$
          End if
        Update the collected scores (μ) on operator ***H***
        Update the number of consecutive iterations without improvement (nit)
      End for
        Update the weight (w) of operators
        Set $\mu = 0$ for all operators
    End while
  End while
Return $\boldsymbol{x_{best}}$**;**

6-4-1- Construction phase

The construction phase unfolds sequentially in three main steps:

1) Student allocation step: Allocating students to potential bus stops.

2) Stop allocation step: Assigning the allowable bus stop to available garages, preferably the closest garage.

3) Route determination step: Establishing the route between potential bus stops.

In the first stage, each student is assigned to the nearest bus stop, ensuring the observance of maximum walking distance and safety.

The safety consideration mandates that the number of allocated students to each allowable bus stop should not surpass a predefined value. The mechanism for assigning students to allowable bus stops is comprehensively detailed in the study by Fallah et al. (2017).

In the second step, stops are assigned to the closest garages until the total number of allocated stops reaches the threshold. For each instance, the threshold value is determined by dividing the total number of available stops by the number of available garages. If the number of assigned stops exceeds this threshold for a particular garage, the stop is subsequently assigned to the next closest garage. This process continues until all eligible stops (those to which students are already assigned) find their respective garages.

In the third step, routing is conducted using a Greedy Randomized Adaptive Search Procedure (GRASP). GRASP is employed to overcome the myopic behavior of the greedy heuristic, striking a balance between intensification and diversification.

In the routing procedure, there is a focus on two key aspects when inserting non-visiting nodes: the generation of new routes from garages and, correspondingly, the addition of non-visited stops to the existing route. For each new route originating from the garage, all non-visited stops are arranged in ascending order based on their criterion function (further explained) and are added to the list U. Subsequently, a restricted candidate list (referred to as RCL), comprising the first α nodes from the U list, is created. Following this, a node is randomly selected from the RCL and inserted as the initial node for the current route. The chosen node is then removed from the U list and retained as the current node.

The selection and addition of new non-visited stops in the list U to the current route follow the outlined procedure:

Creation of eligible list (Le): An eligible list is formed to sort all non-visited stops in increasing order of the criterion function value. It's important to note that the eligible list (Le) is applicable to nodes adhering to predefined constraints.

Construction of restricted candidate list for eligible nodes (RCL-e): In this step, the list RCL-e is constructed, comprising the α first nodes. If the considered RCL-e is empty, new routes are generated. Otherwise, one stop is selected from the RCL-e, inserted into the current route, saved as the current node, and removed from the list U.

If a new route needs to be generated, the bus (i.e., route) returns to its respective school(s) for dropping off students and then goes back to the closest garage. This implies that it is not mandatory for the bus to return to the garage from which its operation started.

This study incorporates two distinct criterion functions. The first one relies on the mechanism of selecting the next non-visited stop based on the closest distance. The second criterion function is introduced as $cf = (n + m)/d$, where:

n: Represents the number of eligible students (s).

m: Denotes the number of students in the candidate stop who share a common school with students already picked up by the bus.

d: Indicates the distance between the last visited node (garage or stop) and the candidate stop.

Each criterion function is computed for every non-visited node in the list U.

The procedure is finalized and updated upon receiving a new request. The construction phase has two primary objectives: 1) Focusing on the selection of the next non-visited node based on the characteristics of the problem; and 2) Creating two distinct types of initial solutions. It's crucial to note that during the visitation of each node, the health risk requires more thorough investigation.

To align the problem with reality, the health risk is identified and formulated using recent statistical data on the prevalence of the coronavirus in each district. If the visited node is located in an area with a high prevalence rate, it falls into the highly risky or high-danger district. If the incidence rate is slightly lower, the area is considered part of the orange or danger district. An area with a relatively normal and safe situation is categorized within the yellow district. Region classification is determined based on the results of risk analysis. As a result, the north of Tehran is designated as the yellow district, the center corresponds to the orange district, and the south is associated with the red district.

When the bus visits nodes, the penalty demand is applied proportionately based on the location of the stop. Specifically:

In the yellow district, each student is considered as 1 demand.

In the orange district, each student is considered as 1.1 demands.

In the red district, each student is considered as 1.2 demands.

This simple strategy ensures that the bus is required to pick up a smaller number of students in areas with a high prevalence rate, emphasizing social distancing.

Social distancing directly influences the health of students by preventing the spread of illness through the minimization of respiratory droplets. The implementation of social distancing measures on buses, including spacing out seating or limiting the number of students, can significantly decrease the likelihood of disease transmission.

The following constraints are checked during each stage of the construction phase:

During the allocation phase, when assigning a student to a bus stop, simultaneous checks must be conducted for both the maximum walking distance (from student location to bus stop) and the maximum allowable number of students for each stop (risk constraints).

In the routing stage, new, unvisited nodes are selected and incorporated into the route. This continues as long as the risk threshold (a combination of traffic volume risk and density of population risk), bus capacity, and school time window constraints are not exceeded, all without violating these constraints. The values of traffic volume and density of population are mentioned in Section 6.6.1.

The crucial parameter is the size of α: if it is small, the construction heuristic focuses on the greedy mechanism, whereas if it is large (equal to N), the solution adopts a more random behavior.

6-4-2- Improvement phase

After the construction of an initial solution, it proceeds to the next stage for further enhancement. As previously mentioned, two improvement strategies are proposed: 1) selecting neighborhoods using a fixed traditional order; 2) selecting neighborhoods based on their performance. The performance of each neighborhood is contingent on the execution time of the operators and their impact on altering the objective function. In essence, the first strategy substantiates the VND heuristic as a simple yet effective algorithm, while the second strategy facilitates an appropriate trade-off between intensification and diversification mechanisms.

6-4-2-1- Variable neighborhood descent (VND)

The improvement phase is anchored in the Variable Neighborhood Descent (VND), a variant of the Variable Neighborhood Search (VNS) metaheuristic method (Mladenović, 1995; Hansen and Mladenović, 1997, 1999). The VND heuristic requires a hierarchical order of neighborhoods, typically arranged with simpler, less complex, and less perturbative neighborhoods at the beginning of the list. To break free from local optima, larger neighborhoods are chosen when no further improvement can be attained from the smaller heuristics.

Small neighborhoods intensify fewer solutions and can be explored in less time compared to large neighborhoods. Larger neighborhoods are only invoked when all smaller neighborhoods prove ineffective, signifying that the current solution has reached a local optimum with respect to all smaller neighborhoods. Once any local search improves the current solution, the move is executed, and VND recommences the search from the first neighborhood in the list. The operation of VND ceases when the solution attains a local optimum for all considered neighborhoods.

To determine the right order of neighborhoods, various combinations are tested, and the most promising order is integrated into the VND algorithm (refer to Section 6-6-1 for further details).

Before applying any intra or inter-local search operator, it is crucial to assess both the cost of the solution and its compliance with the predefined constraints. If the local search operator discovers superior solutions and the considered constraints are satisfied, the corresponding move is executed; otherwise, it is discarded. The local search concludes when the solution becomes trapped in a local optimum and cannot be further improved by executing any of the local search operators. As mentioned earlier, both m-VND and p-VND metaheuristics utilize VND in the improvement phase.

6-4-3- Modified adaptive neighborhood selection

In the second improvement strategy, following the well-known method of ALNS (adaptive large neighborhood search, proposed by Ropke and Pisinger (2006)), albeit with a slight modification, we select the neighborhoods based on their performance. In this strategy, the invocation of the neighborhoods occurs in the order of their effectiveness with respect to the problem at hand.

In practice, the search process is divided into a number of segments, each consisting of a number of consecutive iterations, denoted as n. The roulette-wheel mechanism is utilized to select the next neighborhood h_i for the operation in each iteration n as follows:

$$\frac{w(h_i)}{\sum_{i\in I} w(h_i)} \quad (6\text{-}37)$$

The expression $w(h_i)$ represents the weight assigned to each employed neighborhood. This probability is contingent on the weight of each operator. Initially, all neighborhoods have equal weight. However, at the end of each segment (following a series of consecutive iterations), the weights of all heuristics are updated based on the following formula:

$$w_{i,j+1} = (1-\gamma)w_{i,j} + \gamma\frac{\mu(h_i)}{\varphi_{ij}} \quad (6\text{-}38)$$

When a segment ends, new weights are calculated based on the accumulated score of each heuristic in that segment.

In the formula above, $w_{i,j}$ represents the weight of neighborhood i in segment j, $\mu(h_i)$ indicates the accumulated score of neighborhood i in the last segment, and φ_{ij} denotes the number of times neighborhood i is repeated in the last segment. The value of $0 < \gamma$<1 is a crucial factor for controlling the adaptive mechanism's behavior in the proposed algorithm. When γ is set close to 0, the previous heuristic weight is considered, but if γ is set to 1, the heuristic weight is determined solely based on new accumulated scores.

This study's modification pertains to the neighborhood scoring mechanism. Specifically, instead of assigning a fixed score to neighborhoods that generate better or worse solutions, each neighborhood's score is dynamically calculated based on its performance and role in the intensification and diversification mechanism (refer to Nasri et al. (2021) for more details). This method dynamically influences operator selection through a combination of three partial functions, which are aggregated to determine the score of heuristic i in each iteration n as follows:

$$\mu(h_i) = 0.5 * \emptyset_n(f_1(h_i) + 0.5 * \emptyset_n f_2(h_i, h_k) + \delta_n f_3(h_i) \quad (6\text{-}39)$$

The first criterion is expressed by the following formula:

$$f_1(h_i) = \frac{I_n(h_i)}{T_n(h_i)} \qquad (6\text{-}40)$$

where $I_n(h_i)$ represents the change in the objective function when the heuristic is applied, and $T_n(h_j)$ is the time it takes the heuristic to explore.

The second criterion explores the dependency among operators in terms of their performance, expressed by the following formula:

$$f_2(h_i) = \frac{I_n(h_i, h_k)}{T_n(h_{i,}h_k)} \qquad (6\text{-}41)$$

where $I_n(h_i, h_k)$ represents the change in the fitness function, and $T_n(h_{i,}h_k)$ represents the time it takes the heuristic i to be recalled after the heuristic k.

The third criterion, presented below, conveys the time elapsed since the last execution of the heuristic. It indicates the duration of inactivity for the heuristic and assesses the possibility of using it.

$$f_3(h_i) = \tau_n(h_i) \qquad (6\text{-}42)$$

The measures f_1 and f_2 are used to control intensification, and the measure f_3 is used to control diversification. $\emptyset_n$ and δ_n parameters in this model are used to weigh f_1, f_2 and f_3. Specifically, $\emptyset_n$ is the intensification parameter that controls and weighs f_1 and f_2, and δ_n is used to weigh f_3. In each iteration, if there is an improvement in the objective function, the value of $\emptyset_n$ is increased and the value of δ_n is reduced. Conversely, the value of $\emptyset_n$ decreases and the value of δ_n increases when the objective function shows a worse performance. This concept is expressed as follows:

$$\emptyset_n((h_j) = \begin{Bmatrix} 0.99 & if\ improve & \\ max\{(\emptyset_{n-1}) - 0.01, 0.01\} & & otherwise \end{Bmatrix} \qquad (6\text{-}43)$$

$$\delta_n = 1 - \emptyset_n(h_j) \qquad (6\text{-}44)$$

Compared to the traditional VND, our adaptive model is novel in several aspects: 1) The assigned weight to each neighborhood aligns with its performance, operation time, and operator's dependency; 2) Our approach is more adaptable to problem size; 3) It achieves a balance between diversification and intensification in the search space; and 4) Even bad moves have a chance of selection, albeit with a lower probability.

6-4-4- Type of neighborhoods

This study incorporates two types of neighborhoods, specifically designed for the special and common school bus routing problem (SBRP). The common neighborhood resembles the well-known VRP, while Special cases align with problem characteristics such as single-load, mixed load, and health concepts. The selection mechanism for neighborhoods differs between the two strategies outlined. In the adaptive layer, neighborhoods are chosen based on their performance, while the Variable Neighborhood Descent (VND) approach requires recalling neighborhoods in order of complexity, from small to large (more complex).

To achieve the right combination of neighborhoods using the VND approach, a pilot study is conducted (refer to Section 6-6-1).

- **Remove-insert intra route operator**

This operator removes a stop from the candidate route and inserts it at another location within the route without checking the feasibility constraint.

- **2-opt intra route operator**

This operator tries to select two edges and removes them from a candidate route, resulting in an incomplete route. Subsequently, two new edges are introduced to reconnect the route, necessitating a reversal in the order of visited stops.

- **Remove-insert operator (considering health feature)**

This operator tries to remove a stop and reinsert it within the same district, whether on the same route or not. It executes this move independently in each district (red, orange, and yellow), thereby (1) mitigating the potential spread of coronavirus between districts and (2) enhancing intensification in the solution space.

- **Similar school-remove-insert**

This operator segregates bus stops and incorporates them into routes where the associated schools for those stops' students are already assigned. This action prevents additional school insertions on the route, contributing to local intensification in the search space.

- **Convert single to mixed load**

This operator aims to decrease the number of single-load routes by relocating stops from single-load routes to mixed-load routes. This aligns with the concept of the mixed load effect.

- **Convert mixed load to single load**

This operator removes a stop from a mixed-load route and inserts it into another route to eliminate stops where students travel a long distance relative to existing school(s).

- **Replace**

To minimize total travel time, this operator selects a non-visited stop from the list and replaces it with a stop already included in a route. The list of non-visited stops is sorted in descending order based on the number of students who can reach that stop. Before applying any move, both a cost check and a student allocation subproblem (refer to Chapters 2 and 3) must be executed.

- **Remove**

This operator is executed by removing a stop from the current route to decrease the cost while maintaining solution feasibility. Similar to the previous move, before applying any move, a student allocation subproblem must be solved. If it contributes to achieving a feasible solution, the move is performed. Due to the triangular inequality characteristic, performing a cost check is not essential since stop removal always results in a reduction in travel cost.

6-4-5- Diversification stage

Diversification strategies aim to generate new initial solutions, helping escape local optima and explore diverse areas of the search space. This contributes to establishing a promising starting point for the subsequent local search block. The diversification mechanism guides the algorithm to search unexplored regions of the solution. To discover the impact of diversification, it's essential to examine the degree of perturbation. The size of perturbation is crucial, as it can lead to significant improvement or exacerbate the current solution.

Previous evidence indicates that lower deterioration increases the risk of getting trapped in local optima. Conversely, large degrees of deterioration can lead the algorithm away from the path toward the global optimum. Therefore, diversification heuristics must be designed to align with the size of perturbation and the status of exploration. In this context, two diversification structures are proposed: 1) Perturbing part of the solution space (randomly or constructively); 2) Generating new initial solutions compatible with the multi-start procedure.

The multi-start metaheuristic aims to escape from local optima by repeating a number of iterations in both the construction and improvement phases. In contrast, the perturbation configuration involves executing the construction phase only once and then repeating the local search and perturbed heuristic for consecutive iterations.

In the perturbation configuration, destroy and repair heuristics are employed. Initially, the best-found solution is partially ruined by the destroy operator. Subsequently, the repair heuristic aids in reconstructing the new current solution, preparing it for the local search block. The destruction phase introduces two types of heuristics: 1) a random-destroy heuristic, which randomly destroys a part of the solution space; 2) a constructed-destroy operator, which aims to destroy the part of the solution with the highest removal gain. When a random part of the solution is chosen, there is no prior knowledge about which part needs to be destroyed. In contrast, when the constructed-destroy heuristic is used, the part of the solution space with the lowest gain (based on previous performance) is destroyed. The following explanation focuses solely on the perturbation configuration, encompassing the destroy and repair operators.

6-4-5-1- **Destroy phase**

□ Random destroy heuristic

In this method, a random part of the solution space (i.e., random bus stops) is selected, removed, and added to the list U. The perturbation size is controlled by the value of $p * q$, where p denotes the number of routes constructed in the improvement phase, and q is the perturbation size, controlled between q_{min} and q_{max}. If the perturbation mechanism guides the search towards the global best solution in the next improvement phase, the value of q is set to q_{min} to preserve the intensification strategy. However, if no global best solution emerges, the value of q is increased by 10% to maintain diversification. This process continues until the value of q reaches q_{max}. In this case, the value of q is set to q_{min} again. The reason behind this procedure lies in the fact that when the value of q is close to q_{max} , the large value of q proves inadequate for enhancing the current solution. Consequently, a smaller value of q can contribute significantly to the improvement of the current solution. Switching from q_{max} to q_{min} involves addressing two objectives: controlling computing time and considering the intensification in the search space.

□ Constructed destroy heuristic

In this stage, the heuristic attempts initially to select the stops based on the ratio of $\frac{d_i}{s_i}$, where the value of d_i represents the amount of demand in each stop, while the value of s_i is represented by the following formula: $dist(prev(i), i) + dist\big(i, next(i)\big) - dist(prev(i), next(i))$.

This ratio is sorted for all bus stops in ascending order. Following this, k stops from the top of the list are selected and added to list U to choose the bus stop with the lowest demand and the highest removal gain. The priority is to remove stops and add them in a location that reduces the cost. The value of k is controlled between $k_{min} (k_{min} = n.\varepsilon_{min})$ and k_{max} $(k_{amx} = n.\varepsilon_{max})$, where n is the number of stops, and ε_{min} and ε_{max} are the minimum and maximum percentages of stops to be removed in the diversification stage. Each time diversification is applied, one of the aforementioned destroy operators is randomly chosen. The mechanism of updating ε is similar to q in the random-destroy heuristic.

□ Repair phase

The repair operation is straightforward and efficient. All removed stops in list U are added to the solution through the GRASP procedure, ensuring the completion of the current solution without violating the considered constraints. During this stage, all students associated with a given stop in list U remain fixed, eliminating the need for student allocation. The newly generated solution becomes the input for the local search heuristic, considered for further improvement.

6-5- Instance generation

The process of generating certain features is akin to what is outlined in Chapter 4, which includes generating stops, schools, students, and garages. However, our focus here is specifically on issues related to risk consideration. In this context, our generated instance is divided into three districts: north (represented by area a), center (represented by area b), and south (represented by area c). To integrate the risk concept into the current problem, we consider characteristics such as population density and traffic volume.

To be more specific, the risks associated with high-density population and high traffic volume are confined to the center and south districts (B and C), underscoring the significance of integrating risk analysis into the studied problem. In these areas, the risk factors for each traffic volume and population density are randomly generated within the intervals of (1.1 to 1.40), respectively.

For other areas with low and medium levels of risks, both traffic volume and population density risks are set to 0 to mitigate the complexity of the problem. To visualize the prevalence distribution of COVID-19, three health coefficients are considered for the north (1), center (1.1), and south (1.2) districts. This coefficient is multiplied by the number of students to be picked up or dropped off. Accordingly, when the bus picks up a student from the north district, they are equivalent to 1, but when a student is picked up in the south district, they are equivalent to 1.2. Therefore, for a bus with a capacity of 25 passengers, the maximum number of students that can be picked up in the southern district adheres to the social distancing rule on the bus (25/1.2). In other words, for the district with a high prevalence of COVID-19, the bus picks up a limited number of students lower than its default capacity.

6-6- Computational results

Our computational experiments consist of three stages outlined below:
In the first stage, the heuristic parameters are tuned (refer to Table 6-3) through a full factorial experiment conducted on a subset of instances.
In the second stage, the performance of the four proposed algorithms is compared across all instances, including small, medium, and large.
In the third stage, some risk analyses are undertaken to illustrate the impact of risk constraints on the objective function.

6-6-1- Calibration of metaheuristic

The previously introduced metaheuristics have parameters that require adjustment and calibration to strike a reasonable balance between solution quality and computing time. The configuration of controllable parameters in metaheuristic algorithms is crucial in their design, aiding in a better understanding of their behavior. To tune the controllable parameters of the multi-start and perturbation configurations, an evaluation is conducted under various values, and the optimal values are selected.

The primary parameters are detailed in Table 6-3. A full factorial experiment, combining all parameter settings shown in Table 6-3, is conducted to solve 10 instances (4 from set S, 4 from set M, and 2 from set L) using the four proposed metaheuristics. For each instance and specific parameter, 5 runs are executed. In some analysis runs, all neighborhoods in the improvement phase are deactivated for all metaheuristics, rendering the algorithm behaves like a random restart.

Two performance metrics are examined: the average solution cost and the average computational time. It's essential to highlight that the maximum number of iterations is not included in the list of analyzed parameters. A larger number of iterations logically leads to better solutions but requires more computational time. Hence, in this calibration stage, the number of iterations is kept constant at 400 for all calculations, and only the maximum number of non-improvement iterations is considered in our analysis.

Table 6-4 displays the P-values obtained from the F-tests. Asterisk values signify the significance of each parameter on both the objective function and the solution time of the algorithm. As per Table 6-3, crucial parameters of the algorithm that significantly impact both solution quality and computational time include the majority of local search operators, the minimum and maximum percentage of routes and stops to be destroyed, and the maximum number of non-improvement iterations.

Meanwhile, parameters like school removal and mixed-to-single-load operators only exhibit a significant effect on computing time. The optimal values for these parameters are provided in Table 6-5. Analysis of the considered neighborhoods indicates that, on average, the impacts of similar school removal (for all metaheuristic configurations) and mixed load-to-single-load operators (for m-VND and p-HA metaheuristics) on the quality of solutions are less pronounced than other operators. However, they do yield a slight improvement in results when activated. As a result, these two operators are excluded from further analysis.

It is worth noting that we independently explore the effect of different numbers of iterations on the solution's quality. Other parameters are held constant at their optimal values as listed in Table 6-5. Similar to the previous analysis, 10 instances are utilized, and the solution is assessed for varying numbers (200, 250, 300, 350, 400, 450, 500, 550, and 600) of iterations. The analyses reveal that the optimal number of iterations, striking an ideal balance between computing time and solution quality, is 450.

Table (6-3) Parameters and Levels Tested

Parameter	Description	Value	No. of levels
N1=remove-insert within route	Remove and insert the stop on the same route	On, off	2
N2= remove-insert (considering healthy feature)	Remove and insert the stop in the area with the same level of risk	On, off	2
N3=Similar school removal	Remove and insert the stop on the route that has the same school	On, off	2
N4=convert single to mixed load	Remove the stop from the single-load route and insert it on a mixed-load route	On, off	2
N5=convert mixed load to single load	Remove the stop from the mixed-load route and insert it on another route	On, off	2
N6=replace	Replace a non-visited stop on the current route	On, off	2
N7=remove	Remove the stop on the current solution and reallocate its students to all stops	On, off	2
N8=2-opt		On, off	2
α	Size of restricted candidate list	1,2,3,4	4
q_{min}	Minimum percentage of routes to be removed at perturbation phase	1%,5%,10%	3
q_{max}	Maximum percentage of routes to be removed at perturbation phase	20%,30%,40%	3
γ	Reaction factor in adaptive mechanism	0.5, 0.6,0.7	3
nit	Maximum number of iterations without improvement	10,15,20	3
ε_{min}	Minimum percentage of stops to be removed	1%,5%,10%	3
ε_{max}	Maximum percentage of stops to be removed	25%,30%,35%	3

Table (6-4) P-values of the F-tests

Parameter	Average solution cost	CPU time
N1=remove-insert within route	<0.05	<0.05
N2= remove-insert (considering healthy feature)	<0.05	<0.05
N3=Similar school removal	>0.05	<0.05
N4=convert single to mixed load	<0.05	<0.05
N5=convert mixed load to single load	>0.05	<0.05
N6=replace	<0.05	<0.05
N7=remove	<0.05	<0.05
N8=2-opt	<0.05	<0.05
α	<0.05	>0.05
q_{min}	<0.05	<0.05
q_{max}	<0.05	<0.05
ε_{min}	<0.05	<0.05
ε_{max}	<0.05	<0.05
γ	<0.05	>0.05
nit	<0.05	<0.05

Table (6-5) Optimal Setting of Considered Metaheuristics

Parameters	m-HA	m-VND	p-HA	p–VNDA
N1=remove-insert within route	On	On	On	On
N2= remove-insert (considering healthy feature)	On	On	On	On
N3=similar school removal	Off	Off	Off	Off
N4=convert single to mixed load	On	On	On	On
N5=convert mixed load to single load	On	Off	Off	On
N6=replace	On	On	On	On
N7=remove	On	On	On	On
N8=2-opt	On	On	On	On
α	3	3	2	3
q_{min}			%5	5%
q_{max}			30%	40%
γ	0.5			0.7
nit	10	15	15	20
ε_{min}			5%	10%
ε_{max}			30%	25%

In general, determining the optimal order of neighborhoods used in a Variable Neighborhood Descent (VND) heuristic can significantly impact the solution's quality. To comprehend which combination works best, we test different orders of neighborhoods on a subset of instances. On average, the most effective order of neighborhoods incorporated in the VND heuristic is as follows: Remove-insert intra-route operator, 2-opt intra-route operator, remove-insert (considering the health situation), convert single-load to mixed-load, remove, and replace operators.

6-6-2- Metaheuristic comparison

Having established the optimal parameter settings for each solution approach, we now compare the proposed metaheuristics by testing them on small, medium, and large instances. To ensure a fair comparison in terms of solution quality and computing time, we run each metaheuristic five times with a fixed number of 450 iterations for each instance.

The experimental analysis is carried out on 100 instances divided into three subsets: Set S, Set M, and Set L. Set S encompasses 30 instances (10-30 stops), Set M comprises 30 instances (40-60 stops), and Set L includes 40 instances (70-100 stops). Furthermore, we take into account maximum walking distances of 5, 10, 15, and 20 in our calculations. (For a more in-depth look at the results of each experiment, refer to Appendix 9).

The consolidated results are displayed in Tables 6-6 and 6-7. Table 6-6 delineates the percentage gap between the best-found solution of each metaheuristic after 5 runs and the optimal solution (GAMS/CPLEX solver). Meanwhile, Table 6-7 illustrates the percentage difference between the four metaheuristic configurations. Specifically, after all runs are executed, the metaheuristic with a better quality of solution (on average) serves as the basis, and the performance of other metaheuristics is compared against it. In both Tables 6-6 and 6-7, the first column indicates the type of metaheuristic, while the subsequent three columns present the set of instances (from small to large).

Table (6-6) Best Gap from Optimal Solution

Metaheuristic	SET S	SET M	SET L
p-HA	2.28%	----	----
m-VND	3.23%	----	----
m-HA	4.37%	----	----
p-VND	5.05%	----	----

Table (6-7) Percentage Gap between p-HA and the other Three Proposed Metaheuristics in Sets S, M, and L

Metaheuristic	SET S	SET M	Set L
m-VND	2.20%	2.95%	3.51%
m-HA	2.61%	3.36%	4.05%
p-VND	3.81%	4.32%	4.70%

Table (6-8) Total Computing Time of each Metaheuristic in Small, Medium, and Large Instances (in Seconds)

Metaheuristic	SET S	SET M	Set L
p-HA	86,476	1,662,819	64,237,940
m-VND	147,065	2,887,037	98,656,048
m-HA	209,155	3,856,332	126,537,481
p-NVD	80,165	1,544,327	61,366,479

Analyzing the results in Table 6-6 reveals that p-HA features the lowest percentage gap with the optimum solutions (exact method) in small instances. Additionally, m-VND ranks second in terms of this gap.

Comparing the four proposed metaheuristics (Table 6-7), we observe that, on average, p-HA outperforms the other configurations. It appears that the combination of perturbation and adaptive mechanisms can lead to better performance. Accordingly, p-HA utilizes two mechanisms simultaneously: the perturbation heuristic and the selection of a local search operator based on its performance. Hence, p-HA is considered the baseline algorithm against which comparisons are made with the other proposed metaheuristics.

For a better understanding of the performance of the four metaheuristics, we compared them across small, medium, and large instances. It's worth noting that m-

VND, followed by m-HA, exhibits a lower percentage gap than the baseline metaheuristic. On the other hand, p-VND shows a higher deviation from the baseline, indicating that this combination is not consistent with the considered instances and is therefore of little interest. Additionally, in terms of computing time, Table 6-8 indicates that m-HA is the slowest metaheuristic, and relying on this combination requires more computational time.

6-6-3- Risk factor effect

This supplementary analysis aims to investigate the effect of each risk constraint on the objective function. Considering its superior performance compared to the other three proposed metaheuristics, the p-HA configuration is selected for this purpose. The experiment is conducted on a subset of 12 instances.

Specifically, we aim to understand the extent to which the value of the objective function would increase if all the risk constraints are active in the model, and how much it can decrease if the risk constraints are relaxed.

The results indicate that omitting the safety risk constraints (constraints 1) leads to a decrease in the value of the objective function by up to 6.53%. If all risk constraints (safety, healthy, and traffic) are relaxed, the objective function decreases by 15.07%. This substantial difference should be taken into consideration by the municipality or any organization in the private sector. Additionally, the results show that the size of the problem has a greater effect on the risk constraints. Furthermore, risk constraint relaxation significantly impacts the objective function in all instances, particularly in large ones.

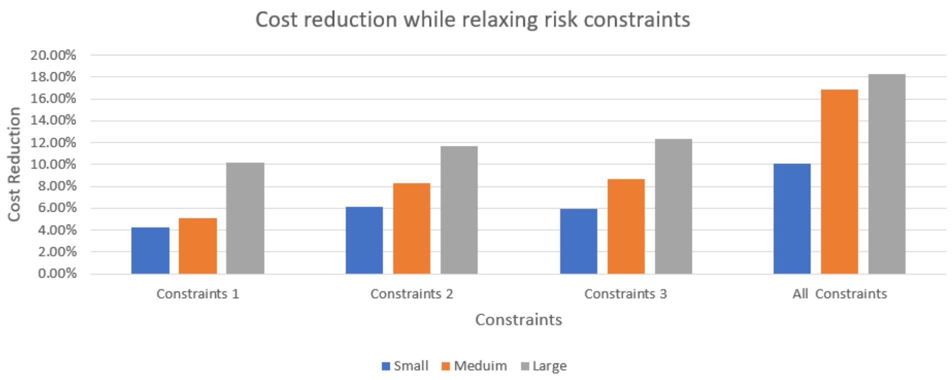


Figure (6-2) Objective Function Reduction due to Relaxing Risk Constraints

6-7- Conclusions and future research

This chapter addresses the school bus routing problem by integrating both bus stop selection and route generation while considering a mixed-load context and various risk constraints. Specifically, students are assigned to potential bus stops based on the walking distance from their homes. Simultaneously, these stops are incorporated into the bus route to transport the students to school with the aim of minimizing travel time. The study incorporates safety, traffic, and health risk constraints, including a maximum number of students to be assigned to a bus stop, as well as traffic and health risk constraints imposed on the arcs and nodes.

To efficiently solve the generated small, medium, and large instances of the problem (totaling 100 instances) within a reasonable computing time, we have devised and proposed four different metaheuristics capable of generating various diversification strategies in the solution space. Overall, on average, the p-HA metaheuristic outperforms other metaheuristics in terms of solution quality. Regarding computing time, the p-VND metaheuristic, followed by p-HA (though with a slight difference), demonstrates faster performance than others. Consequently, p-HA appears to be a more reliable metaheuristic. On the other hand, the m-HA configuration cannot provide the solution in a reasonable time and, therefore, requires more computing time.

These metaheuristic configurations operate in two directions, placing as much emphasis on the mechanism of neighborhood selection as on the status of diversification. It can be inferred that in designing the metaheuristic, consideration needs to be given to the mechanism of diversification, the type of neighborhoods, and the execution time of neighborhoods.

We have conducted risk analyses and identified three major risks. Clearly, these risk constraints will become more prominent as the student transportation situation returns to normal. With the reopening of schools, the safety of school buses will be a major concern for many transport companies and students' parents. In this regard, taking safety instructions seriously can help efficiently mitigate and control such concerns, thereby reducing the probability of accidents. Concerning safety risk, many accidents occur for students when they are waiting for the bus or getting on or off the vehicle.

Future research can aim to include stochastic parameters in the model. In all chapters presented so far, all input parameters of the case study (e.g., arrival time of the bus, speed of the bus) are considered to be deterministic. However, in real-world applications, there are different factors that make these parameters stochastic. For example, due to an unpredictable situation, a bus may have a delay on the route.

Moreover, the expected speed of the bus is not usually a fixed parameter due to traffic conditions. As a future work, these parameters can be considered within uncertain conditions. Another interesting idea is to consider an objective function to minimize the total student journey, including travel time from home to stop, waiting time at the stop, travel time on the bus, as well as pick-up time at the stop and drop-off time at the school. Finally, it is more rewarding to investigate both hard and soft time window constraints simultaneously.

Chapter 7:

Conclusion and future research

7-1- Conclusion

A bus network plan greatly influences an urban transport system in real-life situations, impacting safety, reliability, and desirability. Transporting students to and from school poses a challenge for local governments aiming to optimize budgets. Inefficient transportation plans may result in issues like heightened noise, pollution, accident rates, and dissatisfaction among students and citizens. The surge in fuel prices and extended time spent in traffic has prompted families to explore the public bus transportation system for their children.

Efficient coordination and planning in the urban transportation network are crucial for any action or policy in this area. Authorities aim to provide an effective transport system, generating significant budget savings by considering limited resources. However, due to resource constraints, assigning a bus to each school is not feasible (see Park and Kim, 2010).

In student transportation, a key challenge involves effectively managing both morning and afternoon shifts in an integrated manner during student pick-up and drop-off. Municipalities are actively seeking solutions to efficiently share resources between these two shifts.

Sharing resources between schools enhances the efficiency of the school bus system but introduces complexity, leading to overcrowded buses and lengthy routes. Designing bus routes based on a mixed-load/multi-load plan appears to be an effective solution. This approach considers appropriate objectives, assumptions, and constraints.

In another scenario, a student from a specific bus stop may have different school time windows, involving both primary and elementary school students at each stop. When a bus serves students from different schools simultaneously, there's a higher chance of missing eligible students at the next stops, leading to increased computing time and costs. Unlike the classical routing problems where each customer is visited only once, allowing split loading means students from the same stop can be served through multiple visits. This potentially results in significant savings in travel costs and the number of buses.

The split pick-up and drop-off method allows multiple visits to serve each stop, proving particularly beneficial when dealing with students from different schools at a candidate stop or facing tight capacity constraints. In Tehran, where the majority of students rely on public transport, it is crucial to raise awareness about the risks associated with each mode of transportation. This awareness should extend to the school bus routing problem, considering factors such as the time taken for bus stops, evacuating a bus due to an accident, and the duration students spend on the bus.

These policies establish rules that facilitate the safe and convenient transportation of students to their respective schools. In real-life applications, Student Bus Routing (SBR) must consider additional constraints and factors related to the usability of the transportation network. Large municipalities consistently strive to develop operational strategies for efficiently managing school bus transport systems. Real-world student transportation typically involves incorporating various features such as student riding time, stop and school time windows, mixed-load and single-load planning, and forward and backward trips. Undoubtedly, these features align with the real-life nature of the problem.

This research makes key contributions, summarized as follows:

- Proposing a novel mathematical formulation for the Student Bus Routing Problem (SBRP) that incorporates defined objectives and constraints.
- Introducing a distinctive SBRP scheme with morning and afternoon features, addressing mixed-load (transporting students from different schools on the same bus), multi-shift load (carrying students from morning and afternoon shifts simultaneously), and split-load all at once.
- Comparing the impact of mixed-load and single-load approaches on minimizing the number of buses, total traveled distance, weighted average riding distance of students, and occupied bus capacity.
- Offering a risk assessment method to identify more significant threats that adversely affect the health and safety of students.
- Examining the total transportation cost when relaxing risk constraints in the model.
- Addressing three decision-making subproblems: location, allocation, and routing.

The SBRP is an NP-hard problem, making polynomial-time solutions impractical. To tackle large instances effectively, employing metaheuristic approaches is more efficient. Numerous heuristic and metaheuristic methods exist in the literature for solving SBRP. The challenge lies in designing a suitable heuristic tailored to the specific problem type, striking a balance between computing time and solution quality, and managing the trade-off between intensification and diversification.

An effective metaheuristic should possess enhanced capabilities, including the ability to generate new solutions likely to improve current or previous ones. It should also facilitate exploration of the most promising search areas to reach the global optimum and be capable of escaping local optima. These conditions emphasize the significance of intensification and diversification strategies in metaheuristic approaches.

Two crucial considerations in designing a metaheuristic are diversification and intensification. Diversification involves the ability to explore various and different parts of a search space, while intensification focuses on attaining high-quality solutions within those areas. Both diversification and intensification are essential strategies that guide the search process efficiently. In clearer terms, the current research aims to devise a heuristic capable of exploring both new regions of the search space and the existing desirable areas.

The combination of diversification and intensification helps the algorithm find the optimal solution. However, the challenge lies in determining the appropriate extent to apply each strategy to the solution space. Excessive exploration increases the likelihood of finding the global optimum but reduces efficiency, while heavy exploitation can lead the algorithm to get stuck in local optima. Continuous intensification raises the risk of being trapped in local optima. Therefore, a careful analysis is necessary to calculate the right degree of exploration. Despite its importance, there is currently no robust practical guideline for achieving this balance.

Each algorithm employs a unique balance between exploitation and exploration. Therefore, it is crucial to design an algorithm with a flexible architecture that can strike a compromise between diversification and intensification. Intensification and diversification can be implemented as operators, actions, or acceptance criteria strategies in metaheuristics. Addressing these considerations involves defining specific neighborhoods, perturbation mechanisms, probability of accepting objective functions, size of perturbation, and other relevant factors.

The contributions of this thesis in incorporating metaheuristic approaches are as follows:

- Incorporating an oscillation strategy to explore the infeasible part of the solution space.
- Investigating the heuristic's ability to navigate the infeasible part of the solution space.
- Examining the speed of transition between the feasible and infeasible parts of the solution space.
- Proposing an ALNS (Adaptive Large Neighborhood Search) metaheuristic for solving medium and large instances of SBRP, tuning its parameters through a statistical experiment, and subsequently comparing it with existing benchmarks.
- Analyzing separately the performance of each insertion and removal operator;
- Developing neighborhoods specific to SBRP;
- Configuring the Iterated Local Search (ILS) metaheuristic with an adaptive mechanism.
- Introducing heuristics that investigate and define the type of diversification and intensification within the solution space.

In Chapter 2, we initially introduced two metaheuristics, N-ILS and I-ILS, designed to address the school bus routing problem. The core concept involves introducing an oscillation strategy with three key features: exploring both the infeasible and feasible parts of the solution space, regulating both the exploration capability and the transition speed between feasible and infeasible parts of the solution space, and implementing the restore operator when the solution exhibits a high rate of violation.

This chapter focuses on the School Bus Routing Problem (SBRP), integrating bus stop selection and route generation into a single optimization approach. Students are assigned to potential bus stops based on their walking distance from home. Simultaneously, these stops are incorporated into the bus route to enable students to reach school while minimizing travel distance.

In comparison with Schittekat et al. (2013), computational experiment results indicate that N-ILS outperforms I-ILS. In terms of computation time, N-ILS can yield near-optimal solutions within a limited timeframe. Consequently, the N-ILS metaheuristic emerges as the superior approach concerning solution quality, robustness, and computation time compared to I-ILS.

In Chapter 3, various types of Large Neighborhood Search (LNS) and Adaptive Large Neighborhood Search (ALNS) metaheuristics are introduced, leveraging the oscillation strategy. Similar to the second chapter, the problem presented by Schittekat et al. (2013) is examined as an interesting variant of the Vehicle Routing Problem (VRP). The metaheuristic is evaluated on a set of 104 School Bus Routing Problem (SBRP) benchmark instances proposed by Schittekat et al. (2013). In the initial stage, the formulation presented is precisely solved using the CPLEX solver in GAMS.

Since only 43 instances could be solved using the mentioned method, simplified Large Neighborhood Search (LNS) and Adaptive Large Neighborhood Search (ALNS) metaheuristics are designed to tackle small, medium, and large problem instances within a reasonable timeframe. Statistical analysis is conducted for each metaheuristic to determine the optimal parameter settings. After establishing the best parameter settings for each solution, a comprehensive comparison is made between the two metaheuristics, considering solution quality, robustness, and computing time across all instances. Additionally, two scenarios are devised to delve more precisely into the proposed algorithm, aiding in understanding the behavior of algorithms, whether the oscillation strategy is employed or not.

Evaluating the percentage gap between the best solution found and the best-known solutions (BKS_{MH}), we have observed that both LNS-1 and LNS-2 produced approximately similar results for both scenarios. However, other heuristics exhibited poor performance in the second scenario. Conversely, heuristics in the first scenario delivered results in a shorter computing time. In summary, the first scenario appears to be more reliable and offers better performance in terms of solution quality and computing time.

In the first scenario, computational experiment results reveal that the Adaptive Large Neighborhood Search (ALNS) is highly competitive compared to the best metaheuristic introduced by Schittekat et al. (2013). Moreover, the ALNS algorithm systematically explores extensive portions of the solution space, demonstrating its robustness by adapting to various cases and avoiding frequent entrapment in local optima. A pivotal conclusion from this research is that ALNS stands out as the most effective metaheuristic among all proposed solution algorithms. Additionally, in terms of computing time, both ALNS and Rand-removal with Regret-2 outperform other algorithms.

In Chapter 4, our objective is to introduce a new mathematical formulation and solution approach, considering a mixed-load mode. The results validate the effectiveness of the proposed framework, as it yields higher cost savings compared to the single load mode. The characteristics considered in this study for the School Bus Routing Problem (SBRP) include homogeneous buses, a maximum allowable number of students for each stop, school arrival time, and multiple garages.

In the initial phase, the proposed model is solved for small instances using the CPLEX solver in GAMS. Since this solver can handle up to 20 instances within a reasonable computing time, an Adaptive Large Neighborhood Search (ALNS) with a different configuration is introduced to solve all generated instances. Four experiments are conducted to assess the algorithm's efficiency. In the first scenario, it is evident that implementing mixed-load yields superior solutions. However, concerning student maximum riding time and total route length, the results indicate that, on average, the single-load strategy incurs lower costs. Hence, it becomes imperative to consider accurate constraints for maximum riding time and total route length.

In the second stage, various metaheuristic configurations are applied, and among these, the Shaw removal with Regret-2 demonstrates the best performance. In the third stage, the solutions obtained by the proposed metaheuristic are compared with those obtained by the CPLEX solver in GAMS. The results indicate the average percentage gap with CPLEX is lower than 1.5%, and in seven instances, it attains an optimal solution. Finally, the results of analyzing the proposed metaheuristics are compared with those of the best-known solutions, revealing that ALNS stands out as the most effective metaheuristic configuration. The results of other Large Neighborhood Search (LNS) configurations are not particularly promising.

In Chapter 5, we have explored additional aspects of the School Bus Routing Problem (SBRP), considering both forward and backward trips corresponding to morning and afternoon shifts. This problem not only examines costs but also considers average riding time and the fluctuation of time windows in the two shifts.

Instances for the problem are generated and then categorized into three subsets: small, medium, and large. As anticipated, as the problem size grows, the exact method cannot find an optimal solution within a reasonable timeframe, particularly after instance 16. To address this, we've developed a metaheuristic with an adaptive mechanism. This mechanism selects and implements a local search operator from a set of predefined operators based on its performance, thereby enhancing the likelihood of choosing an efficient operator. The aim is to solve small, medium, and large problem instances within a reasonable time frame using this adaptive metaheuristic.

Two scenarios are devised to comprehend the configuration of the proposed metaheuristic. The objective is to compare neighborhood selection conventionally (based on size) versus systematically.

In the second aspect, we delved deeper into understanding the influence of metaheuristic elements in achieving improved solutions. It's observed that employing the problem-specific knowledge heuristic followed by the cross-exchange heuristic yields superior results. Specifically, we explored the impact of imposing maximum riding time constraints on reducing both total travel time and the number of required buses. Although imposing a maximum riding time can decrease the number of buses, it comes at the cost of longer total travel time. Therefore, policymakers should adopt a trade-off mechanism to efficiently lower both the number of buses and total travel time. The hybrid policy of mixed- and multi-shift loading can result in approximately a 6.82% reduction in the number of buses needed for small instances.

Chapter 6 builds on the discussions of mixed-load mode, bus stop selection, and the school bus routing problem from earlier chapters. It introduces risk characteristics, including safety, traffic, and the maximum number of students allowed at each stop, to enhance the model's realism. To solve the generated small, medium, and large instances of the problem within a reasonable computing time, four different metaheuristics are proposed, each offering distinct diversification strategies.

On average, the p-HA metaheuristic demonstrates superior performance in solution quality compared to other metaheuristics. In terms of computing time, the p-VND metaheuristic, closely followed by p-HA, exhibits faster performance than the others. Consequently, p-HA appears to be a more reliable metaheuristic. However, the m-HA configuration struggles to attain an effective solution within a reasonable time, necessitating more computing time.

7-2- Future works

This dissertation explores various aspects of the School Bus Routing Problem (SBRP) to enhance its realism, as discussed in earlier chapters. Additionally, diverse approaches to designing metaheuristics for this problem are proposed. Despite the contributions of this study, further research in this area is essential. Opportunities for future work have been extensively outlined at the conclusion of each chapter; however, they are summarized as follows:

- **Constraints and objective functions**

Further research can be expanded in several ways, such as considering:

A) Expanding research by incorporating additional objective functions, constraints, and features to model real-life situations. Specifically, minimizing students' walking distance from their houses to stops could be introduced as a second objective function or embedded in the existing one. Another interesting idea is to include an objective function aimed at minimizing the total student journey, encompassing travel time from home to stop, waiting time at the stop, travel time on the bus, as well as pick-up and drop-off times. Additionally, exploring the transportation of some students through an outsourcing mechanism while adhering to budget constraints is essential. This research proposes a beneficial trade-off between the private sector's involvement and the municipality's commitment to cost optimization.

B) Including stochastic parameters into the SBRP model can be beneficial. In the preceding chapters, all input parameters in the case study (e.g., bus arrival time, bus speed) are treated as deterministic. However, in real-world scenarios, various factors can make these parameters stochastic. For instance, unforeseen situations may cause delays in the bus route. Additionally, the expected speed of the bus is typically not fixed due to varying traffic conditions. Future research can explore incorporating these parameters under uncertain conditions.

- **Metaheuristic Approach**

For future research, one of my main priorities is to design metaheuristics that can adapt to the specific problem under investigation. Substantial progress in this field, both practical and theoretical, adds value for any researcher aiming to develop commercial software.

The second focus area involves enhancing specific parts of the metaheuristic to boost its performance and reduce computing time. Moreover, finding more effective methods to verify student allocation feasibility before applying each improvement operator is particularly interesting.

Concerning the oscillation strategy, efforts can be directed towards enhancing its performance by creating a memory list that retains infeasible solutions leading to the global best solution. As capacity violations rise, focus can shift to devising fast heuristics capable of returning to feasible sections of the search space. Ultimately, the researcher is keen on exploring new ideas to strike a profitable balance between diversification and intensification.

References

Ansari, M., Farrokhvar, L., Kamali, B. (2021). Integrated student to school assignment and school bus routing problem for special needs students. Transportation Research Part E.

Archetti, C., Speranza MG. (2012). Vehicle routing problems with split deliveries. International Transaction in operation research, 19(1-2):3-22

Angel, R., Caudle W., Noonan R. and Whinston A. (1972). Computer-assisted school bus scheduling. Management Science, Vol. 18, pp. 279-288.

Azi, N., Gendreau, M., and Potvin, J. Y. (2010). An adaptive large neighborhood search for a vehicle routing problem with multiple trips (Vol. 8). CIRRELT.

Abbasi-Pooya, A., and Kashan, A. H. (2017). New mathematical models and a hybrid Grouping Evolution Strategy algorithm for optimal helicopter routing and crew pickup and delivery. Computers & Industrial Engineering, 112, 35-56.

Arnold, F., and Sörensen, K. (2017). What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. Computers & Operations Research.

Amanda Fernandes., Mònica Ubalde-López., Tiffany C. Yang., Rosemary R. C. McEachan., Rukhsana Rashid., Léa Maitre ., Mark J. Nieuwenhuijsen and Martine Vrijheid.(2023). School-Based Interventions to Support Healthy Indoor and Outdoor Environments for Children: A Systematic Review. Environmental Research and Public Health.

Ana Camila Pérez., Eduardo Sánchez-Ansola., Alejandro Rosete., Omar Rojas., Guillermo Sosa-Gómez. (2022).
A partial evaluation approach for the School Bus Routing Problem. Heliyon 8 (2022).

Bodin, L. D., and Berman, L. (1979). Routing and scheduling of school buses by computer. Transportation Science, 13, 113-129

Bertsimas, D., Delarue, A., and Martin, S. (2019), From school buses to start times: Driving policy with optimization

Bertsimas, D., Delarue A., Eger W, Hanlon J, Martin S. (2020), Bus Routing Optimization Helps Boston Public Schools Design Better Policies, INFORMS Journal on Applied Analytics.

Bennett, B.T., Gazis, D.C., (1972). School bus routing by computer. Transportation Research 6, 317–325.

Braca, J., Bramel J., Posner, B, D S-L. (1997). A computerized approach to the New York City school bus routing problem. IIE Trans,29(8):693–702.

Brandao, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research, 173(2), 540-555.

Bögl, M., Doerner, K. F., and Parragh, S. N. (2015). The school bus routing and scheduling problem with transfers. Networks, 65(2), 180-203.

Boudia, M., Prins, C., Reghioui, M. (2007), An Effective Memetic Algorithm with Population Management for the Split Delivery Vehicle Routing Problem. Hybrid Metaheuristics. Lecture Notes in Computer Science book series (LNCS, volume 4771).

Bock, A., Grant, E., Kinsman, J., and Sanità, L. (2011). The school bus problem on trees. International Symposium on Algorithms and Computation, Springer, pp. 10-19.

Bock, A., Grant, E., Könemann, J., and Sanità, L. (2013). The school bus problem on trees. Algorithmica, 67(1), 49-64.

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

Bearman, N., Singleton, A.D. (2014). Modelling the potential impact on CO2 emissions of an increased uptake of active travel for the home to school commute using individual level data. J. Transp. Heal. 1, 295–304. https://doi.org/10.1016/j.jth.2014.09.009.

Campbell, J. F., North, J. W., and Ellegood, W. A. (2015). Modeling mixed load school bus routing. In Quantitative approaches in logistics and supply chain management (pp. 3–30). Springer.

Campbell, A. M., and Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. Transportation science, Vol. 38, pp. 369-378.

Caceres, H., Batta, R., He, Q. (2017). School bus routing with stochastic demand and duration constraints. Transport Science.

Caceres, H., Batta, R., He, Q. (2018). Special need students school bus routing: consider- action for mixed load and heterogeneous fleet. Socioeconomic Plan Sci.

Calvete, H., Galé, C., Iranzo, J., Toth, P (2020). A Partial Allocation Local Search Matheuristic for Solving the School Bus Routing Problem with Bus Stop Selection. Mathematics.

Chen, D., Kallsn, H.A., Snider, R.C. (1988), School bus routing and scheduling: an expert system approach. Computers and Industrial Engineering 15, 179–183.

Chen, X., Kong, Y., Dang, L., Hou, Y., and Ye, X. (2015). Exact and metaheuristic approaches for a bi-objective school bus scheduling problem. PLoS One, 10(7), e0132600.

Corberán, A., Fernández, E., Laguna, M., Martí, R. (2002). Heuristic solutions to the problem of routing school buses with multiple objectives. Journal of Operational Research Society 53, 427–435.

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

Cuervo, D. P., Goos, P., Sörensen, K., and Arráiz, E. (2014). An iterated local search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research, 237(2), 454-464.

Chalkia, E., Salanova Grau, J., Bekiaris, E., Ayfandopoulou, G., Ferarini,C., Mitsakis, E.(2016). Safety Bus Routing for the Transportation of Pupils to School. Traffic Safety, Volume 4, Chapter 18.

G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12:568–581, 1964.

Cordeau, J., Laporte, G. (2007). The dial-a-ride problem: models and algorithms. Annals of Operations Research 153, 1,29–46.

Dube, P. (2017). The route of the problem. Ombudsman of Ontario. Retrieved from. https ://www.ombudsman.on.ca/resources/reports-and-case-summaries/reports-on-in vestigations/2017/the-route-of-the-problem.

Daojun Han., Hongzhen Xiang., Yi Xie.(2022). Solving multi depot school bus routing problem based on simulated annealing algorithm. Proceedings Volume 12176, International Conference on Algorithms, Microchips and Network.

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

Desrosiers, J., Commerciales, É. D. H. É. (1986a). Transports, U. D. M. C. D. R. S. L. and Operationally, U. D. M. D. D. I. E. D. R. TRANSCOL: A multi-period school bus routing and scheduling system, Montréal: Université de Montréal, Centre de recherche sur les transports.

Desrosiers, J. (1980). Transports U. D. M. C. D. R. S. L. and Opérationnelle, U. D. M. D. D. I. E. D. R. An Overview of School Busing System, Montréal: Université de Montréal, Centre de recherche sur les transports.

Desrosiers, J., Ferland, JA., Rousseau, JM., Lapalme, G., Chapleau L .(1986). TRANSCOL:a multi-period school bus routing and scheduling system. In: Swersey AJ, I EJ, editors. Delivery of Urban Services - TIMS Studies in the Management Sci- ences, 22; p. 47–71.

Desrosiers, J., Ferland, J., Rousseau, J ., Lapalme, G ., Chapleau, L . (1981). An overview of a school busing system. In: Jaiswal N, editor. Scientific Management of Trans- port Systems. North-Holland: Amsterdam; p. 235–43.

De Siqueira, V. S., da Silva, E. N., da Silva, R. V. S., and Rocha, M. L. (2016). Implementation of the metaheuristic grasp applied to the school bus routing problem. International Journal of e-Education, e-Business, e-Management and e-Learning, 6(2), 137.

De Souza Lima, F. M., Pereira, D. S., da Conceição, S. V., and de Camargo, R. S. (2017). A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads. 4OR, 15(4), 359-386.

Euchi, J., and Mraihi, R. (2012). The urban bus routing problem in the Tunisian case by the hybrid artificial ant colony algorithm. Swarm and Evolutionary Computation, 2:15–24.

Ellegood, W. A., Campbell, J. F., and North, J. (2015). Continuous approximation models for mixed load school bus routing. Transportation Research Part B: Methodological, 77, 182-198.

Ellegood, W.A., Solomon, S., North, J., Campbell , JF. (2019), School bus routing problem: Contemporary trends and research directions, Omega,volume (95). Elsevier.

Eguizábal, S. E., Berodia, J. L. M., Portilla, Á. I., and Ponce, J. B. (2018). Optimization model for school transportation design based on economic and social efficiency. Transport Policy.

Fügenschuh, A. (2009). Solving a school bus scheduling problem with integer programming. European Journal of Operational Research, 193(3), 867–884.

Fallah, Mohammad Saied., Talarica, Luca and Sajadifar, Mehdi. (2017). Iterated Local Search algorithm with Strategic Oscillation for School Bus Routing Problem with Bus Stop Selection. International Journal of Supply and Operations Management.

Feo, T.A., Resende, M.G.C. (1995). Greedy randomized adaptive search procedures. Journal of Global Optimization, 6, 109–133.

Falkmer, T., Gregersen, N.P. (2002). Perceived risk among parents concerning the travel situation for children with disabilities. Accid. Anal. Prev. 34 (4), 553–562.

Faulkner, G.E.J., Buliung, R.N., Flora, P.K., Fusco, C. (2009). Active school transport, physical activity levels and body weight of children and youth: a systematic review. Prev. Med. (Baltim) 48, 3–8. **https://doi.org/10.1016/j.ypmed.2008.10.017**.

Falkmer, M. (2004) The transport mobility situation for children with autism spectrum disorders. Scand. J. Occup. Ther. 11 (2), 90–100.

Falkmer, T., Horlin, C., Dahlman, J., Dukic, T., Barnett, T., Anund, A. (2014). Usability of the SAFEWAY2SCHOOL system in children with cognitive disabilities. Eur. Transport Res. Rev. 6 (2), 127–137.

Gavish, B., and Shlifer, E. (1979). An approach for solving a class of transportation scheduling problems. European Journal of Operational Research, Vol. 3, pp. 122-134.

Gawande, P. V., and Lokhande, S. V. (2018). Improving efficiency of school bus routing using AI based on bio-inspired computing: A Survey.

Goeke, D., and Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles', European Journal of Operational Research 245 (1), 81 – 99. 13, 41, 65, 66.

Graham, D., Nuttle, H.L.W. (1986). A comparison of heuristics for a school bus scheduling problem. Transportation Research B 20 (2), 175–182.

Hargroves, B.T., Demetsky, M.J. (1981), A computer assisted school bus routing strategy: a case study. Socio-Economic Planning Sciences 15 (6), 341–345.

Hansen, P., and Mladenovic, N. (1999). An introduction to variable neighborhood search. In Meta-heuristics, pp. 433458). Springer US.

Hernan Caceresa., Rajan Battaa., Qing Hea. (2015),Special Need Students School Bus Routing: Consideration for Mixed Load and Heterogeneous Fleet.

Hou,Y., Dang, L., Dong, W., Kong, Y.(2020). A Metaheuristic Algorithm for Routing School Buses with Mixed Load. IEEE Access.

Hiermann, G., Puchinger, J., Ropke, S. .,and Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations', European Journal of Operational Research 252 (3), 995 –1018.

Herminia I. Calvete., Carmen Galé., José A. Iranzo., Paolo Toth. (2023). The school bus routing problem with student choice: a bilevel approach and a simple and effective metaheuristic. International Transaction in Operational Research.

Huang Xiong., Hu Zuxiang., Zhou Bo.(2022). School bus path planning problem based on improved simulatedannealing algorithm. IPEC '22: Proceedings of the 3rd Asia-Pacific Conference on Image Processing, Electronics and Computers**.**

Irnich, S., Schneider, M., Vigo, D. (2014). Four variants of the vehicle routing problem, in: Toth, P., Vigo, D. (Eds.), Vehicle Routing: Problems, Methods, and Applications. MOS-SIAM Series on Optimization, Philadelphia, pp. 241–271.

Ikeda, E., Hinckson, E., Witten, K., Smith, M. (2018). Associations of children's active school travel with perceptions of the physical environment and characteristics of the social environment: a systematic review. Health Place 54, 118–131. **https://doi.org/10.1016/j.healthplace.2018.09.009**.

Iman Dayarian., Teodor Gabriel., Crainic Michel., Gendreau Walter Re. (2013). An Adaptive Large Neighborhood Search Heuristic for a Multi-Period Vehicle Routing Problem.

Karami, Z., Kashef, R. (2020). Smart Transportation Planning: Data, Models, and Algorithms. Transportation Engineering, Volume 2, December 2020, 100013.

Kinable, J., Spieksma, F. C., and Vanden Berghe, G. (2014). School bus routing—a column generation approach. International Transactions in Operational Research, 21(3), 453-478.

Kang, M., Kim, S. K., Felan, J. T., Choi, H. R., and Cho, M. (2015). Development of a genetic algorithm for the school bus routing problem. International Journal of Software Engineering and Its Applications, 9(5), 107–126

Kim, BI., Kim, S., Park, J. (2012). A school bus scheduling problem. Eur J operation Res;218(2):577–85.

Kirkpatrick, S., C. D. Gelatt, Jr., M. P. Vecchi. (1983). Optimization by simulated annealing. Science 220(4598) 671–680.

Lima, F. M. S., Pereira, D. S., Conceição, S. V., and Nunes, N. T. R. (2016). A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context. Expert Systems with Applications, 56, 320-334.

Lima, F. M. d. S., Pereira, D. S., da Conceio, S. V., de Camargo, R. S. (2017). A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads. 4or, 1–28.684.

Levin, MW., Boyles, SD. (2016). Summary: Improving bus routing for KIPP charter schools. (2016), Interfaces;46(2):196–9.

Li, L., Fu, Z., (2002). The school bus routing problem: a case study. Journal of the Operational Research Society 53, 552–558.3:75-85.

Lenstra, J., A. H. G. Rinnooy Kan. (1981). "Complexity of vehicle routing and scheduling problems", Networks 11 221–227.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2010). Iterated local search: Framework and applications. Hand book of Metaheuristics. Springer.

Lourenço, H., Martin, O., and Stützle, T. (2003), Iterated local search. In F. Glover, & G. Kochenberger (Eds.). Handbook of metaheuristics (pp. 321–353). Boston: Kluwer.

Li, M., Chow, J.Y.J.(2021), A school bus routing problem with a mixed ride, mixed load, and heterogeneous fleet, Transportation Research Record.

Lalitkumar Patil, A., V. Rane, B. (2021). Analyzing the Importance of Logistics in Supply Chain Management. Palarch's Journal Of Archaeology Of Egypt/Egyptology 18(7). ISSN 1567-214x.

Lissa Melis., Kenneth Sörensen. (2022). the static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment. International Transaction in Operational Research.

Linda Rothman., Brent Hagel., Andrew Howard., Marie Soleil Cloutier. (2021). Active school transportation and the built environment across Canadian cities: Findings from the child active transportation safety and the environment (CHASE) study. Preventive Medicine.

Leticia Caldas. , Rafael Martinelli1., and Bruno Rosa. (2022). Solving a School Bus Routing Problem in Rural Areas: An Application in Brazil. International Conference on Computational Logistics.

Larouche, R., Mammen, G., Rowe, D.A., Faulkner, G. (2018). Effectiveness of active school transport interventions: a systematic review and update. BMC Public Health 18, 1–18. **https://doi.org/10.1186/s12889-017-5005-1**.

Luattman, K., Friman, M., Olsson, L.E., 2020. Restricted car-use and perceived accessibility. Transp. Res. Part D: Transport Environ. 78. Lim, A., Zhang, Z., Qin, H., 2017. Pick-up and delivery service with manpower planning in Hong Kong public hospitals. Transp. Sci. 51, 688–705.

Maciel Silva, C., Sarubbi, J., Fonseca Silva, D., Porto, M., and Nunes, N. (2015). A mixed load solution for the rural school bus routing problem. In Proceedings of the IEEE 18th international conference on Intelligent transportation systems (ITSC), 2015 (pp. 1940–1945).

Mamoona, H., NZ, J., Bushra, H., and Ghufran, A. (2020). Emerging Smart Logistics and Transportation Using IoT and Blockchain. IEEE Internet of Things Magazine (Volume: 3, Issue: 2, June 2020).

Martinez, L. M., and Viegas, J. M. (2011). Design and Deployment of an Innovative School Bus Service in Lisbon. Procedia Social and Behavioral Sciences, pages 120–130.

McDonald, N.C. (2007). Active transportation to school: trends among US schoolchildren, 1969–2001. Am. J. Prev. Med. 32, 509–516. https://doi.org/10.1016/j. amepre.2007.02.022.

Miranda, D. M., de Camargo, R. S., Conceição, S. V., Porto, M. F., and Nunes, N. T. (2018). A multi-loading school bus routing problem. Expert Systems with Applications, 101, 228-242.

Mokhtari, N. A., and Ghezavati, V. (2018). Integration of efficient multi-objective ant-colony and a heuristic method to solve a novel multi-objective mixed load school bus routing model. Applied Soft Computing, 68, 92-109.

McAndrews, C., Pollack, K.M., Berrigan, D., Dannenberg, A.L., Christopher, E.J.(2017). Understanding and improving arterial roads to support public health and transportation goals. Am. J. Public Health 107, 1278–1282. https://doi.org/10.2105/AJPH.2017.303898.

Maibach, E., Steg, L., Anable, J. (2009). Promoting physical activity and reducing climate change: Opportunities to replace short car trips with active transportation. Prev. Med. (Baltim) 49, 326–327.

Mitra, R. (2013). Independent mobility and mode choice for school transportation: a review and framework for future research. Transp. Rev. 33, 21–43. https://doi. org/10.1080/01441647.2012.743490.

Martens, K. (2016). Chapter 10: Seeking Transportation Justice, in: Transport Justice: Designing Fair Transportation Systems. Routledge, New York. Martens, K., Golub, A., Robinson, G., 2012. A justice-theoretic approach to the distribution of transportation benefits: Implications for transportation planning practice in the United States. Transp. Res. Part A Policy Pract. 46, 684–695.

Monique Sciortino., Rhyd Lewis., Jonathan Thompson. (2023). A School Bus Routing Heuristic Algorithm Allowing Heterogeneous Fleets and Bus Stop Selection. SN Computer Science (2023) 4:74.

Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. Journal of the ACM 7 (4), 326–329.

Newton, R. M., and Thomas, W. H. (1969). Design of school bus routes by computer. Socio-Economic Planning Sciences, 3(1), 75-85.

Newton, R. M., and Thomas, W. H. (1974). Bus routing in a multi-school system. *Computers & Operations Research*, Vol.1, pp. 213-222.

Nagata, Y., Br¨aysy, O. (2009). "Edge assembly-based memetic algorithm for the capacitated vehicle routing Problem". Networks, 54(4):205–215.

Oluwadare, SA., Oguntuyi, IP., Nwaiwu JC. (2018). Solving school bus routing problem using genetic algorithm-based model. Int J Intell Syst Appl;10(3):50–8 .

Orejuela Cabrera, J., Londoñob, M., Chud Pantoja, V.L. (2021). School bus routing problem considering affinity among children. Decision Science Letters. 535–548.

Park, J., and Kim, B. I. (2010). The school bus routing problem: A review. European Journal of operational research, Vol. 202, pp. 311-319.

Park, J., Tae, H., and Kim, B. I. (2012). A post-improvement procedure for the mixed load school bus routing problem. European Journal of Operational Research, 217(1), 204-213.

Pacheco, J., Caballero, R., Laguna, M., and Molina J. (2013). Bi-objective bus routing: an application to school buses in rural areas. Transportation Science, Vol.47, pp. 397-411.

Pacheco, J., Martí, R. (2006). Tabu search for a multi-objective routing problem. Journal of the Operational Research Society 57, 29–37.

Pérez-Rodríguez, R., Hernández-Aguirre, A. (2016), Probability model to solve the school bus routing problem with stops selection. Int J Comb Optimize Prob Inform;7(1):30–9.

Parvasi, S. P., Mahmoodjanloo, M., and Setak, M. (2017). A bi-level school bus routing problem with bus stops selection and possibility of demand outsourcing. Applied Soft Computing, 61, 222-238.

Parvasi, S., Tavakkoli-Moghaddam, R., Taleizadeh, A., Soveizy,M.(2019). A bilevel bi-objective mathematical model for stop location in a school bus routing problem. IFAC-PapersOnLine 52, 1120–1125.

Ropke, S., and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science, 40(4), 455-472.

Rothman, L., Macpherson, A.K., Ross, T., Buliung, R.N. (2018). The decline in active school transportation (AST): A systematic review of the factors related to AST and changes in school transport over time in North America. Prev. Med. (Baltim) 111, 314–322. https://doi.org/10.1016/j.ypmed.2017.11.018.

Ribeiro, G. M., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Computers & operations research, 39(3), 728-735.

Ruiz, P.A ., Sanz, J.M , Martín J.G. (2015). Mixed trips in the school bus routing problem with public subsidy. In: Enhancing Synergies in a Collaborative Environment. International Publishing: Springer; p. 105–13

Riera-ledesma, J., Salazar-gonzalez, J. J. (2013). A column generation approach for a school bus routing problem with resource constraints. Computers & Operations Research, Vol. 40, pp. 566-583.

Rodríguez-Parra, G. R., Guerrero, W. J., and Sarmiento-Lepesqueur, A. (2017). Cooperation strategies featuring optimization in the school transportation system in Bogota. Dyna, 84(202), 164-174.

Rego, C. (2001), Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. Parallel Computing, 27(3):201–222.

Saaty, T. L. (1987). Risk—its priority and probability: the analytic hierarchy process. Risk Analysis, 7(2), 159-172. http://dx.doi.org/10.1111/j.1539-6924.1987.tb00980.x.

Savas, E. (1978). On equity in providing public services. Management Science 24, 800–808.

Spada, M., Bierlaire, M., and Liebling T.M., (2005). Decision-aiding methodology for the school bus routing and scheduling problem. Transportation Science, 39(4), pp.477-490.

Schittekat, P., Sevaux, M., and Sörensen, K. (2006). A mathematical formulation for a school bus routing problem. In Service Systems and Service Management, 2006 International Conference on (Vol. 2, pp. 1552-1557). IEEE.

Schittekat, P., Kinable J., Sörensen, K., Sevaux, M., Spieksma F., and Springael J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research,* Vol. 229, pp. 518-528.

Subramanian, A., Drummond, L. M. D. A., Bentes C., Ochi L. S., and Farias R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research, Vol. 37, pp. 1899-1911.

Silva, C.M , Sarubbi, J.F , Silva, D.F , Porto, M.F , Nunes N.T.R .(2015), A mixed load solution for the rural school bus routing problem. In: Intelligent Transportation Systems (ITSC), IEEE 18th International Conference. IEEE; 2015. p. 1940–5 .

Shafahi, A., Wang, Z., Haghani, A. (2017), Solving the school bus routing problem by maximizing trip compatibility. Transp Res Rec J Transp Res Board (2667):17–27.

Shafahi, A., Wang, Z., Haghani, A. (2018), A matching-based heuristic algorithm for school bus routing problems.Computer Science - Data Structures and Algorithms

Shafahi, A., Aliari, S., Haghani, A. (2018). BALANCED SCHEDULING OF SCHOOL BUS TRIPS USING A PERFECT MATCHING HEURISTIC, Journal of the Transportation Research Board

Shafahi, A., Wang, Z ., Haghani, A . (2018). Speed Route: Fast, efficient solutions for school bus routing problems. Transp Res Part B Methodol 117:473–93 .

Swersey, A. J., and Ballard, W. (1984). Scheduling school buses. Management Science, Vol. 30, pp. 844-853.

Shimizu, Y., and Sakaguchi, T. (2013). Generalized vehicle routing problem for reverse logistics aiming at low carbon transportation. Industrial Engineering and Management Systems, 12(2), 161-170.

Shimizu, Y., and Sakaguchi, T. (2014). A hierarchical hybrid meta-heuristic approach to coping with large practical multi-depot VRP. Industrial Engineering & Management Systems, 13(2), 163-171.

Souza Lima, F. M., Pereira, D. S., Conceição, S. V., and Ramos Nunes, N. T. (2016). A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context", Expert Systems with Applications, vol.56, pp320–334, 2016.

Sciortino, M., Lewis, R., Thompson, J.(2021). A Heuristic Algorithm for School Bus Routing with Bus Stop Selection.

Statistics Canada, 2020. Persons living in low-income neighbourhoods [WWW Document]. URL https://www12.statcan.gc.ca/nhs-enm/2011/as-sa/99-014-x/99- 014-x2011003_3-eng.cfm (accessed 12.16.21).

Statistics Canada, 2017. Visible Minority and Population Group Reference Guide, Census of Population, 2016 [WWW Document]. URL https://www12.statcan.gc.ca/ census-recensement/2016/ref/guides/006/98-500-x2016006-eng.cfm (accessed 12.16.21).

Turkeš, R., Sörensen K., Hvattum, L.(2021). Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search, European Journal of Operational Research, Vol 292, Issue 2, 16 July 2021, Pages 423-442.

Talarico, L., Sörensen, K., and Springael, J. (2015). Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. European Journal of Operational Research, Vol. 244, pp. 457-470.

Toth, P., and Vigo, D. (2002). Vehicle Routing Problem SIAM Monographs on Discrete Mathematics and Applications, Vol. 9. SIAM: Philadelphia, PA.

Timothy Ross., Patrick Bilas., Ronald Buliung ., Ahmed El-Geneidy.(2020). A scoping review of accessible student transport services for children with disabilities. Transport policy.

Toth, P., and Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. Informs Journal on computing, Vol. 15, pp. 333-346.

Thangiah, S.R., Nygard, K.E. (1992). School bus routing using genetic algorithms. In: Proceedings of the SPIE Conference on Applications of Artificial Intelligence X: Knowledge-Based Systems, Orlando, Florida. doi:1-.1117/12.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation science, 31(2):170–186 -cross-exchange.

Verderber, W. J. (1974). Automated pupil transportation. Computers & Operations Research, Vol.1, pp. 235-245.

Vijayagovry Kunaratnam ., Naomi Schwartz ., Andrew Howard ., Raktim Mitra ., Natasha Saunders ., Marie-Soleil Cloutier ., Alison Macpherson ., Pamela Fuselli ., Linda Rothman.(2022). Equity, walkability, and active school transportation in Toronto, Canada: A cross-sectional study. Transportation Research Part D.

Wong, B.-Y.-M., Faulkner, G., Buliung, R. (2011). GIS measured environmental correlates of active school transport: A systematic review of 14 studies. Int. J. Behav. Nutr. Phys. Act. 8, 22. **https://doi.org/10.1186/1479-5868-8-39**.

Wang, Z., Shafahi A., Haghani, A. (2017). AN ITERATIVE SCHOOL DECOMPOSITION ALGORITHM FOR SOLVING THE MULTI-SCHOOL BUS ROUTING AND SCHEDULING PROBLEM.

Yao, B., Cao, Q., Wang, Z., Hu, P., Zhang, M., and Yu, B. (2016). A two-stage heuristic algorithm for the school bus routing problem with mixed load plan. Transportation Letters, 8(4), 205-219.

Yigit, T., Unsal, O., and Deperlioglu, O. (2018). Using the metaheuristic methods for real-time optimization of dynamic school bus routing problem and an application. International Journal of Bio-Inspired Computation, 11(2), 123-133.

Yane Hou, Bingbing Liu., Lanxue Dang., Wenwen He and Wenbo Gu. (2022). A Local Search-based Metaheuristic Algorithm Framework for the School Bus Routing Problem. Engineering Letters, 30:1, EL_30_1_03.

Appendices

Appendix 1

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	N-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
1	5	25	25	5	141.01	141.01	141.01	142.26	0.000%	0.000%	0.89%	0.89%	1.06
2	5	25	50	5	161.62	161.62	162.12	162.22	0.309%	0.309%	0.37%	0.37%	1.52
3	5	25	25	10	182.14	182.14	182.14	182.35	0.000%	0.000%	0.12%	0.12%	1.83
4	5	25	50	10	195.80	195.80	195.80	196.22	0.000%	0.000%	0.21%	0.21%	1.61
5	5	25	25	20	111.65	111.65	111.93	112.36	0.251%	0.251%	0.64%	0.64%	1.79
6	5	25	50	20	103.18	103.18	103.18	103.39	0.000%	0.000%	0.20%	0.20%	1.81
7	5	25	25	40	7.63	7.63	7.87	8.02	3.098%	3.098%	5.11%	5.11%	1.33
8	5	25	50	40	25.64	25.64	25.83	26.12	0.741%	0.741%	1.87%	1.87%	1.75
9	5	50	25	5	286.68	286.68	286.68	288.13	0.000%	0.000%	0.51%	0.51%	1.81
10	5	50	50	5	197.20	197.20	199.41	201.38	1.121%	1.121%	2.12%	2.12%	1.43
11	5	50	25	10	193.55	193.55	193.55	198.12	0.000%	0.000%	2.36%	2.36%	1.89
12	5	50	50	10	215.86	215.85	215.86	217.20	0.005%	0.000%	0.63%	0.62%	1.79
13	5	50	25	20	130.53	130.53	131.85	133.11	1.014%	1.014%	1.98%	1.98%	3.14
14	5	50	50	20	96.26	96.26	98.56	100.02	2.389%	2.389%	3.91%	3.91%	2.97
15	5	50	25	40	12.89	12.89	13.94	13.98	8.131%	8.131%	8.46%	8.46%	3.56
16	5	50	50	40	30.24	30.24	30.63	30.93	1.283%	1.283%	2.27%	2.27%	2.85
17	5	100	25	5	360.35	360.35	360.35	372.54	0.000%	0.000%	3.38%	3.38%	2.82
18	5	100	50	5	304.23	304.23	309.94	312.25	1.877%	1.877%	2.64%	2.64%	2.73
19	5	100	25	10	294.21	294.21	302.57	303.12	2.843%	2.842%	3.03%	3.03%	4.34
20	5	100	50	10	229.41	229.41	232.45	235.25	1.326%	1.326%	2.55%	2.55%	3.76
21	5	100	25	20	134.95	134.95	137.54	139.90	1.917%	1.917%	3.67%	3.67%	6.90
22	5	100	50	20	144.41	144.41	144.48	146.20	0.048%	0.048%	1.24%	1.24%	3.37
23	5	100	25	40	58.95	58.95	58.95	59.25	0.000%	0.000%	0.51%	0.51%	9.22
24	5	100	50	40	39.44	39.44	41.90	42.84	6.248%	6.248%	8.63%	8.63%	6.89
25	10	50	25	5	242.85	242.85	242.85	247.76	0.000%	0.000%	2.02%	2.02%	3.72
26	10	50	50	5	282.12	282.12	285.34	287.26	1.142%	1.142%	1.82%	1.82%	3.47

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	N-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
27	10	50	25	10	244.54	244.54	252.51	255.12	3.260%	3.260%	4.33%	4.33%	5.32
28	10	50	50	10	288.33	283.33	297.32	297.65	4.937%	3.117%	5.05%	3.23%	3.82
29	10	50	25	20	108.98	108.98	110.69	112.58	1.567%	1.567%	3.30%	3.30%	4.89
30	10	50	50	20	157.48	157.48	159.76	160.92	1.448%	1.448%	2.18%	2.18%	4.77
31	10	50	25	40	32.25	32.25	34.01	34.87	5.451%	5.451%	8.14%	8.14%	5.20
32	10	50	50	40	36.66	36.66	38.85	39.07	5.983%	5.983%	6.57%	6.57%	4.93
33	10	100	25	5	403.18	403.18	404.36	410.51	0.293%	0.293%	1.82%	1.82%	2.20
34	10	100	50	5	296.53	296.53	297.29	298.12	0.256%	0.256%	0.54%	0.54%	2.45
35	10	100	25	10	388.87	388.87	395.01	396.62	1.580%	1.580%	1.99%	1.99%	8.50
36	10	100	50	10	294.80	294.80	306.26	306.61	3.889%	3.887%	4.01%	4.00%	9.47
37	10	100	25	20	178.28	178.28	178.28	181.04	0.000%	0.000%	1.55%	1.55%	9.53
38	10	100	50	20	175.96	175.96	180.60	182.47	2.636%	2.636%	3.70%	3.70%	10.66
39	10	100	25	40	57.50	57.50	60.27	61.83	4.816%	4.816%	7.53%	7.53%	11.03
40	10	100	50	40	31.89	31.89	32.19	33.63	0.951%	0.951%	5.46%	5.46%	11.18
41	10	200	25	5	735.27	735.27	736.92	757.55	0.224%	0.224%	3.03%	3.03%	10.65
42	10	200	50	5	512.16	506.06	512.32	528.97	1.237%	0.031%	4.53%	3.28%	9.71
43	10	200	25	10	513.00	513.00	523.30	529.48	2.008%	2.008%	3.21%	3.21%	31.19
44	10	200	50	10	475.21		484.17	486.82		1.885%		2.4%	14.37
45	10	200	25	20	347.29		362.12	367.81		4.271%		5.6%	26.54
46	10	200	50	20	217.46		217.69	221.39		0.106%		1.8%	25.36
47	10	200	25	40	102.93		104.42	105.84		1.444%		2.7%	39.43
48	10	200	50	40	55.05		60.01	60.93		9.010%		9.7%	16.76
49	20	100	25	5	520.24		526.54	529.18		1.211%		1.7%	11.09
50	20	100	50	5	420.64		421.09	425.12		0.106%		1.1%	6.55
51	20	100	25	10	422.21		426.91	428.40		1.113%		1.4%	10.23
52	20	100	50	10	360.86		364.05	379.58		0.883%		4.9%	9.45
53	20	100	25	20	245.17		246.45	251.69		0.522%		2.6%	11.38
54	20	100	50	20	185.06		190.28	192.00		2.820%		3.6%	9.13

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	N-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
55	20	100	25	40	52.52		52.85	53.60		0.631%		2.0%	12.73
56	20	100	50	40	19.05		19.65	20.16		3.150%		5.5%	29.14
57	20	200	25	5	903.84		923.81	926.72		2.209%		2.5%	13.76
58	20	200	50	5	485.65		497.83	500.31		2.507%		2.9%	30.65
59	20	200	25	10	616.93		626.48	627.86		1.548%		1.7%	32.87
60	20	200	50	10	462.31		479.73	483.26		3.768%		4.3%	22.19
61	20	200	25	20	373.21		381.15	384.45		2.126%		2.9%	57.18
62	20	200	50	20	250.75		251.36	257.13		0.241%		2.5%	30.16
63	20	200	25	40	93.01		95.05	95.52		2.193%		2.6%	77.44
64	20	200	50	40	45.40		46.18	48.25		1.721%		5.9%	38.64
65	20	400	25	5	1323.35		1376.13	1390.65		3.988%		4.8%	293.12
66	20	400	50	5	733.54		734.12	750.28		0.079%		2.2%	42.21
67	20	400	25	10	975.12		990.23	1002.58		1.549%		2.7%	194.19
68	20	400	50	10	614.67		635.65	641.25		3.413%		4.1%	88.19
69	20	400	25	20	763.76		790.02	795.39		3.438%		4.0%	177.37
70	20	400	50	20	298.47		309.32	310.25		3.636%		3.8%	105.13
71	20	400	25	40	239.58		242.56	246.32		1.245%		2.7%	354.02
72	20	400	50	40	84.49		89.56	90.06		6.001%		6.2%	142.75
73	40	200	25	5	831.94		846.71	852.12		1.775%		2.4%	80.27
74	40	200	50	5	593.35		608.22	617.20		2.507%		3.9%	55.95
75	40	200	25	10	728.44		734.98	737.59		0.898%		1.2%	902.38
76	40	200	50	10	481.05		498.53	499.02		3.634%		3.6%	129.50
77	40	200	25	20	339.75		341.19	348.51		0.422%		2.5%	192.11
78	40	200	50	20	273.88		276.25	279.49		0.865%		2.0%	56.27
79	40	200	25	40	76.77		78.25	79.98		1.928%		4.0%	151.78
80	40	200	50	40	58.46		58.56	59.51		0.171%		1.8%	95.20
81	40	400	25	5	1407.05		1431.90	1458.52		1.766%		3.5%	430.71
82	40	400	50	5	858.80		885.80	889.55		3.144%		3.5%	773.02

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	N-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
83	40	400	25	10	891.02		899.49	925.75		0.950%		3.8%	580.62
84	40	400	50	10	757.42		776.30	790.72		2.493%		4.2%	469.93
85	40	400	25	20	586.29		600.38	605.92		2.403%		3.2%	897.30
86	40	400	50	20	395.95		404.51	411.01		2.161%		3.7%	323.68
87	40	400	25	40	195.33		198.26	201.99		1.500%		3.3%	1491.45
88	40	400	50	40	70.77		73.95	74.20		4.493%		4.6%	690.38
89	40	800	25	5	2900.14		3005.33	3102.91		3.627%		6.5%	4023.58
90	40	800	50	5	1345.70		1374.92	1377.93		2.171%		2.3%	1795.17
91	40	800	25	10	2200.57		2308.45	2347.69		4.902%		6.3%	4930.25
92	40	800	50	10	1025.16		1045.23	1068.89		1.958%		4.1%	5125.67
93	40	800	25	20	1404.16		1458.25	1459.94		3.852%		3.8%	4142.30
94	40	800	50	20	616.58		630.19	632.49		2.207%		2.5%	4246.67
95	40	800	25	40	396.92		405.26	408.96		2.101%		2.9%	4332.19
96	40	800	50	40	200.94		207.45	211.25		3.240%		4.9%	3836.31
97	80	400	25	5	1546.23		1579.37	1593.47		2.143%		3.0%	1225.87
98	80	400	50	5	1048.56		1071.65	1085.19		2.202%		3.4%	681.78
99	80	400	25	10	1216.74		1258.94	1273.16		3.468%		4.4%	2470.60
100	80	400	50	10	760.61		768.92	781.97		1.093%		2.7%	745.81
101	80	400	25	20	565.49		581.38	588.23		2.811%		3.9%	1790.90
102	80	400	50	20	372.05		383.18	383.48		2.992%		3.0%	1281.41
103	80	400	25	40	131.75		133.96	138.45		1.677%		4.8%	1356.89
104	80	400	50	40	95.84		98.25	99.17		2.515%		3.4%	4020.56

Appendix 2

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	I-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
1	5	25	25	5	141.01	141.01	141.01	144.10	0.000%	0.000%	2.19%	2.19%	1.05
2	5	25	50	5	161.62	161.62	161.62	162.19	0.000%	0.000%	0.36%	0.36%	1.73
3	5	25	25	10	182.14	182.14	182.14	183.48	0.000%	0.000%	0.74%	0.74%	2.02
4	5	25	50	10	195.80	195.80	200.01	200.76	2.150%	2.150%	2.53%	2.53%	1.76
5	5	25	25	20	111.65	111.65	115.31	116.54	3.278%	3.278%	4.38%	4.38%	1.98
6	5	25	50	20	103.18	103.18	105.35	106.93	2.103%	2.103%	3.63%	3.63%	2.03
7	5	25	25	40	7.63	7.63	7.93	8.01	3.932%	3.932%	5.01%	5.01%	1.73
8	5	25	50	40	25.64	25.64	27.06	27.58	5.538%	5.538%	7.58%	7.58%	1.82
9	5	50	25	5	286.68	286.68	289.95	291.17	1.141%	1.141%	1.56%	1.56%	1.73
10	5	50	50	5	197.20	197.20	200.16	201.50	1.501%	1.501%	2.18%	2.18%	1.62
11	5	50	25	10	193.55	193.55	195.19	197.85	0.847%	0.847%	2.22%	2.22%	1.86
12	5	50	50	10	215.86	215.85	215.85	218.35	0.000%	-0.005%	1.16%	1.15%	2.07
13	5	50	25	20	130.53	130.53	132.25	133.84	1.318%	1.318%	2.53%	2.53%	3.47
14	5	50	50	20	96.26	96.26	97.80	97.91	1.601%	1.601%	1.71%	1.71%	3.34
15	5	50	25	40	12.89	12.89	13.36	13.69	3.637%	3.637%	6.19%	6.19%	3.94
16	5	50	50	40	30.24	30.24	32.02	32.45	5.879%	5.879%	7.30%	7.30%	3.20
17	5	100	25	5	360.35	360.35	360.35	368.15	0.000%	0.000%	2.16%	2.16%	3.09
18	5	100	50	5	304.23	304.23	313.72	315.75	3.120%	3.120%	3.79%	3.79%	2.62
19	5	100	25	10	294.21	294.21	306.94	309.12	4.329%	4.328%	5.07%	5.07%	4.73
20	5	100	50	10	229.41	229.41	232.19	237.59	1.212%	1.212%	3.56%	3.56%	4.13
21	5	100	25	20	134.95	134.95	139.41	140.23	3.305%	3.305%	3.92%	3.92%	6.81
22	5	100	50	20	144.41	144.41	149.17	152.95	3.295%	3.295%	5.92%	5.92%	3.23
23	5	100	25	40	58.95	58.95	58.95	60.05	0.000%	0.000%	1.87%	1.87%	9.09
24	5	100	50	40	39.44	39.44	41.12	41.90	4.260%	4.260%	6.24%	6.24%	7.76
25	10	50	25	5	242.85	242.85	242.85	247.25	0.000%	0.000%	1.81%	1.81%	4.76
26	10	50	50	5	282.12	282.12	286.30	289.37	1.482%	1.482%	2.57%	2.57%	3.37

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	I-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
27	10	50	25	10	244.54	244.54	244.68	249.63	0.056%	0.056%	2.08%	2.08%	4.93
28	10	50	50	10	288.33	283.33	288.33	289.09	1.765%	0.000%	2.03%	0.26%	3.97
29	10	50	25	20	108.98	108.98	110.34	112.19	1.250%	1.250%	2.94%	2.94%	4.84
30	10	50	50	20	157.48	157.48	160.28	161.13	1.779%	1.779%	2.32%	2.32%	4.73
31	10	50	25	40	32.25	32.25	32.76	32.88	1.587%	1.587%	1.96%	1.96%	5.82
32	10	50	50	40	36.66	36.66	37.81	38.23	3.146%	3.146%	4.29%	4.29%	4.78
33	10	100	25	5	403.18	403.18	404.21	409.80	0.255%	0.255%	1.64%	1.64%	2.63
34	10	100	50	5	296.53	296.53	299.76	307.18	1.088%	1.088%	3.59%	3.59%	2.35
35	10	100	25	10	388.87	388.87	390.53	397.55	0.427%	0.427%	2.23%	2.23%	8.96
36	10	100	50	10	294.80	294.80	307.16	308.88	4.194%	4.192%	4.78%	4.78%	9.03
37	10	100	25	20	178.28	178.28	178.28	181.48	0.000%	0.000%	1.79%	1.79%	8.89
38	10	100	50	20	175.96	175.96	178.68	182.28	1.548%	1.548%	3.59%	3.59%	10.46
39	10	100	25	40	57.50	57.50	59.66	59.99	3.751%	3.751%	4.33%	4.33%	11.28
40	10	100	50	40	31.89	31.89	32.92	33.99	3.231%	3.231%	6.57%	6.57%	10.42
41	10	200	25	5	735.27	735.27	739.23	745.72	0.539%	0.539%	1.42%	1.42%	12.38
42	10	200	50	5	512.16	506.06	511.42	515.29	1.059%	-0.145%	1.82%	0.61%	9.16
43	10	200	25	10	513.00	513.00	518.36	522.29	1.045%	1.045%	1.81%	1.81%	33.38
44	10	200	50	10	475.21		489.71	490.84		3.051%		3.2%	14.05
45	10	200	25	20	347.29		360.99	361.56		3.945%		3.9%	30.96
46	10	200	50	20	217.46		225.17	229.25		3.545%		5.1%	29.17
47	10	200	25	40	102.93		109.19	111.13		6.084%		7.4%	44.06
48	10	200	50	40	55.05		56.92	57.99		3.392%		5.1%	18.03
49	20	100	25	5	520.24		535.19	535.70		2.874%		2.9%	12.61
50	20	100	50	5	420.64		420.64	432.13		0.000%		2.7%	6.83
51	20	100	25	10	422.21		437.38	439.89		3.594%		4.0%	11.23
52	20	100	50	10	360.86		368.17	375.02		2.025%		3.8%	9.62
53	20	100	25	20	245.17		245.17	247.63		0.000%		1.0%	12.38
54	20	100	50	20	185.06		190.36	194.07		2.864%		4.6%	8.65

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	I-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
55	20	100	25	40	52.52		53.92	55.12		2.666%		4.7%	14.23
56	20	100	50	40	19.05		19.36	20.57		1.627%		7.4%	30.18
57	20	200	25	5	903.84		924.01	930.06		2.231%		2.8%	15.14
58	20	200	50	5	485.65		497.39	497.51		2.417%		2.4%	34.19
59	20	200	25	10	616.93		621.84	626.44		0.797%		1.5%	35.55
60	20	200	50	10	462.31		476.09	476.46		2.981%		3.0%	24.06
61	20	200	25	20	373.21		373.21	380.91		0.000%		2.0%	53.96
62	20	200	50	20	250.75		259.68	264.21		3.561%		5.1%	32.88
63	20	200	25	40	93.01		97.92	98.21		5.279%		5.3%	83.13
64	20	200	50	40	45.40		45.40	46.80		0.000%		3.0%	36.53
65	20	400	25	5	1323.35		1343.98	1353.94		1.559%		2.3%	315.01
66	20	400	50	5	733.54		755.96	760.54		3.056%		3.6%	47.96
67	20	400	25	10	975.12		1000.30	1001.71		2.582%		2.7%	215.92
68	20	400	50	10	614.67		629.16	631.72		2.357%		2.7%	92.85
69	20	400	25	20	763.76		765.61	785.33		0.242%		2.7%	192.72
70	20	400	50	20	298.47		312.61	316.90		4.738%		5.8%	99.37
71	20	400	25	40	239.58		245.48	246.76		2.464%		2.9%	387.12
72	20	400	50	40	84.49		86.63	89.23		2.533%		5.3%	156.41
73	40	200	25	5	831.94		858.11	870.25		3.146%		4.4%	89.31
74	40	200	50	5	593.35		609.72	619.09		2.760%		4.2%	61.73
75	40	200	25	10	728.44		750.38	750.71		3.012%		3.0%	1002.42
76	40	200	50	10	481.05		503.17	505.05		4.598%		4.8%	138.96
77	40	200	25	20	339.75		353.30	353.12		3.987%		3.8%	210.45
78	40	200	50	20	273.88		275.46	278.04		0.577%		1.5%	62.70
79	40	200	25	40	76.77		77.34	80.35		0.737%		4.5%	160.79
80	40	200	50	40	58.46		59.12	61.78		1.129%		5.4%	101.98
81	40	400	25	5	1407.05		1441.25	1468.33		2.431%		4.2%	422.19
82	40	400	50	5	858.80		865.19	883.78		0.745%		2.8%	853.34

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	I-ILS						
							cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
83	40	400	25	10	891.02		909.31	916.30		2.053%		2.8%	623.18
84	40	400	50	10	757.42		771.12	772.44		1.809%		1.9%	533.15
85	40	400	25	20	586.29		587.12	596.07		0.142%		1.6%	963.29
86	40	400	50	20	395.95		396.39	397.77		0.111%		0.5%	355.63
87	40	400	25	40	195.33		200.35	203.10		2.570%		3.8%	1657.30
88	40	400	50	40	70.77		73.92	75.32		4.451%		6.0%	785.89
89	40	800	25	5	2900.14		2951.94	2965.10		1.786%		2.2%	4363.20
90	40	800	50	5	1345.70		1392.20	1411.06		3.455%		4.6%	2012.87
91	40	800	25	10	2200.57		2231.96	2254.61		1.426%		2.4%	5471.92
92	40	800	50	10	1025.16		1039.70	1053.44		1.419%		2.7%	5552.97
93	40	800	25	20	1404.16		1430.19	1487.48		1.854%		5.6%	4559.61
94	40	800	50	20	616.58		630.44	632.70		2.248%		2.5%	4765.21
95	40	800	25	40	396.92		397.21	409.71		0.073%		3.1%	4816.45
96	40	800	50	40	200.94		210.77	214.09		4.890%		6.1%	4256.84
97	80	400	25	5	1546.23		1578.36	1578.40		2.078%		2.0%	1347.76
98	80	400	50	5	1048.56		1086.21	1105.93		3.591%		5.2%	782.60
99	80	400	25	10	1216.74		1254.93	1273.44		3.139%		4.5%	2693.68
100	80	400	50	10	760.61		762.82	776.29		0.291%		2.0%	846.50
101	80	400	25	20	565.49		578.12	585.38		2.233%		3.4%	2012.93
102	80	400	50	20	372.05		377.34	387.22		1.422%		3.9%	1464.64
103	80	400	25	40	131.75		136.72	137.92		3.775%		4.5%	1564.08
104	80	400	50	40	95.84		101.07	101.29		5.457%		5.4%	4531.00

Appendix 3

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
1	5	25	25	5	141.01	141.01	141.01	144.98	0.00	0.00	2.82	2.82	1.06	141.01	145.28	0.00	0.00	3.03	3.03	1.04
2	5	25	50	5	161.62	161.62	164.38	164.44	1.71	1.71	1.74	1.74	1.48	163.78	166.19	1.34	1.34	2.83	2.83	1.35
3	5	25	25	10	182.14	182.14	186.32	187.29	2.29	2.29	2.83	2.83	1.74	182.14	188.13	0.00	0.00	3.29	3.29	1.78
4	5	25	50	10	195.80	195.80	195.80	196.09	0.00	0.00	0.15	0.15	1.56	195.80	198.62	0.00	0.00	1.44	1.44	1.70
5	5	25	25	20	111.65	111.65	111.65	115.18	0.00	0.00	3.16	3.16	1.81	112.98	115.09	1.19	1.19	3.08	3.08	1.70
6	5	25	50	20	103.18	103.18	105.89	106.90	2.63	2.63	3.60	3.60	1.91	103.18	105.47	0.00	0.00	2.22	2.22	1.98
7	5	25	25	40	7.63	7.63	7.89	7.98	3.41	3.41	4.59	4.59	1.50	7.75	7.78	1.57	1.57	1.97	1.97	1.45
8	5	25	50	40	25.64	25.64	26.69	26.98	4.10	4.10	5.23	5.23	1.45	26.95	27.01	5.11	5.11	5.34	5.34	1.48
9	5	50	25	5	286.68	286.68	286.68	292.34	0.00	0.00	1.97	1.97	1.68	286.68	293.35	0.00	0.00	2.33	2.33	1.65
10	5	50	50	5	197.20	197.20	197.20	200.07	0.00	0.00	1.46	1.46	1.57	204.93	208.87	3.92	3.92	5.92	5.92	1.50
11	5	50	25	10	193.55	193.55	193.55	194.29	0.00	0.00	0.38	0.38	1.82	197.27	199.93	1.92	1.92	3.30	3.30	1.70
12	5	50	50	10	215.86	215.85	215.85	218.89	0.00	0.00	1.41	1.40	1.79	219.49	228.45	1.69	1.68	5.84	5.83	1.87
13	5	50	25	20	130.53	130.53	130.53	132.78	0.00	0.00	1.72	1.72	3.16	131.34	136.58	0.62	0.62	4.63	4.63	2.98
14	5	50	50	20	96.26	96.26	99.02	99.24	2.87	2.87	3.10	3.10	3.09	99.73	102.13	3.60	3.60	6.10	6.10	2.83
15	5	50	25	40	12.89	12.89	13.38	13.48	3.80	3.80	4.58	4.58	3.53	13.54	13.69	5.04	5.04	6.21	6.21	3.37
16	5	50	50	40	30.24	30.24	31.12	31.23	2.91	2.91	3.27	3.27	2.96	31.53	31.82	4.27	4.27	5.22	5.22	2.81
17	5	100	25	5	360.35	360.35	360.35	367.39	0.00	0.00	1.96	1.95	3.01	370.39	377.71	2.79	2.79	4.82	4.82	2.56
18	5	100	50	5	304.23	304.23	307.43	307.86	1.05	1.05	1.19	1.19	2.79	312.75	318.65	2.80	2.80	4.74	4.74	2.50
19	5	100	25	10	294.21	294.21	294.21	296.13	0.00	0.00	0.65	0.65	4.26	300.85	305.42	2.26	2.26	3.81	3.81	4.06
20	5	100	50	10	229.41	229.41	232.29	232.59	1.26	1.26	1.39	1.39	3.85	238.54	241.90	3.98	3.98	5.44	5.44	3.68
21	5	100	25	20	134.95	134.95	134.95	136.14	0.00	0.00	0.88	0.88	7.43	141.52	143.02	4.87	4.87	5.98	5.98	6.40
22	5	100	50	20	144.41	144.41	144.41	145.32	0.00	0.00	0.63	0.63	3.43	146.39	151.08	1.37	1.37	4.62	4.62	3.73
23	5	100	25	40	58.95	58.95	58.95	60.32	0.00	0.00	2.32	2.32	8.85	61.54	62.12	4.39	4.39	5.38	5.38	9.37
24	5	100	50	40	39.44	39.44	40.98	41.11	3.90	3.90	4.23	4.23	7.09	40.83	41.85	3.52	3.52	6.11	6.11	6.80

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
25	10	50	25	5	242.85	242.85	250.12	251.02	2.99	2.99	3.36	3.36	3.74	247.49	252.97	1.91	1.91	4.17	4.17	3.60
26	10	50	50	5	282.12	282.12	289.73	290.07	2.70	2.70	2.82	2.82	3.20	289.30	290.39	2.55	2.55	2.93	2.93	3.40
27	10	50	25	10	244.54	244.54	244.54	249.18	0.00	0.00	1.90	1.90	4.85	259.17	262.12	5.98	5.98	7.19	7.19	4.66
28	10	50	50	10	288.33	283.33	297.65	298.63	5.05	3.23	5.40	3.57	3.88	300.24	307.20	5.97	4.13	8.42	6.54	3.82
29	10	50	25	20	108.98	108.98	111.67	112.57	2.47	2.47	3.29	3.29	4.84	112.73	112.87	3.44	3.44	3.57	3.57	4.83
30	10	50	50	20	157.48	157.48	159.20	160.19	1.09	1.09	1.72	1.72	4.67	160.94	163.09	2.20	2.20	3.56	3.56	4.50
31	10	50	25	40	32.25	32.25	33.48	33.54	3.81	3.81	4.00	4.00	5.25	32.85	33.91	1.86	1.86	5.15	5.15	4.97
32	10	50	50	40	36.66	36.66	36.89	37.91	0.63	0.63	3.40	3.40	5.12	38.31	38.52	4.50	4.50	5.07	5.07	4.74
33	10	100	25	5	403.18	403.18	403.18	404.98	0.00	0.00	0.45	0.45	2.84	403.18	414.55	0.00	0.00	2.82	2.82	2.09
34	10	100	50	5	296.53	296.53	305.87	306.57	3.15	3.15	3.39	3.39	2.20	308.17	309.90	3.93	3.93	4.51	4.51	2.69
35	10	100	25	10	388.87	388.87	388.87	389.41	0.00	0.00	0.14	0.14	8.84	388.87	403.11	0.00	0.00	3.66	3.66	8.64
36	10	100	50	10	294.80	294.80	301.17	302.26	2.16	2.16	2.53	2.53	9.59	309.18	312.85	4.88	4.88	6.12	6.12	9.08
37	10	100	25	20	178.28	178.28	178.28	183.12	0.00	0.00	2.71	2.71	9.81	189.30	192.94	6.18	6.18	8.22	8.22	10.18
38	10	100	50	20	175.96	175.96	182.78	183.53	3.88	3.88	4.30	4.30	10.71	182.28	185.60	3.59	3.59	5.48	5.48	11.59
39	10	100	25	40	57.50	57.50	57.50	57.74	0.00	0.00	0.42	0.42	10.68	59.42	61.25	3.34	3.34	6.52	6.52	9.94
40	10	100	50	40	31.89	31.89	32.59	33.03	2.20	2.20	3.57	3.57	10.37	33.12	33.82	3.86	3.86	6.05	6.05	11.06
41	10	200	25	5	735.27	735.27	738.42	749.47	0.43	0.43	1.93	1.93	10.91	735.27	753.39	0.00	0.00	2.46	2.46	11.74
42	10	200	50	5	512.16	506.06	509.03	519.08	0.59	-0.61	2.57	1.35	8.74	525.12	530.05	3.77	2.53	4.74	3.49	9.62
43	10	200	25	10	513.00	513.00	513.00	519.27	0.00	0.00	1.22	1.22	34.13	522.14	534.11	1.78	1.78	4.12	4.12	30.83
44	10	200	50	10	475.21		479.54	481.36		0.91		1.29	13.72	490.35	493.06		3.19		3.76	14.75
45	10	200	25	20	347.29		359.07	364.08		3.39		4.83	30.14	360.11	364.93		3.69		5.08	26.75
46	10	200	50	20	217.46		222.98	225.62		2.54		3.75	26.71	219.41	230.12		0.90		5.82	25.07
47	10	200	25	40	102.93		102.93	106.49		0.00		3.46	40.57	105.39	107.19		2.39		4.14	38.71
48	10	200	50	40	55.05		57.66	58.01		4.74		5.38	16.75	56.83	59.91		3.23		8.83	16.30
49	20	100	25	5	520.24		534.12	536.01		2.67		3.03	11.38	534.19	542.10		2.68		4.20	11.05
50	20	100	50	5	420.64		418.73	429.16		-0.45		2.03	6.18	434.17	438.95		3.22		4.35	6.21

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
51	20	100	25	10	422.21		422.21	432.98		0.00		2.55	10.46	434.26	437.74		2.85		3.68	10.68
52	20	100	50	10	360.86		368.02	368.87		1.98		2.22	8.53	368.93	382.19		2.24		5.91	9.22
53	20	100	25	20	245.17		245.17	249.12		0.00		1.61	11.58	251.19	260.17		2.46		6.12	10.90
54	20	100	50	20	185.06		190.32	191.07		2.84		3.25	8.22	192.74	194.90		4.15		5.32	9.27
55	20	100	25	40	52.52		53.73	53.85		2.30		2.53	12.73	53.90	55.83		2.63		6.30	12.36
56	20	100	50	40	19.05		20.01	20.13		5.04		5.67	29.11	20.49	21.04		7.56		10.45	28.83
57	20	200	25	5	903.84		926.65	929.52		2.52		2.84	13.33	925.19	935.37		2.36		3.49	13.56
58	20	200	50	5	485.65		481.11	495.12		-0.93		1.95	32.22	481.93	503.82		-0.77		3.74	31.63
59	20	200	25	10	616.93		629.95	633.93		2.11		2.76	32.81	630.25	639.17		2.16		3.60	32.41
60	20	200	50	10	462.31		469.21	470.47		1.49		1.77	22.67	475.19	489.81		2.79		5.95	21.82
61	20	200	25	20	373.21		380.97	385.05		2.08		3.17	59.99	386.31	389.16		3.51		4.27	55.96
62	20	200	50	20	250.75		250.75	258.23		0.00		2.98	30.98	259.45	271.26		3.47		8.18	29.80
63	20	200	25	40	93.01		96.25	96.30		3.48		3.54	80.13	96.90	97.75		4.18		5.10	77.25
64	20	200	50	40	45.40		45.40	48.09		0.00		5.93	40.67	47.39	48.26		4.38		6.30	38.71
65	20	400	25	5	1323.35		1349.87	1354.94		2.00		2.39	291.37	1349.45	1368.19		1.97		3.39	283.28
66	20	400	50	5	733.54		730.94	747.12		-0.35		1.85	43.67	751.38	768.13		2.43		4.72	43.51
67	20	400	25	10	975.12		988.11	989.59		1.33		1.48	203.85	999.32	1017.82		2.48		4.38	193.90
68	20	400	50	10	614.67		627.12	628.78		2.03		2.29	86.97	637.45	639.35		3.71		4.02	85.22
69	20	400	25	20	763.76		763.76	789.13		0.00		3.32	181.83	788.42	795.04		3.23		4.10	172.73
70	20	400	50	20	298.47		309.12	311.29		3.57		4.30	110.01	309.38	311.71		3.66		4.44	105.00
71	20	400	25	40	239.58		239.58	243.12		0.00		1.48	371.78	244.19	248.12		1.92		3.56	357.18
72	20	400	50	40	84.49		88.05	88.70		4.21		4.98	148.93	87.09	89.01		3.08		5.35	142.38
73	40	200	25	5	831.94		831.94	845.67		0.00		1.65	84.38	867.41	878.06		4.26		5.54	80.00
74	40	200	50	5	593.35		593.35	600.07		0.00		1.13	58.22	614.83	621.92		3.62		4.82	55.14
75	40	200	25	10	728.44		743.43	748.92		2.06		2.81	952.89	746.11	759.19		2.43		4.22	889.80
76	40	200	50	10	481.05		486.12	489.09		1.05		1.67	131.57	508.39	511.97		5.68		6.43	119.52

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
77	40	200	25	20	339.75		347.12	349.90		2.17		2.99	207.38	349.18	366.95		2.78		8.01	181.03
78	40	200	50	20	273.88		282.09	282.97		3.00		3.32	61.64	282.07	293.12		2.99		7.02	59.48
79	40	200	25	40	76.77		79.92	79.96		4.10		4.16	155.05	79.75	81.02		3.88		5.54	143.31
80	40	200	50	40	58.46		58.46	59.92		0.00		2.50	97.26	60.24	62.27		3.04		6.52	92.79
81	40	400	25	5	1407.05		1477.23	1483.22		4.99		5.41	402.59	1473.54	1489.41		4.73		5.85	399.68
82	40	400	50	5	858.80		858.80	883.21		0.00		2.84	767.15	889.19	898.19		3.54		4.59	737.98
83	40	400	25	10	891.02		909.32	911.55		2.05		2.30	600.42	916.52	938.73		2.86		5.35	566.48
84	40	400	50	10	757.42		775.09	779.98		2.33		2.98	480.05	789.12	792.27		4.19		4.60	449.96
85	40	400	25	20	586.29		598.12	607.12		2.02		3.55	895.72	600.91	609.81		2.49		4.01	868.21
86	40	400	50	20	395.95		407.89	408.81		3.02		3.25	340.75	420.96	422.75		6.32		6.77	308.89
87	40	400	25	40	195.33		204.67	206.32		4.78		5.63	1573.11	201.38	211.06		3.10		8.05	1456.04
88	40	400	50	40	70.77		70.77	72.12		0.00		1.91	710.02	74.03	75.41		4.61		6.56	671.06
89	40	800	25	5	2900.14		2949.54	2955.61		1.70		1.91	4007.23	3011.26	3028.11		3.83		4.41	4034.39
90	40	800	50	5	1345.70		1360.54	1371.13		1.10		1.89	1908.75	1389.55	1427.38		3.26		6.07	1697.66
91	40	800	25	10	2200.57		2200.57	2298.45		0.00		4.45	4963.79	2239.78	2342.87		1.78		6.47	4757.52
92	40	800	50	10	1025.16		1039.45	1040.17		1.39		1.46	5077.73	1062.44	1086.52		3.64		5.99	4959.48
93	40	800	25	20	1404.16		1456.19	1467.12		3.71		4.48	4377.77	1469.52	1478.39		4.65		5.29	4266.13
94	40	800	50	20	616.58		630.18	634.76		2.21		2.95	4358.62	643.83	649.12		4.42		5.28	4228.77
95	40	800	25	40	396.92		407.21	409.42		2.59		3.15	4526.01	417.56	421.56		5.20		6.21	4220.77
96	40	800	50	40	200.94		207.78	209.59		3.40		4.31	4076.39	211.53	214.35		5.27		6.67	3762.38
97	80	400	25	5	1546.23		1584.78	1589.76		2.49		2.82	1262.13	1616.90	1618.19		4.57		4.65	1229.19
98	80	400	50	5	1048.56		1048.56	1065.12		0.00		1.58	711.48	1089.45	1103.92		3.90		5.28	672.84
99	80	400	25	10	1216.74		1276.26	1282.14		4.89		5.37	2513.03	1262.09	1280.54		3.73		5.24	2275.78
100	80	400	50	10	760.61		760.61	773.45		0.00		1.69	746.12	796.41	799.35		4.71		5.09	704.78
101	80	400	25	20	565.49		580.38	581.24		2.63		2.78	1795.11	591.45	595.01		4.59		5.22	1752.69
102	80	400	50	20	372.05		378.98	380.90		1.86		2.38	1339.74	389.14	391.86		4.59		5.32	1199.91

ID	stop	stud	cap	wd	BKS(MH)	BKS (exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	Avg time
103	80	400	25	40	131.75		136.09	137.07		3.29		4.04	1285.62	137.98	144.06		4.73		9.34	1289.44
104	80	400	50	40	95.84		95.84	100.03		0.00		4.37	3998.27	100.35	102.90		4.71		7.37	3981.10

ID	stop	stud	cap	wd	BKS (MH)	BKS (exact)	LNS-2							LNS-3						
							cost						time (s)	cost						time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time
1	5	25	25	5	141.01	141.01	141.01	144.37	0.00	0.00	2.38	2.38	1.06	141.01	145.12	0.00	0.00	2.91	2.91	1.15
2	5	25	50	5	161.62	161.62	163.07	164.28	0.90	0.90	1.65	1.65	1.52	161.62	164.12	0.00	0.00	1.55	1.55	1.66
3	5	25	25	10	182.14	182.14	182.14	185.78	0.00	0.00	2.00	2.00	1.71	182.89	183.29	0.41	0.41	0.63	0.63	1.81
4	5	25	50	10	195.80	195.80	197.09	200.49	0.66	0.66	2.40	2.40	1.55	195.80	197.12	0.00	0.00	0.67	0.67	1.73
5	5	25	25	20	111.65	111.65	112.29	114.63	0.57	0.57	2.67	2.67	1.84	114.87	115.32	2.88	2.88	3.29	3.29	2.11
6	5	25	50	20	103.18	103.18	103.18	104.21	0.00	0.00	1.00	1.00	1.93	103.45	107.54	0.26	0.26	4.23	4.23	2.04
7	5	25	25	40	7.63	7.63	8.10	8.27	6.16	6.16	8.39	8.39	1.62	7.63	7.78	0.00	0.00	1.97	1.97	2.08
8	5	25	50	40	25.64	25.64	25.64	27.09	0.00	0.00	5.66	5.66	1.47	26.18	26.45	2.11	2.11	3.16	3.16	1.51
9	5	50	25	5	286.68	286.68	286.93	288.56	0.09	0.09	0.66	0.66	1.75	286.68	289.42	0.00	0.00	0.96	0.96	1.81
10	5	50	50	5	197.20	197.20	201.37	205.73	2.11	2.11	4.33	4.33	1.71	197.20	199.54	0.00	0.00	1.19	1.19	1.66
11	5	50	25	10	193.55	193.55	193.55	194.12	0.00	0.00	0.29	0.29	1.85	193.55	195.67	0.00	0.00	1.10	1.10	1.91
12	5	50	50	10	215.86	215.85	217.14	218.45	0.60	0.59	1.20	1.20	1.84	215.86	219.42	0.00	0.00	1.65	1.65	1.93
13	5	50	25	20	130.53	130.53	130.53	132.19	0.00	0.00	1.27	1.27	3.18	131.68	133.78	0.88	0.88	2.49	2.49	3.41
14	5	50	50	20	96.26	96.26	99.77	100.22	3.65	3.65	4.11	4.11	3.14	98.79	98.93	2.63	2.63	2.77	2.77	3.74
15	5	50	25	40	12.89	12.89	13.38	13.48	3.80	3.80	4.58	4.58	3.62	13.19	13.31	2.33	2.33	3.26	3.26	3.82

ID	stop	stud	cap	wd	BKS (MH)	BKS (exact)	LNS-2							LNS-3						
							cost						time (s)	cost						time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time
16	5	50	50	40	30.24	30.24	31.44	31.75	3.97	3.97	4.99	4.99	2.97	31.02	31.48	2.58	2.58	4.10	4.10	3.40
17	5	100	25	5	360.35	360.35	371.67	379.19	3.14	3.14	5.23	5.23	3.02	367.89	369.41	2.09	2.09	2.52	2.51	3.14
18	5	100	50	5	304.23	304.23	311.41	315.04	2.36	2.36	3.55	3.55	2.82	311.67	312.28	2.45	2.45	2.65	2.65	2.93
19	5	100	25	10	294.21	294.21	301.78	305.88	2.57	2.57	3.97	3.97	4.35	305.12	307.42	3.71	3.71	4.49	4.49	4.51
20	5	100	50	10	229.41	229.41	234.09	236.87	2.04	2.04	3.25	3.25	3.88	229.41	243.78	0.00	0.00	6.26	6.26	4.15
21	5	100	25	20	134.95	134.95	138.96	140.76	2.97	2.97	4.31	4.31	7.56	139.08	139.37	3.06	3.06	3.28	3.28	7.98
22	5	100	50	20	144.41	144.41	145.39	148.87	0.68	0.68	3.09	3.09	3.52	147.87	149.41	2.40	2.40	3.46	3.46	3.62
23	5	100	25	40	58.95	58.95	58.95	60.43	0.00	0.00	2.51	2.51	9.36	58.95	59.91	0.00	0.00	1.63	1.63	9.87
24	5	100	50	40	39.44	39.44	41.20	42.07	4.46	4.46	6.67	6.67	7.17	41.12	41.17	4.26	4.26	4.39	4.39	7.69
25	10	50	25	5	242.85	242.85	242.85	251.37	0.00	0.00	3.51	3.51	4.00	242.85	247.31	0.00	0.00	1.84	1.84	4.28
26	10	50	50	5	282.12	282.12	286.30	289.76	1.48	1.48	2.71	2.71	3.40	285.39	285.73	1.16	1.16	1.28	1.28	3.49
27	10	50	25	10	244.54	244.54	252.59	257.92	3.29	3.29	5.47	5.47	5.09	250.25	251.31	2.33	2.33	2.77	2.77	5.32
28	10	50	50	10	288.33	283.33	297.67	301.13	5.06	3.24	6.28	4.44	4.03	288.75	299.45	1.91	0.15	5.69	3.86	4.16
29	10	50	25	20	108.98	108.98	112.01	114.10	2.78	2.78	4.70	4.70	5.91	110.72	111.38	1.60	1.60	2.20	2.20	5.33
30	10	50	50	20	157.48	157.48	160.90	162.35	2.17	2.17	3.09	3.09	4.79	159.27	159.91	1.14	1.14	1.54	1.54	4.99
31	10	50	25	40	32.25	32.25	33.09	33.48	2.60	2.60	3.81	3.81	5.40	32.45	33.49	0.62	0.62	3.84	3.84	5.62
32	10	50	50	40	36.66	36.66	38.84	39.52	5.95	5.95	7.80	7.80	5.18	37.82	38.10	3.16	3.16	3.93	3.93	5.45
33	10	100	25	5	403.18	403.18	403.18	408.73	0.00	0.00	1.38	1.38	2.93	404.21	408.21	0.26	0.26	1.25	1.25	3.04
34	10	100	50	5	296.53	296.53	296.53	309.12	0.00	0.00	4.25	4.25	2.47	302.69	303.91	2.08	2.08	2.49	2.49	2.51
35	10	100	25	10	388.87	388.87	394.12	396.19	1.35	1.35	1.88	1.88	8.98	388.87	394.76	0.00	0.00	1.51	1.51	9.39
36	10	100	50	10	294.80	294.80	306.05	310.24	3.82	3.82	5.24	5.24	9.51	304.98	307.45	3.45	3.45	4.29	4.29	10.17
37	10	100	25	20	178.28	178.28	179.45	182.65	0.66	0.66	2.45	2.45	9.98	187.12	187.42	4.96	4.96	5.13	5.13	10.43
38	10	100	50	20	175.96	175.96	182.16	183.34	3.52	3.52	4.19	4.19	11.12	180.32	181.19	2.48	2.48	2.97	2.97	11.42
39	10	100	25	40	57.50	57.50	60.89	61.90	5.90	5.90	7.65	7.65	11.00	59.13	59.45	2.83	2.83	3.39	3.39	11.33
40	10	100	50	40	31.89	31.89	32.65	33.29	2.38	2.38	4.39	4.39	10.59	33.12	33.27	3.86	3.86	4.33	4.33	10.98
41	10	200	25	5	735.27	735.27	745.87	751.19	1.44	1.44	2.17	2.17	11.19	735.27	740.21	0.00	0.00	0.67	0.67	13.77

ID	stop	stud	cap	wd	BKS (MH)	BKS (exact)	LNS-2							LNS-3						
							cost						time (s)	cost						time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time
42	10	200	50	5	512.16	506.06	**511.42**	525.92	1.06	-0.14	3.92	2.69	9.12	**510.73**	512.82	0.92	-0.28	1.34	0.13	9.51
43	10	200	25	10	513.00	513.00	522.12	525.84	1.78	1.78	2.50	2.50	34.94	513.00	520.67	0.00	0.00	1.50	1.50	36.22
44	10	200	50	10	475.21		485.12	489.32		2.09		2.97	14.59	487.92	488.13		2.67		2.72	15.58
45	10	200	25	20	347.29		361.73	367.83		4.16		5.91	31.06	357.02	358.91		2.80		3.35	32.52
46	10	200	50	20	217.46		217.46	222.11		0.00		2.14	27.63	223.09	226.65		2.59		4.23	28.25
47	10	200	25	40	102.93		104.69	108.39		1.71		5.30	41.88	107.21	108.91		4.16		5.81	43.85
48	10	200	50	40	55.05		57.34	58.12		4.16		5.58	16.97	56.98	57.02		3.51		3.58	18.13
49	20	100	25	5	520.24		531.92	533.76		2.25		2.60	11.66	531.09	533.12		2.09		2.48	12.23
50	20	100	50	5	420.64		420.64	434.75		0.00		3.35	6.29	**419.05**	431.49		-0.38		2.58	6.52
51	20	100	25	10	422.21		433.38	438.22		2.65		3.79	10.70	434.29	436.25		2.86		3.33	11.21
52	20	100	50	10	360.86		368.85	371.39		2.21		2.92	9.06	367.98	376.45		1.97		4.32	9.49
53	20	100	25	20	245.17		245.17	251.83		0.00		2.72	12.07	245.17	252.19		0.00		2.86	12.48
54	20	100	50	20	185.06		189.32	191.69		2.30		3.58	8.41	191.46	193.49		3.46		4.56	8.01
55	20	100	25	40	52.52		52.98	53.97		0.88		2.76	13.22	54.12	54.27		3.05		3.33	13.65
56	20	100	50	40	19.05		19.65	20.37		3.15		6.93	30.11	19.84	19.91		4.15		4.51	30.95
57	20	200	25	5	903.84		919.74	925.39		1.76		2.38	14.70	922.89	925.92		2.11		2.44	15.39
58	20	200	50	5	485.65		497.12	502.57		2.36		3.48	33.19	496.08	497.09		2.15		2.36	34.10
59	20	200	25	10	616.93		628.89	631.98		1.94		2.44	34.57	616.93	620.41		0.00		0.56	36.08
60	20	200	50	10	462.31		478.30	480.57		3.46		3.95	23.28	477.12	478.12		3.20		3.42	24.39
61	20	200	25	20	373.21		376.52	382.12		0.89		2.39	61.79	373.21	380.19		0.00		1.87	63.84
62	20	200	50	20	250.75		258.29	261.07		3.01		4.12	31.89	258.67	264.19		3.16		5.36	33.20
63	20	200	25	40	93.01		96.53	97.72		3.78		5.06	83.10	95.02	96.22		2.16		3.45	86.64
64	20	200	50	40	45.40		46.11	47.20		1.56		3.96	42.44	45.40	47.12		0.00		3.79	44.60
65	20	400	25	5	1323.35		1364.90	1384.49		3.14		4.62	298.62	1341.07	1353.29		1.34		2.26	324.71
66	20	400	50	5	733.54		**731.67**	755.90		-0.25		3.05	45.59	752.98	759.28		2.65		3.51	47.86

ID	stop	stud	cap	wd	BKS (MH)	BKS (exact)	LNS-2							LNS-3						
							cost						time (s)	cost						time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time
67	20	400	25	10	975.12		992.10	1004.17		1.74		2.98	208.98	998.30	1000.45		2.38		2.60	220.56
68	20	400	50	10	614.67		634.12	646.33		3.16		5.15	90.30	627.14	629.81		2.03		2.46	95.37
69	20	400	25	20	763.76		789.02	805.35		3.31		5.45	187.45	763.76	775.56		0.00		1.54	197.13
70	20	400	50	20	298.47		309.45	313.11		3.68		4.91	112.59	310.49	315.43		4.03		5.68	118.89
71	20	400	25	40	239.58		244.12	246.02		1.89		2.69	378.38	243.39	244.98		1.59		2.25	399.32
72	20	400	50	40	84.49		87.90	88.73		4.04		5.02	152.40	86.29	87.45		2.13		3.50	161.65
73	40	200	25	5	831.94		848.29	853.39		1.97		2.58	86.99	858.12	860.39		3.15		3.42	91.32
74	40	200	50	5	593.35		608.73	619.25		2.59		4.37	60.74	607.42	608.25		2.37		2.51	63.50
75	40	200	25	10	728.44		734.19	737.68		0.79		1.27	981.90	748.37	750.53		2.74		3.03	1028.44
76	40	200	50	10	481.05		500.98	508.12		4.14		5.63	137.54	501.94	502.97		4.34		4.56	142.86
77	40	200	25	20	339.75		339.75	346.28		0.00		1.92	215.40	351.29	357.49		3.40		5.22	225.19
78	40	200	50	20	273.88		275.42	281.74		0.56		2.87	63.14	273.88	286.49		0.00		4.60	66.52
79	40	200	25	40	76.77		79.90	80.12		4.08		4.36	158.58	77.65	80.02		1.15		4.23	166.92
80	40	200	50	40	58.46		60.21	60.37		2.99		3.27	99.92	60.12	61.01		2.84		4.36	104.61
81	40	400	25	5	1407.05		1462.19	1481.19		3.92		5.27	423.40	1448.65	1458.38		2.96		3.65	442.93
82	40	400	50	5	858.80		879.32	890.21		2.39		3.66	808.04	858.80	880.27		0.00		2.50	845.43
83	40	400	25	10	891.02		899.12	901.93		0.91		1.22	617.99	913.39	919.87		2.51		3.24	643.77
84	40	400	50	10	757.42		784.30	791.26		3.55		4.47	492.75	774.39	776.65		2.24		2.54	515.28
85	40	400	25	20	586.29		594.19	600.76		1.35		2.47	951.78	586.29	599.21		0.00		2.20	992.06
86	40	400	50	20	395.95		402.17	408.12		1.57		3.07	351.32	412.65	413.97		4.22		4.55	365.26
87	40	400	25	40	195.33		199.03	201.54		1.89		3.18	1607.68	203.09	205.41		3.97		5.16	1704.52
88	40	400	50	40	70.77		74.93	75.59		5.88		6.81	723.28	73.12	73.54		3.32		3.91	764.01
89	40	800	25	5	2900.14		2976.54	2997.28		2.63		3.35	4297.35	2952.39	2953.21		1.80		1.83	4495.55
90	40	800	50	5	1345.70		1373.19	1406.78		2.04		4.54	1955.48	1385.67	1399.43		2.97		3.99	2031.99
91	40	800	25	10	2200.57		2223.30	2287.91		1.03		3.97	5060.71	2289.38	2293.12		4.04		4.21	5438.28

ID	stop	stud	cap	wd	BKS (MH)	BKS (exact)	LNS-2							LNS-3						
							cost						time (s)	cost						time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap (MH)	%Avg Gap (exact)	% Avg Gap (MH)	avg time
92	40	800	50	10	1025.16		1048.31	1074.17		2.26		4.78	5239.34	1047.39	1058.15		2.17		3.22	5518.28
93	40	800	25	20	1404.16		1438.90	1459.95		2.47		3.97	4455.99	1430.19	1476.21		1.85		5.13	4677.14
94	40	800	50	20	616.58		635.29	640.12		3.03		3.82	4394.49	631.98	636.29		2.50		3.20	4612.66
95	40	800	25	40	396.92		411.94	415.39		3.78		4.65	4555.08	409.38	412.90		3.14		4.03	4958.06
96	40	800	50	40	200.94		208.65	210.28		3.84		4.65	4190.83	209.89	210.45		4.45		4.73	4484.15
97	80	400	25	5	1546.23		1576.19	1600.39		1.94		3.50	1328.56	1579.45	1583.45		2.15		2.41	1303.99
98	80	400	50	5	1048.56		1083.21	1085.34		3.30		3.51	744.23	1068.45	1079.41		1.90		2.94	717.44
99	80	400	25	10	1216.74		1258.43	1278.67		3.43		5.09	2584.15	1242.89	1267.56		2.15		4.18	2559.52
100	80	400	50	10	760.61		785.12	785.35		3.22		3.25	768.05	760.61	773.39		0.00		1.68	769.32
101	80	400	25	20	565.49		584.38	587.45		3.34		3.88	1890.40	578.43	582.12		2.29		2.94	1944.50
102	80	400	50	20	372.05		385.74	386.22		3.68		3.81	1381.76	380.12	383.12		2.17		2.98	1375.31
103	80	400	25	40	131.75		137.39	139.61		4.28		5.97	1348.11	134.68	136.98		2.22		3.97	1660.14
104	80	400	50	40	95.84		98.56	99.78		2.84		4.11	4254.52	100.90	101.84		5.28		6.26	4721.85

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
1	5	25	25	5	141.01	141.01	141.01	143.19	0.00	0.00	1.55	1.55	1.18	141.01	144.06	0.00	0.00	2.16	2.16	1.04
2	5	25	50	5	161.62	161.62	161.67	161.78	0.03	0.03	0.10	0.10	1.72	161.62	163.02	0.00	0.00	0.87	0.87	1.57
3	5	25	25	10	182.14	182.14	184.02	184.49	1.03	1.03	1.29	1.29	1.92	187.86	188.48	3.14	3.14	3.48	3.48	1.79

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
4	5	25	50	10	195.80	195.80	195.80	195.87	0.00	0.00	0.04	0.04	1.83	201.17	201.45	2.74	2.74	2.89	2.89	1.63
5	5	25	25	20	111.65	111.65	114.12	115.09	2.21	2.21	3.08	3.08	2.16	111.78	112.06	0.12	0.12	0.37	0.37	1.90
6	5	25	50	20	103.18	103.18	103.18	104.12	0.00	0.00	0.91	0.91	2.10	103.78	104.27	0.58	0.58	1.06	1.06	1.92
7	5	25	25	40	7.63	7.63	7.96	7.98	4.33	4.33	4.59	4.59	1.72	7.73	7.91	1.31	1.31	3.67	3.67	1.87
8	5	25	50	40	25.64	25.64	26.65	26.92	3.94	3.94	4.99	4.99	1.49	26.32	27.92	2.65	2.65	8.89	8.89	1.47
9	5	50	25	5	286.68	286.68	288.02	288.93	0.47	0.47	0.78	0.78	1.80	286.97	293.84	0.10	0.10	2.50	2.50	1.72
10	5	50	50	5	197.20	197.20	199.91	207.87	1.37	1.37	5.41	5.41	1.67	205.67	207.73	4.30	4.30	5.34	5.34	1.58
11	5	50	25	10	193.55	193.55	193.55	194.12	0.00	0.00	0.29	0.29	1.92	193.55	196.65	0.00	0.00	1.60	1.60	1.83
12	5	50	50	10	215.86	215.85	215.86	219.87	0.00	0.00	1.86	1.86	1.90	218.78	223.12	1.36	1.35	3.37	3.36	1.90
13	5	50	25	20	130.53	130.53	131.02	134.09	0.38	0.38	2.73	2.73	3.36	130.53	135.12	0.00	0.00	3.52	3.52	3.28
14	5	50	50	20	96.26	96.26	99.02	100.26	2.87	2.87	4.16	4.16	2.87	99.72	102.19	3.59	3.59	6.16	6.16	3.55
15	5	50	25	40	12.89	12.89	13.45	13.58	4.34	4.34	5.35	5.35	3.76	13.29	13.47	3.10	3.10	4.50	4.50	3.69
16	5	50	50	40	30.24	30.24	31.29	31.39	3.47	3.47	3.80	3.80	3.35	30.24	30.67	0.00	0.00	1.42	1.42	3.11
17	5	100	25	5	360.35	360.35	360.35	362.18	0.00	0.00	0.51	0.51	3.27	370.72	376.19	2.88	2.88	4.40	4.40	3.07
18	5	100	50	5	304.23	304.23	310.94	311.81	2.21	2.21	2.49	2.49	3.04	310.94	313.19	2.21	2.21	2.95	2.95	2.67
19	5	100	25	10	294.21	294.21	294.21	297.84	0.00	0.00	1.23	1.23	4.65	297.19	301.09	1.01	1.01	2.34	2.34	4.08
20	5	100	50	10	229.41	229.41	232.39	234.98	1.30	1.30	2.43	2.43	4.29	234.89	238.12	2.39	2.39	3.80	3.80	3.53
21	5	100	25	20	134.95	134.95	134.95	139.92	0.00	0.00	3.68	3.68	8.77	139.97	140.98	3.72	3.72	4.47	4.47	7.01
22	5	100	50	20	144.41	144.41	147.81	149.21	2.35	2.35	3.32	3.32	3.83	147.67	149.73	2.26	2.26	3.68	3.68	3.09
23	5	100	25	40	58.95	58.95	60.02	60.84	1.82	1.82	3.21	3.21	9.94	61.08	61.42	3.61	3.61	4.19	4.19	8.54
24	5	100	50	40	39.44	39.44	41.83	41.93	6.06	6.06	6.31	6.31	7.84	41.83	42.63	6.06	6.06	8.09	8.09	6.71
25	10	50	25	5	242.85	242.85	242.85	246.53	0.00	0.00	1.52	1.52	4.95	246.90	247.67	1.67	1.67	1.98	1.98	3.71
26	10	50	50	5	282.12	282.12	282.45	283.11	0.12	0.12	0.35	0.35	3.86	282.12	283.12	0.00	0.00	0.35	0.35	3.11
27	10	50	25	10	244.54	244.54	251.05	253.90	2.66	2.66	3.83	3.83	5.02	250.39	255.13	2.39	2.39	4.33	4.33	4.55
28	10	50	50	10	288.33	283.33	289.67	293.05	2.24	0.46	3.43	1.64	4.43	294.86	295.39	4.07	2.26	4.26	2.45	3.51
29	10	50	25	20	108.98	108.98	112.98	113.28	3.67	3.67	3.95	3.95	5.40	113.46	114.87	4.11	4.11	5.40	5.40	4.70

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4 cost Best sol	LNS-4 cost Avg sol	LNS-4 cost %Best Gap (exct)	LNS-4 cost %Best Gap(MH)	LNS-4 cost %Avg Gap (exact)	LNS-4 cost % Avg Gap(MH)	LNS-4 Time(s) avg time	LNS-5 cost Best sol	LNS-5 cost Avg sol	LNS-5 cost %Best Gap (exact)	LNS-5 cost %Best Gap(MH)	LNS-5 cost %Avg Gap (exact)	LNS-5 cost % Avg Gap(MH)	LNS-5 Time (s) Avg time
30	10	50	50	20	157.48	157.48	161.19	161.19	2.36	2.36	2.36	2.36	6.10	160.82	162.89	2.12	2.12	3.44	3.44	4.34
31	10	50	25	40	32.25	32.25	33.29	33.62	3.22	3.22	4.25	4.25	6.73	33.87	34.05	5.02	5.02	5.58	5.58	4.86
32	10	50	50	40	36.66	36.66	37.46	37.90	2.18	2.18	3.38	3.38	6.11	37.83	39.07	3.19	3.19	6.57	6.57	4.81
33	10	100	25	5	403.18	403.18	403.97	405.76	0.20	0.20	0.64	0.64	3.32	405.79	407.19	0.65	0.65	0.99	0.99	2.64
34	10	100	50	5	296.53	296.53	309.21	310.56	4.28	4.28	4.73	4.73	2.95	310.54	312.21	4.72	4.72	5.29	5.29	2.30
35	10	100	25	10	388.87	388.87	388.87	393.76	0.00	0.00	1.26	1.26	11.82	388.87	394.42	0.00	0.00	1.43	1.43	8.34
36	10	100	50	10	294.80	294.80	307.21	307.38	4.21	4.21	4.27	4.27	11.23	305.39	309.98	3.59	3.59	5.15	5.15	8.76
37	10	100	25	20	178.28	178.28	178.28	179.92	0.00	0.00	0.92	0.92	10.51	178.28	180.09	0.00	0.00	1.02	1.02	9.08
38	10	100	50	20	175.96	175.96	181.93	182.90	3.39	3.39	3.94	3.94	12.03	184.93	186.84	5.10	5.10	6.18	6.18	10.00
39	10	100	25	40	57.50	57.50	58.09	58.96	1.03	1.03	2.54	2.54	11.80	58.89	59.76	2.42	2.42	3.93	3.93	9.99
40	10	100	50	40	31.89	31.89	32.08	32.45	0.60	0.60	1.76	1.76	12.01	32.58	32.84	2.16	2.16	2.98	2.98	9.64
41	10	200	25	5	735.27	735.27	743.39	745.12	1.10	1.10	1.34	1.34	13.97	754.46	758.54	2.61	2.61	3.16	3.16	11.86
42	10	200	50	5	512.16	506.06	**511.08**	525.90	0.99	-0.21	3.92	2.68	10.63	**511.90**	532.19	1.15	-0.05	5.16	3.91	8.40
43	10	200	25	10	513.00	513.00	522.12	524.12	1.78	1.78	2.17	2.17	36.85	524.39	529.02	2.22	2.22	3.12	3.12	31.64
44	10	200	50	10	475.21		475.21	486.92		0.00		2.46	16.00	487.62	488.59		2.61		2.82	13.46
45	10	200	25	20	347.29		361.19	362.59		4.00		4.41	32.08	361.78	368.93		4.17		6.23	29.06
46	10	200	50	20	217.46		217.46	219.87		0.00		1.11	27.99	219.85	226.45		1.10		4.13	26.58
47	10	200	25	40	102.93		106.29	107.45		3.26		4.39	43.97	103.93	104.90		0.97		1.91	41.07
48	10	200	50	40	55.05		57.02	57.83		3.58		5.05	17.00	56.84	57.66		3.25		4.74	18.14
49	20	100	25	5	520.24		530.98	532.92		2.06		2.44	12.44	526.89	528.19		1.28		1.53	12.37
50	20	100	50	5	420.64		432.39	433.12		2.79		2.97	7.00	431.09	432.05		2.48		2.71	6.73
51	20	100	25	10	422.21		433.89	436.98		2.77		3.50	11.37	437.54	440.35		3.63		4.30	11.52
52	20	100	50	10	360.86		364.89	368.90		1.12		2.23	12.34	364.92	364.83		1.13		1.10	9.52
53	20	100	25	20	245.17		245.17	252.92		0.00		3.16	12.73	245.67	255.19		0.20		4.09	12.53
54	20	100	50	20	185.06		186.80	186.99		0.94		1.04	8.92	185.06	190.95		0.00		3.18	8.62
55	20	100	25	40	52.52		53.93	54.43		2.68		3.64	13.84	54.39	54.62		3.56		4.00	14.01

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
56	20	100	50	40	19.05		19.49	19.65		2.31		3.15	30.13	19.62	19.91		2.99		4.51	30.99
57	20	200	25	5	903.84		922.17	923.98		2.03		2.23	14.29	931.28	937.95		3.04		3.77	15.17
58	20	200	50	5	485.65		**481.42**	503.76		-0.87		3.73	34.50	501.90	508.45		3.35		4.69	34.05
59	20	200	25	10	616.93		631.49	633.41		2.36		2.67	38.18	636.18	638.19		3.12		3.45	35.23
60	20	200	50	10	462.31		474.29	478.39		2.59		3.48	24.63	478.89	481.12		3.59		4.07	24.37
61	20	200	25	20	373.21		373.21	379.67		0.00		1.73	65.25	379.56	385.05		1.70		3.17	62.50
62	20	200	50	20	250.75		250.75	252.98		0.00		0.89	33.89	262.89	264.72		4.84		5.57	32.01
63	20	200	25	40	93.01		94.93	95.48		2.06		2.66	89.66	97.78	98.16		5.13		5.54	81.26
64	20	200	50	40	45.40		46.41	46.98		2.22		3.48	44.48	46.03	47.82		1.39		5.33	41.39
65	20	400	25	5	1323.35		1369.47	1386.90		3.49		4.80	326.67	1371.19	1383.56		3.62		4.55	307.76
66	20	400	50	5	733.54		748.12	756.20		1.99		3.09	48.66	753.12	756.42		2.67		3.12	45.06
67	20	400	25	10	975.12		1008.31	1010.54		3.40		3.63	219.47	975.12	989.59		0.00		1.48	212.67
68	20	400	50	10	614.67		639.87	645.97		4.10		5.09	90.08	642.91	644.12		4.59		4.79	89.79
69	20	400	25	20	763.76		792.32	803.92		3.74		5.26	196.87	799.36	806.85		4.66		5.64	188.48
70	20	400	50	20	298.47		312.49	312.87		4.70		4.82	119.56	313.29	315.84		4.97		5.82	113.95
71	20	400	25	40	239.58		239.58	245.39		0.00		2.43	382.09	244.97	247.13		2.25		3.15	385.83
72	20	400	50	40	84.49		87.39	87.79		3.43		3.91	176.00	87.65	88.67		3.74		4.95	153.76
73	40	200	25	5	831.94		848.92	853.12		2.04		2.55	90.73	872.97	877.32		4.93		5.45	86.53
74	40	200	50	5	593.35		613.95	620.76		3.47		4.62	60.12	618.42	624.19		4.23		5.20	60.50
75	40	200	25	10	728.44		753.62	755.43		3.46		3.71	1039.42	728.44	741.08		0.00		1.74	990.75
76	40	200	50	10	481.05		506.42	506.98		5.27		5.39	152.99	506.98	507.48		5.39		5.49	138.67
77	40	200	25	20	339.75		345.19	349.76		1.60		2.95	228.99	346.95	352.36		2.12		3.71	220.56
78	40	200	50	20	273.88		283.12	283.39		3.37		3.47	67.29	279.91	281.13		2.20		2.65	64.12
79	40	200	25	40	76.77		78.83	78.92		2.68		2.80	170.18	78.48	79.53		2.23		3.60	160.89
80	40	200	50	40	58.46		59.65	59.92		2.04		2.50	109.62	59.92	60.83		2.50		4.05	97.16

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time (s)
							Best sol	Avg sol	%Best Gap (exct)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
81	40	400	25	5	1407.05		1464.12	1475.54		4.06		4.87	449.47	1479.89	1486.19		5.18		5.62	430.58
82	40	400	50	5	858.80		887.12	892.21		3.30		3.89	854.59	889.37	901.29		3.56		4.95	821.90
83	40	400	25	10	891.02		891.02	901.38		0.00		1.16	652.54	911.09	924.37		2.25		3.74	634.11
84	40	400	50	10	757.42		786.31	789.31		3.81		4.21	519.68	797.34	803.15		5.27		6.04	499.27
85	40	400	25	20	586.29		598.29	603.37		2.05		2.91	997.36	608.25	612.28		3.75		4.43	961.68
86	40	400	50	20	395.95		402.39	405.83		1.63		2.50	369.18	408.34	410.39		3.13		3.65	356.31
87	40	400	25	40	195.33		199.65	203.45		2.21		4.16	1718.67	202.93	205.31		3.89		5.11	1648.10
88	40	400	50	40	70.77		73.52	74.21		3.89		4.86	764.54	74.41	74.95		5.14		5.91	738.03
89	40	800	25	5	2900.14		2999.43	3033.19		3.42		4.59	4525.32	2965.51	2976.37		2.25		2.63	4306.01
90	40	800	50	5	1345.70		1345.64	1370.12		0.00		1.81	2063.58	1376.15	1397.14		2.26		3.82	1998.46
91	40	800	25	10	2200.57		2258.39	2266.59		2.63		3.00	5499.28	2286.54	2309.55		3.91		4.95	5096.89
92	40	800	50	10	1025.16		1057.31	1068.90		3.14		4.27	5591.41	1025.16	1060.39		0.00		3.44	5271.21
93	40	800	25	20	1404.16		1444.89	1445.90		2.90		2.97	4805.56	1438.19	1457.19		2.42		3.78	4526.49
94	40	800	50	20	616.58		631.29	636.67		2.39		3.26	4741.33	636.93	637.71		3.30		3.43	4504.28
95	40	800	25	40	396.92		403.12	408.39		1.56		2.89	5016.97	410.72	417.12		3.48		5.09	4801.19
96	40	800	50	40	200.94		204.15	205.81		1.60		2.42	4500.06	204.67	210.27		1.86		4.64	4214.64
97	80	400	25	5	1546.23		1546.23	1579.19		0.00		2.13	1317.34	1586.39	1588.32		2.60		2.72	1271.69
98	80	400	50	5	1048.56		1089.64	1091.31		3.92		4.08	740.30	1079.58	1088.39		2.96		3.80	687.93
99	80	400	25	10	1216.74		1268.42	1286.65		4.25		5.75	2619.65	1289.75	1295.53		6.00		6.48	2500.59
100	80	400	50	10	760.61		779.39	782.41		2.47		2.87	851.40	779.18	785.37		2.44		3.26	739.42
101	80	400	25	20	565.49		581.74	585.92		2.87		3.61	1943.57	591.05	591.84		4.52		4.66	1883.37
102	80	400	50	20	372.05		380.29	384.98		2.21		3.48	1621.76	381.12	384.30		2.44		3.29	1351.14
103	80	400	25	40	131.75		135.29	137.98		2.69		4.73	1704.32	136.90	139.34		3.91		5.76	1618.90

							LNS-4							LNS-5						
							cost						Time(s)	cost						Time (s)
ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	Best sol	Avg sol	%Best Gap (exct)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
104	80	400	50	40	95.84		98.12	100.02		2.38		4.36	4897.16	99.83	101.90		4.16		6.32	4520.50

Appendix 4

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
1	5	25	25	5	141.01	141.01	141.01	143.00	0.00	0.00	1.41	1.41	1.10	**141.01**	146.00	0.00	0.00	3.54	3.54	1.05
2	5	25	50	5	161.62	161.62	163.47	165.36	1.14	1.14	2.31	2.31	1.54	162.69	163.23	0.66	0.66	1.00	1.00	1.37
3	5	25	25	10	182.14	182.14	183.03	186.39	0.49	0.49	2.33	2.33	1.88	182.14	186.62	0.00	0.00	2.46	2.46	1.80
4	5	25	50	10	195.80	195.80	195.80	196.64	0.00	0.00	0.43	0.43	2.05	198.19	199.13	1.22	1.22	1.70	1.70	1.72
5	5	25	25	20	111.65	111.65	112.21	114.89	0.50	0.50	2.90	2.90	1.87	113.97	114.87	2.08	2.08	2.88	2.88	1.72
6	5	25	50	20	103.18	103.18	103.18	105.90	0.00	0.00	2.64	2.64	2.11	103.18	104.75	0.00	0.00	1.52	1.52	2.01
7	5	25	25	40	7.63	7.63	7.89	8.10	3.41	3.41	6.18	6.18	1.86	7.85	7.88	2.89	2.89	3.27	3.27	1.47
8	5	25	50	40	25.64	25.64	26.84	28.20	4.67	4.67	9.97	9.97	1.69	27.78	27.88	8.34	8.34	8.72	8.72	1.77
9	5	50	25	5	286.68	286.68	286.68	290.37	0.00	0.00	1.29	1.29	1.81	286.68	299.16	0.00	0.00	4.35	4.35	1.67
10	5	50	50	5	197.20	197.20	201.14	203.04	2.00	2.00	2.96	2.96	1.70	205.81	208.16	4.37	4.37	5.56	5.56	1.52
11	5	50	25	10	193.55	193.55	194.11	196.40	0.29	0.29	1.47	1.47	1.90	195.07	201.16	0.79	0.79	3.93	3.93	1.72
12	5	50	50	10	215.86	215.85	218.64	220.12	1.29	1.29	1.98	1.97	1.95	224.50	229.75	4.01	4.00	6.44	6.43	1.90
13	5	50	25	20	130.53	130.53	131.12	133.76	0.45	0.45	2.47	2.47	3.33	132.17	135.58	1.26	1.26	3.87	3.87	3.03
14	5	50	50	20	96.26	96.26	99.11	101.13	2.96	2.96	5.06	5.06	3.16	102.84	104.79	6.83	6.83	8.86	8.86	2.88
15	5	50	25	40	12.89	12.89	13.44	14.00	4.27	4.27	8.62	8.62	3.71	13.59	13.90	5.43	5.43	7.84	7.84	3.43
16	5	50	50	40	30.24	30.24	31.21	31.82	3.22	3.22	5.22	5.22	3.15	30.82	31.48	1.92	1.92	4.10	4.10	2.86
17	5	100	25	5	360.35	360.35	360.35	362.30	0.00	0.00	0.54	0.54	2.95	369.89	374.99	2.65	2.65	4.06	4.06	2.61
18	5	100	50	5	304.23	304.23	308.41	310.34	1.37	1.37	2.01	2.01	2.92	309.76	312.80	1.82	1.82	2.82	2.82	2.54
19	5	100	25	10	294.21	294.21	295.29	297.16	0.37	0.37	1.00	1.00	4.42	299.25	299.65	1.71	1.71	1.85	1.85	4.13
20	5	100	50	10	229.41	229.41	234.29	237.95	2.13	2.13	3.72	3.72	3.93	240.64	242.90	4.90	4.90	5.88	5.88	3.74
21	5	100	25	20	134.95	134.95	134.95	136.27	0.00	0.00	0.98	0.98	7.84	143.54	144.71	6.37	6.37	7.23	7.23	6.50
22	5	100	50	20	144.41	144.41	145.09	145.75	0.47	0.47	0.93	0.93	3.51	148.74	152.24	3.00	3.00	5.42	5.42	3.79
23	5	100	25	40	58.95	58.95	59.03	60.23	0.14	0.14	2.17	2.17	9.06	58.95	62.87	0.00	0.00	6.64	6.64	9.52
24	5	100	50	40	39.44	39.44	42.01	42.75	6.52	6.52	8.40	8.40	7.31	40.96	42.10	3.85	3.85	6.74	6.74	6.90
25	10	50	25	5	242.85	242.85	249.31	250.09	2.66	2.66	2.98	2.98	3.88	247.75	252.06	2.02	2.02	3.79	3.79	3.66

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
26	10	50	50	5	282.12	282.12	288.95	289.70	2.42	2.42	2.69	2.69	3.26	290.52	292.55	2.98	2.98	3.70	3.70	3.46
27	10	50	25	10	244.54	244.54	244.54	249.84	0.00	0.00	2.17	2.17	5.03	259.24	262.42	6.01	6.01	7.31	7.31	4.73
28	10	50	50	10	288.33	283.33	296.20	296.58	4.54	2.73	4.67	2.86	4.04	299.94	305.88	5.86	4.03	7.96	6.09	3.88
29	10	50	25	20	108.98	108.98	110.48	111.59	1.38	1.38	2.39	2.39	5.06	111.93	113.14	2.71	2.71	3.81	3.81	4.91
30	10	50	50	20	157.48	157.48	158.16	160.41	0.43	0.43	1.86	1.86	4.90	160.12	161.99	1.68	1.68	2.86	2.86	4.57
31	10	50	25	40	32.25	32.25	33.85	33.98	4.96	4.96	5.36	5.36	5.46	33.43	33.81	3.67	3.67	4.84	4.84	5.05
32	10	50	50	40	36.66	36.66	37.24	37.41	1.59	1.59	2.05	2.05	5.23	37.54	38.58	2.39	2.39	5.24	5.24	4.82
33	10	100	25	5	403.18	403.18	403.18	407.82	0.00	0.00	1.15	1.15	2.41	410.32	414.96	1.77	1.77	2.92	2.92	1.98
34	10	100	50	5	296.53	296.53	307.72	308.82	3.77	3.77	4.15	4.15	2.71	302.25	310.80	1.93	1.93	4.81	4.81	2.74
35	10	100	25	10	388.87	388.87	388.87	391.41	0.00	0.00	0.65	0.65	10.05	390.93	404.01	0.53	0.53	3.89	3.89	8.78
36	10	100	50	10	294.80	294.80	302.24	304.76	2.52	2.52	3.38	3.38	10.11	306.61	311.29	4.01	4.01	5.60	5.60	9.23
37	10	100	25	20	178.28	178.28	178.28	182.01	0.00	0.00	2.09	2.09	11.30	191.00	194.84	7.13	7.13	9.29	9.29	10.34
38	10	100	50	20	175.96	175.96	183.00	184.05	4.00	4.00	4.60	4.60	12.10	182.86	185.97	3.92	3.92	5.69	5.69	11.78
39	10	100	25	40	57.50	57.50	59.90	60.06	4.17	4.17	4.45	4.45	11.43	57.97	62.27	0.81	0.81	8.30	8.30	10.10
40	10	100	50	40	31.89	31.89	33.14	33.37	3.90	3.90	4.63	4.63	11.43	34.05	34.99	6.76	6.76	9.73	9.73	11.23
41	10	200	25	5	735.27	735.27	735.27	756.23	0.00	0.00	2.85	2.85	13.00	735.27	756.41	0.00	0.00	2.87	2.87	11.93
42	10	200	50	5	512.16	506.06	**510.36**	512.08	0.85	-0.35	1.19	-0.02	10.14	523.29	532.03	3.40	2.17	5.13	3.88	9.77
43	10	200	25	10	513.00	513.00	514.16	514.34	0.23	0.23	0.26	0.26	35.64	517.22	526.04	0.82	0.82	2.54	2.54	31.33
44	10	200	50	10	475.21		479.12	480.38		0.82		1.09	14.97	493.67	495.04		3.89		4.17	14.98
45	10	200	25	20	347.29		361.08	361.25		3.97		4.02	31.28	361.11	363.80		3.98		4.75	27.18
46	10	200	50	20	217.46		225.20	225.76		3.56		3.81	27.94	219.38	226.65		0.88		4.23	25.47
47	10	200	25	40	102.93		102.93	105.16		0.00		2.17	41.31	107.46	109.40		4.40		6.29	39.33
48	10	200	50	40	55.05		57.69	58.03		4.80		5.41	17.82	56.56	59.20		2.75		7.54	16.57
49	20	100	25	5	520.24		520.24	532.12		0.00		2.28	11.58	520.24	541.01		0.00		3.99	11.22
50	20	100	50	5	420.64		428.17	430.08		1.79		2.24	6.75	435.48	441.53		3.53		4.97	6.31
51	20	100	25	10	422.21		429.12	436.18		1.64		3.31	11.22	437.00	439.83		3.50		4.17	10.85
52	20	100	50	10	360.86		369.65	370.94		2.44		2.79	8.88	371.66	375.19		2.99		3.97	9.37
53	20	100	25	20	245.17		245.17	248.12		0.00		1.20	11.96	254.59	259.06		3.84		5.67	11.07

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
54	20	100	50	20	185.06		190.89	193.05		3.15		4.32	9.05	189.61	193.82		2.46		4.73	9.42
55	20	100	25	40	52.52		54.90	55.06		4.53		4.83	13.36	53.08	54.86		1.07		4.46	12.56
56	20	100	50	40	19.05		20.10	20.43		5.51		7.24	30.42	19.91	20.38		4.52		6.99	29.29
57	20	200	25	5	903.84		903.84	933.58		0.00		3.29	13.64	914.53	932.48		1.18		3.17	13.78
58	20	200	50	5	485.65		486.65	489.31		0.21		0.75	33.55	489.55	512.66		0.80		5.56	32.14
59	20	200	25	10	616.93		633.06	636.55		2.62		3.18	33.71	629.53	644.16		2.04		4.41	32.93
60	20	200	50	10	462.31		465.77	471.46		0.75		1.98	23.04	471.00	479.80		1.88		3.78	22.17
61	20	200	25	20	373.21		380.07	385.97		1.84		3.42	61.65	379.19	391.39		1.60		4.87	56.85
62	20	200	50	20	250.75		250.75	257.76		0.00		2.80	33.71	262.40	274.09		4.65		9.31	30.27
63	20	200	25	40	93.01		96.97	97.94		4.26		5.30	83.06	98.01	100.00		5.37		7.52	78.49
64	20	200	50	40	45.40		45.40	47.33		0.00		4.25	41.87	47.52	49.06		4.67		8.06	39.33
65	20	400	25	5	1323.35		1354.64	1364.86		2.36		3.14	299.95	1352.96	1372.16		2.24		3.69	287.82
66	20	400	50	5	733.54		733.54	750.33		0.00		2.29	44.79	751.40	760.13		2.43		3.63	44.20
67	20	400	25	10	975.12		990.30	991.80		1.56		1.71	207.16	1002.13	1012.79		2.77		3.86	197.00
68	20	400	50	10	614.67		629.23	630.78		2.37		2.62	88.51	637.45	638.64		3.71		3.90	86.59
69	20	400	25	20	763.76		763.76	787.26		0.00		3.08	185.77	791.98	798.79		3.70		4.59	175.50
70	20	400	50	20	298.47		311.49	312.29		4.36		4.63	112.05	312.30	315.09		4.63		5.57	106.68
71	20	400	25	40	239.58		241.12	246.63		0.64		2.94	378.97	247.27	251.45		3.21		4.96	362.90
72	20	400	50	40	84.49		88.67	88.89		4.95		5.21	152.38	88.08	90.05		4.25		6.58	144.66
73	40	200	25	5	831.94		835.78	849.63		0.46		2.13	86.52	880.24	890.10		5.81		6.99	81.28
74	40	200	50	5	593.35		593.35	600.98		0.00		1.29	59.97	614.55	629.90		3.57		6.16	56.02
75	40	200	25	10	728.44		728.44	734.98		0.00		0.90	966.70	741.32	749.32		1.77		2.87	904.05
76	40	200	50	10	481.05		487.10	490.22		1.26		1.91	133.84	514.45	529.89		6.94		10.15	121.43
77	40	200	25	20	339.75		349.05	351.89		2.74		3.57	212.45	350.27	362.93		3.10		6.82	183.93
78	40	200	50	20	273.88		273.88	280.23		0.00		2.32	62.72	283.07	292.96		3.35		6.97	60.43
79	40	200	25	40	76.77		80.84	80.95		5.30		5.44	157.58	80.75	82.01		5.18		6.82	145.60
80	40	200	50	40	58.46		58.46	59.06		0.00		1.02	99.87	61.11	62.99		4.53		7.75	94.27
81	40	400	25	5	1407.05		1487.12	1490.66		5.69		5.94	411.51	1475.23	1498.18		4.85		6.48	406.08

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	ALNS							LNS-1						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
82	40	400	50	5	858.80		858.80	881.21		0.00		2.61	779.89	893.41	902.63		4.03		5.10	749.79
83	40	400	25	10	891.02		912.86	920.20		2.45		3.27	608.61	908.62	921.80		1.98		3.45	575.55
84	40	400	50	10	757.42		777.75	782.26		2.68		3.28	487.77	781.85	788.41		3.23		4.09	457.17
85	40	400	25	20	586.29		604.14	608.32		3.04		3.76	911.11	596.93	603.07		1.82		2.86	882.12
86	40	400	50	20	395.95		410.65	419.73		3.71		6.01	346.14	403.09	405.36		1.80		2.38	313.84
87	40	400	25	40	195.33		207.89	209.68		6.43		7.34	1564.80	205.21	206.96		5.06		5.96	1479.35
88	40	400	50	40	70.77		70.77	72.01		0.00		1.75	721.95	76.00	76.67		7.39		8.34	681.81
89	40	800	25	5	2900.14		2958.20	2961.60		2.00		2.12	4066.00	3015.87	3028.18		3.99		4.41	4098.99
90	40	800	50	5	1345.70		1368.30	1376.92		1.68		2.32	1940.25	1388.13	1417.07		3.15		5.30	1724.85
91	40	800	25	10	2200.57		2256.90	2275.11		2.56		3.39	5034.75	2240.43	2289.25		1.81		4.03	4833.69
92	40	800	50	10	1025.16		1046.24	1053.99		2.06		2.81	5195.63	1082.93	1089.95		5.64		6.32	5038.89
93	40	800	25	20	1404.16		1466.95	1476.93		4.47		5.18	4535.68	1438.95	1496.48		2.48		6.57	4334.43
94	40	800	50	20	616.58		632.08	639.69		2.51		3.75	4460.79	638.86	647.34		3.61		4.99	4296.48
95	40	800	25	40	396.92		409.05	416.91		3.06		5.04	4633.56	413.59	417.33		4.20		5.14	4288.36
96	40	800	50	40	200.94		209.20	212.08		4.11		5.54	4157.34	210.13	219.63		4.57		9.30	3822.62
97	80	400	25	5	1546.23		1591.74	1599.64		2.94		3.45	1282.88	1567.95	1588.78		1.40		2.75	1248.87
98	80	400	50	5	1048.56		1048.56	1060.19		0.00		1.11	725.72	1098.42	1141.76		4.75		8.89	683.61
99	80	400	25	10	1216.74		1298.78	1301.20		6.74		6.94	2559.89	1244.85	1288.51		2.31		5.90	2312.22
100	80	400	50	10	760.61		773.12	775.61		1.64		1.97	757.43	768.82	789.31		1.08		3.77	716.06
101	80	400	25	20	565.49		582.31	586.22		2.97		3.67	1828.95	585.78	598.79		3.59		5.89	1780.76
102	80	400	50	20	372.05		379.88	380.56		2.10		2.29	1364.48	387.50	395.16		4.15		6.21	1219.13
103	80	400	25	40	131.75		135.00	136.00		2.47		3.23	1309.71	143.16	145.52		8.66		10.45	1310.09
104	80	400	50	40	95.84		95.84	97.76		0.00		2.00	4057.13	102.70	104.34		7.16		8.86	4044.84

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-2							LNS-3						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
1	5	25	25	5	141.01	141.01	**142.11**	143.11	0.78	0.78	1.49	1.49	1.10	**141.01**	146.23	0.00	0.00	3.70	3.70	1.21
2	5	25	50	5	161.62	161.62	162.80	163.66	0.73	0.73	1.26	1.26	1.60	161.62	162.35	0.00	0.00	0.45	0.45	1.77
3	5	25	25	10	182.14	182.14	182.14	182.74	0.00	0.00	0.33	0.33	1.93	182.89	183.66	0.41	0.41	0.83	0.83	1.98
4	5	25	50	10	195.80	195.80	196.27	199.12	0.24	0.24	1.69	1.69	1.35	196.21	196.88	0.21	0.21	0.55	0.55	1.74
5	5	25	25	20	111.65	111.65	111.65	112.91	0.00	0.00	1.13	1.13	1.84	115.31	116.65	3.28	3.28	4.48	4.48	2.22
6	5	25	50	20	103.18	103.18	103.18	104.06	0.00	0.00	0.85	0.85	1.74	104.39	107.73	1.18	1.18	4.41	4.41	2.11
7	5	25	25	40	7.63	7.63	7.97	8.15	4.46	4.46	6.81	6.81	1.24	7.69	7.89	0.78	0.78	3.40	3.40	2.17
8	5	25	50	40	25.64	25.64	25.83	26.10	0.74	0.74	1.79	1.79	1.48	26.98	27.61	5.22	5.22	7.68	7.68	1.58
9	5	50	25	5	286.68	286.68	287.07	287.40	0.14	0.14	0.25	0.25	1.91	288.34	291.44	0.58	0.58	1.66	1.66	1.87
10	5	50	50	5	197.20	197.20	199.62	204.41	1.23	1.23	3.66	3.66	1.52	198.19	200.87	0.50	0.50	1.86	1.86	1.68
11	5	50	25	10	193.55	193.55	193.55	199.27	0.00	0.00	2.96	2.96	2.04	194.73	197.77	0.61	0.61	2.18	2.18	1.88
12	5	50	50	10	215.86	215.85	216.15	217.67	0.14	0.13	0.84	0.84	1.92	215.85	218.56	0.00	0.00	1.26	1.25	2.00
13	5	50	25	20	130.53	130.53	133.41	134.16	2.21	2.21	2.78	2.78	3.36	131.68	133.96	0.88	0.88	2.63	2.63	3.32
14	5	50	50	20	96.26	96.26	99.50	101.02	3.37	3.37	4.94	4.94	3.15	97.80	98.00	1.60	1.60	1.81	1.81	3.86
15	5	50	25	40	12.89	12.89	13.28	13.61	3.05	3.05	5.58	5.58	3.77	13.36	13.70	3.64	3.64	6.29	6.29	3.95
16	5	50	50	40	30.24	30.24	31.50	31.93	4.16	4.16	5.59	5.59	3.02	32.02	32.48	5.88	5.88	7.40	7.40	3.53
17	5	100	25	5	360.35	360.35	373.87	380.23	3.75	3.75	5.52	5.52	3.16	360.35	368.50	0.00	0.00	2.26	2.26	3.23
18	5	100	50	5	304.23	304.23	311.41	315.28	2.36	2.36	3.63	3.63	2.90	313.72	316.05	3.12	3.12	3.88	3.88	3.05
19	5	100	25	10	294.21	294.21	302.74	305.85	2.90	2.90	3.96	3.96	4.60	306.94	309.41	4.33	4.33	5.17	5.17	4.65
20	5	100	50	10	229.41	229.41	235.06	236.80	2.46	2.46	3.22	3.22	3.99	230.22	237.81	0.35	0.35	3.66	3.66	4.33
21	5	100	25	20	134.95	134.95	138.62	140.96	2.72	2.72	4.45	4.45	7.88	139.41	140.37	3.30	3.30	4.02	4.02	8.23
22	5	100	50	20	144.41	144.41	145.39	148.10	0.68	0.68	2.55	2.55	3.56	149.17	153.10	3.30	3.30	6.02	6.02	3.83
23	5	100	25	40	58.95	58.95	58.95	59.46	0.00	0.00	0.87	0.87	9.80	58.95	60.11	0.00	0.00	1.97	1.97	10.29
24	5	100	50	40	39.44	39.44	41.61	42.09	5.51	5.51	6.72	6.72	7.40	41.12	41.94	4.26	4.26	6.34	6.34	7.98
25	10	50	25	5	242.85	242.85	243.01	248.15	0.07	0.07	2.18	2.18	3.96	242.85	247.48	0.00	0.00	1.91	1.91	4.57
26	10	50	50	5	282.12	282.12	285.43	288.73	1.17	1.17	2.34	2.34	3.67	282.12	289.65	0.00	0.00	2.67	2.67	3.74
27	10	50	25	10	244.54	244.54	251.49	256.56	2.84	2.84	4.92	4.92	5.55	244.68	249.87	0.06	0.06	2.18	2.18	5.60
28	10	50	50	10	288.33	283.33	296.55	298.96	4.67	2.85	5.52	3.69	4.12	288.33	289.36	1.76	0.00	2.13	0.36	4.32

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-2							LNS-3						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
29	10	50	25	20	108.98	108.98	112.89	114.60	3.59	3.59	5.16	5.16	5.22	110.34	112.30	1.25	1.25	3.04	3.04	5.63
30	10	50	50	20	157.48	157.48	159.87	160.94	1.52	1.52	2.20	2.20	5.08	160.28	161.29	1.78	1.78	2.42	2.42	5.19
31	10	50	25	40	32.25	32.25	33.09	33.62	2.60	2.60	4.24	4.24	6.06	32.76	32.91	1.59	1.59	2.06	2.06	5.73
32	10	50	50	40	36.66	36.66	38.94	39.79	6.22	6.22	8.53	8.53	5.48	37.81	38.27	3.15	3.15	4.39	4.39	5.76
33	10	100	25	5	403.18	403.18	403.18	410.42	0.00	0.00	1.80	1.80	2.81	404.21	410.19	0.26	0.26	1.74	1.74	3.18
34	10	100	50	5	296.53	296.53	296.53	299.91	0.00	0.00	1.14	1.14	2.59	299.76	307.47	1.09	1.09	3.69	3.69	2.69
35	10	100	25	10	388.87	388.87	395.39	398.10	1.68	1.68	2.37	2.37	9.03	388.87	397.93	0.00	0.00	2.33	2.33	9.73
36	10	100	50	10	294.80	294.80	305.40	307.08	3.60	3.60	4.17	4.16	9.96	307.16	309.17	4.19	4.19	4.88	4.88	10.40
37	10	100	25	20	178.28	178.28	178.28	181.71	0.00	0.00	1.92	1.92	10.04	178.28	181.65	0.00	0.00	1.89	1.89	10.80
38	10	100	50	20	175.96	175.96	182.37	184.26	3.64	3.64	4.72	4.72	11.52	178.68	182.46	1.55	1.55	3.69	3.69	11.73
39	10	100	25	40	57.50	57.50	61.01	61.84	6.10	6.10	7.55	7.55	11.93	59.66	60.05	3.75	3.75	4.43	4.43	11.50
40	10	100	50	40	31.89	31.89	32.61	33.23	2.27	2.27	4.19	4.19	10.77	32.92	34.02	3.23	3.23	6.68	6.68	11.40
41	10	200	25	5	735.27	735.27	735.27	753.21	0.00	0.00	2.44	2.44	10.84	735.27	746.43	0.00	0.00	1.52	1.52	14.39
42	10	200	50	5	512.16	506.06	513.50	527.48	1.47	0.26	4.23	2.99	10.26	**511.42**	515.79	1.06	-0.14	1.92	0.71	9.86
43	10	200	25	10	513.00	513.00	522.09	528.23	1.77	1.77	2.97	2.97	35.26	513.00	522.79	0.00	0.00	1.91	1.91	38.20
44	10	200	50	10	475.21		484.13	489.14		1.88		2.93	15.16	489.71	491.31		3.05		3.39	16.41
45	10	200	25	20	347.29		362.85	369.00		4.48		6.25	30.43	360.99	361.91		3.94		4.21	33.36
46	10	200	50	20	217.46		217.46	221.37		0.00		1.80	27.26	225.17	229.47		3.55		5.52	29.77
47	10	200	25	40	102.93		103.49	105.32		0.54		2.33	43.61	109.19	111.24		6.08		8.07	45.71
48	10	200	50	40	55.05		56.56	57.92		2.74		5.21	17.28	56.92	58.05		3.39		5.45	19.19
49	20	100	25	5	520.24		529.83	536.31		1.84		3.09	11.86	532.00	536.55		2.26		3.14	13.16
50	20	100	50	5	420.64		423.76	430.18		0.74		2.27	7.01	420.64	432.54		0.00		2.83	7.02
51	20	100	25	10	422.21		434.40	439.55		2.89		4.11	10.55	437.38	440.31		3.59		4.29	11.81
52	20	100	50	10	360.86		368.94	372.27		2.24		3.16	9.92	368.17	375.38		2.03		4.02	10.12
53	20	100	25	20	245.17		246.21	251.80		0.42		2.70	12.48	245.17	247.87		0.00		1.10	13.03
54	20	100	50	20	185.06		190.20	192.39		2.78		3.96	8.95	191.67	195.26		3.57		5.51	9.11
55	20	100	25	40	52.52		52.98	53.93		0.88		2.68	14.23	55.00	55.60		4.72		5.87	14.95
56	20	100	50	40	19.05		19.76	20.57		3.72		7.99	31.22	19.95	20.57		4.72		7.96	31.96

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-2							LNS-3						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
57	20	200	25	5	903.84		924.75	930.68		2.31		2.97	15.20	924.01	930.95		2.23		3.00	16.35
58	20	200	50	5	485.65		498.41	504.81		2.63		3.94	33.89	497.39	497.99		2.42		2.54	35.97
59	20	200	25	10	616.93		631.04	635.51		2.29		3.01	35.11	621.84	627.11		0.80		1.65	37.40
60	20	200	50	10	462.31		480.38	483.12		3.91		4.50	23.55	476.09	476.92		2.98		3.16	25.31
61	20	200	25	20	373.21		378.32	385.19		1.37		3.21	63.16	373.21	381.28		0.00		2.16	66.25
62	20	200	50	20	250.75		251.49	259.23		0.30		3.38	33.47	259.68	267.47		3.56		6.67	34.60
63	20	200	25	40	93.01		96.41	96.48		3.66		3.73	85.08	97.97	99.31		5.33		6.77	87.42
64	20	200	50	40	45.40		45.60	47.75		0.43		5.17	42.49	45.40	46.85		0.00		3.19	45.65
65	20	400	25	5	1323.35		1376.10	1379.16		3.99		4.22	309.73	1343.98	1355.24		1.56		2.41	331.35
66	20	400	50	5	733.54		733.52	756.63		0.00		3.15	45.89	755.96	761.27		3.06		3.78	50.57
67	20	400	25	10	975.12		994.43	1008.63		1.98		3.44	215.01	1000.30	1002.66		2.58		2.82	227.17
68	20	400	50	10	614.67		636.19	640.29		3.50		4.17	94.45	629.16	632.33		2.36		2.87	97.69
69	20	400	25	20	763.76		790.57	801.05		3.51		4.88	191.26	765.61	786.08		0.24		2.92	202.76
70	20	400	50	20	298.47		309.45	313.11		3.68		4.91	116.14	312.61	317.20		4.74		6.28	124.58
71	20	400	25	40	239.58		242.11	246.80		1.06		3.01	390.95	245.48	247.00		2.46		3.10	407.66
72	20	400	50	40	84.49		89.00	89.69		5.34		6.15	157.71	87.06	89.31		3.04		5.71	165.30
73	40	200	25	5	831.94		848.97	855.53		2.05		2.83	88.03	859.02	871.09		3.25		4.71	94.18
74	40	200	50	5	593.35		609.84	618.71		2.78		4.27	61.72	609.72	619.69		2.76		4.44	64.94
75	40	200	25	10	728.44		736.18	740.47		1.06		1.65	1007.86	750.38	751.43		3.01		3.16	1054.67
76	40	200	50	10	481.05		499.61	501.59		3.86		4.27	142.01	503.17	505.53		4.60		5.09	146.20
77	40	200	25	20	339.75		341.93	349.69		0.64		2.92	219.83	353.54	354.82		4.06		4.44	229.82
78	40	200	50	20	273.88		276.71	282.68		1.03		3.21	62.43	273.88	279.15		0.00		1.92	68.07
79	40	200	25	40	76.77		79.97	81.79		4.17		6.54	160.37	78.72	81.67		2.54		6.39	169.17
80	40	200	50	40	58.46		60.19	60.21		2.96		2.99	100.16	61.38	62.03		5.00		6.10	108.34
81	40	400	25	5	1407.05		1470.42	1490.30		4.50		5.92	427.48	1450.57	1474.15		3.09		4.77	454.25
82	40	400	50	5	858.80		858.80	874.59		0.00		1.84	823.05	865.80	887.29		0.82		3.32	866.00
83	40	400	25	10	891.02		900.32	921.21		1.04		3.39	637.24	911.47	919.93		2.30		3.24	657.18
84	40	400	50	10	757.42		785.30	791.07		3.68		4.44	504.70	773.30	775.50		2.10		2.39	529.37

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-2							LNS-3						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
85	40	400	25	20	586.29		603.29	612.20		2.90		4.42	972.37	586.29	598.43		0.00		2.07	1013.50
86	40	400	50	20	395.95		404.83	410.80		2.24		3.75	358.96	395.95	397.36		0.00		0.36	374.16
87	40	400	25	40	195.33		199.84	202.63		2.31		3.74	1653.99	204.00	204.17		4.44		4.53	1743.68
88	40	400	50	40	70.77		74.70	75.42		5.55		6.58	752.67	74.00	75.78		4.56		7.08	790.02
89	40	800	25	5	2900.14		2985.26	2999.62		2.93		3.43	4457.17	2954.01	2956.78		1.86		1.95	4590.61
90	40	800	50	5	1345.70		1375.09	1386.83		2.18		3.06	1986.41	1395.55	1416.66		3.70		5.27	2073.59
91	40	800	25	10	2200.57		2245.40	2266.51		2.04		3.00	5222.63	2233.52	2263.55		1.50		2.86	5546.70
92	40	800	50	10	1025.16		1030.22	1064.39		0.49		3.83	5406.53	1040.43	1057.61		1.49		3.17	5631.97
93	40	800	25	20	1404.16		1454.79	1467.27		3.61		4.49	4589.12	1461.75	1503.41		4.10		7.07	4849.87
94	40	800	50	20	616.58		632.84	638.77		2.64		3.60	4573.13	630.88	635.21		2.32		3.02	4709.51
95	40	800	25	40	396.92		409.08	416.61		3.06		4.96	4632.69	396.92	411.33		0.00		3.63	5056.97
96	40	800	50	40	200.94		211.42	212.24		5.22		5.63	4249.70	210.91	215.94		4.96		7.47	4583.92
97	80	400	25	5	1546.23		1576.19	1629.29		1.94		5.37	1351.98	1579.47	1593.70		2.15		3.07	1333.83
98	80	400	50	5	1048.56		1081.76	1086.32		3.17		3.60	750.38	1089.72	1120.35		3.93		6.85	742.53
99	80	400	25	10	1216.74		1268.51	1287.65		4.25		5.83	2719.15	1265.87	1298.57		4.04		6.73	2619.44
100	80	400	50	10	760.61		767.94	785.03		0.96		3.21	742.44	763.35	779.36		0.36		2.47	785.41
101	80	400	25	20	565.49		582.74	589.39		3.05		4.23	1958.91	580.51	587.70		2.66		3.93	1988.42
102	80	400	50	20	372.05		385.75	387.89		3.68		4.26	1415.04	379.37	388.76		1.97		4.49	1414.73
103	80	400	25	40	131.75		139.07	140.43		5.56		6.59	1403.39	136.82	138.77		3.85		5.33	1707.68
104	80	400	50	40	95.84		98.57	100.90		2.85		5.28	4393.12	102.14	102.70		6.57		7.15	4830.29

							LNS-4							LNS-5						
							cost						Time(s)	cost						Time(s)
ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
1	5	25	25	5	141.01	141.01	**141.01**	**142.19**	0.00	0.00	0.84	0.84	1.28	**141.01**	144.20	**0.00**	**0.00**	2.26	2.26	1.09
2	5	25	50	5	161.62	161.62	162.28	163.37	0.41	0.41	1.08	1.08	1.84	161.62	164.03	0.00	0.00	1.49	1.49	1.67
3	5	25	25	10	182.14	182.14	186.07	186.29	2.16	2.16	2.28	2.28	2.08	182.76	186.83	0.34	0.34	2.58	2.58	1.88
4	5	25	50	10	195.80	195.80	195.80	197.99	0.00	0.00	1.12	1.12	1.90	195.97	203.21	0.09	0.09	3.79	3.79	1.61
5	5	25	25	20	111.65	111.65	112.90	114.10	1.12	1.12	2.19	2.19	2.33	111.78	114.17	0.12	0.12	2.26	2.26	2.05
6	5	25	50	20	103.18	103.18	104.35	105.98	1.13	1.13	2.71	2.71	2.28	104.83	106.00	1.60	1.60	2.73	2.73	1.94
7	5	25	25	40	7.63	7.63	7.92	7.98	3.80	3.80	4.59	4.59	1.90	7.83	8.12	2.59	2.59	6.42	6.42	1.97
8	5	25	50	40	25.64	25.64	26.67	27.03	4.02	4.02	5.42	5.42	1.68	27.09	28.22	5.66	5.66	10.06	10.06	1.57
9	5	50	25	5	286.68	286.68	289.37	290.18	0.94	0.94	1.22	1.22	2.02	287.41	290.81	0.25	0.25	1.44	1.44	1.82
10	5	50	50	5	197.20	197.20	200.91	209.27	1.88	1.88	6.12	6.12	1.90	206.18	208.62	4.55	4.55	5.79	5.79	1.68
11	5	50	25	10	193.55	193.55	194.23	195.19	0.35	0.35	0.85	0.85	2.12	194.32	195.05	0.40	0.40	0.77	0.77	1.91
12	5	50	50	10	215.86	215.85	215.86	223.08	0.00	0.00	3.35	3.35	2.02	215.86	217.69	0.00	0.00	0.85	0.85	2.14
13	5	50	25	20	130.53	130.53	131.58	136.06	0.81	0.81	4.24	4.24	3.54	136.14	138.98	4.30	4.30	6.47	6.47	3.55
14	5	50	50	20	96.26	96.26	99.93	101.51	3.81	3.81	5.46	5.46	2.98	99.86	104.76	3.74	3.74	8.83	8.83	3.74
15	5	50	25	40	12.89	12.89	13.56	13.95	5.20	5.20	8.22	8.22	3.91	13.08	13.49	1.49	1.49	4.65	4.65	3.94
16	5	50	50	40	30.24	30.24	31.97	32.98	5.73	5.73	9.05	9.05	3.83	31.02	31.54	2.58	2.58	4.31	4.31	3.37
17	5	100	25	5	360.35	360.35	360.35	364.08	0.00	0.00	1.04	1.04	3.52	370.70	372.87	2.87	2.87	3.48	3.48	3.34
18	5	100	50	5	304.23	304.23	312.86	314.28	2.84	2.84	3.30	3.30	3.28	313.27	315.26	2.97	2.97	3.63	3.63	3.00
19	5	100	25	10	294.21	294.21	297.49	306.31	1.12	1.12	4.11	4.11	4.91	300.18	303.18	2.03	2.03	3.05	3.05	4.22
20	5	100	50	10	229.41	229.41	234.42	238.96	2.19	2.19	4.16	4.16	4.34	236.35	241.20	3.02	3.02	5.14	5.14	3.67
21	5	100	25	20	134.95	134.95	134.95	139.83	0.00	0.00	3.61	3.61	9.10	134.95	137.98	0.00	0.00	2.25	2.25	7.14
22	5	100	50	20	144.41	144.41	147.81	152.24	2.35	2.35	5.42	5.42	4.08	146.86	149.73	1.70	1.70	3.68	3.68	3.22
23	5	100	25	40	58.95	58.95	60.74	61.94	3.03	3.03	5.06	5.06	10.28	62.13	62.48	5.39	5.39	5.99	5.99	8.89
24	5	100	50	40	39.44	39.44	42.02	42.75	6.54	6.54	8.40	8.40	8.91	39.95	41.72	1.29	1.29	5.79	5.79	7.01
25	10	50	25	5	242.85	242.85	246.12	251.23	1.35	1.35	3.45	3.45	5.18	252.32	256.54	3.90	3.90	5.64	5.64	3.83
26	10	50	50	5	282.12	282.12	282.12	284.03	0.00	0.00	0.68	0.68	4.10	289.12	292.51	2.48	2.48	3.68	3.68	3.36
27	10	50	25	10	244.54	244.54	252.33	256.09	3.19	3.19	4.72	4.72	5.24	251.51	254.49	2.85	2.85	4.07	4.07	4.76
28	10	50	50	10	288.33	283.33	288.33	293.06	1.76	0.00	3.43	1.64	4.37	299.83	304.29	5.82	3.99	7.40	5.53	3.69

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
29	10	50	25	20	108.98	108.98	113.47	115.46	4.12	4.12	5.95	5.95	5.63	114.36	115.18	4.94	4.94	5.69	5.69	4.96
30	10	50	50	20	157.48	157.48	161.70	164.29	2.68	2.68	4.32	4.32	6.16	160.84	162.98	2.13	2.13	3.49	3.49	4.55
31	10	50	25	40	32.25	32.25	33.56	34.92	4.05	4.05	8.29	8.29	6.77	34.87	36.02	8.11	8.11	11.68	11.68	5.00
32	10	50	50	40	36.66	36.66	37.51	38.68	2.33	2.33	5.51	5.51	6.26	38.80	39.80	5.85	5.85	8.57	8.57	4.98
33	10	100	25	5	403.18	403.18	404.91	408.79	0.43	0.43	1.39	1.39	3.31	407.45	411.16	1.06	1.06	1.98	1.98	2.88
34	10	100	50	5	296.53	296.53	311.23	314.96	4.96	4.96	6.21	6.21	3.20	311.52	317.32	5.05	5.05	7.01	7.01	2.46
35	10	100	25	10	388.87	388.87	389.95	397.80	0.28	0.28	2.30	2.30	12.61	390.16	393.15	0.33	0.33	1.10	1.10	8.66
36	10	100	50	10	294.80	294.80	311.63	313.58	5.71	5.71	6.37	6.37	11.70	304.39	308.43	3.26	3.25	4.62	4.62	9.03
37	10	100	25	20	178.28	178.28	179.21	181.21	0.52	0.52	1.65	1.65	11.11	178.28	179.08	0.00	0.00	0.45	0.45	9.35
38	10	100	50	20	175.96	175.96	182.97	184.71	3.98	3.98	4.97	4.97	12.42	186.40	188.11	5.93	5.93	6.90	6.90	10.33
39	10	100	25	40	57.50	57.50	58.90	60.05	2.43	2.43	4.43	4.43	12.33	59.87	60.19	4.12	4.12	4.68	4.68	10.47
40	10	100	50	40	31.89	31.89	31.89	32.87	0.00	0.00	3.08	3.08	12.42	31.89	32.79	0.00	0.00	2.82	2.82	9.97
41	10	200	25	5	735.27	735.27	744.39	749.77	1.24	1.24	1.97	1.97	14.57	754.64	759.03	2.63	2.63	3.23	3.23	12.18
42	10	200	50	5	512.16	506.06	526.26	530.49	3.99	2.75	4.83	3.58	11.71	514.87	529.23	1.74	0.53	4.58	3.33	8.89
43	10	200	25	10	513.00	513.00	522.88	527.01	1.93	1.93	2.73	2.73	38.93	522.30	525.13	1.81	1.81	2.36	2.36	32.41
44	10	200	50	10	475.21		475.21	489.94		0.00		3.10	16.90	489.55	492.26		3.02		3.59	13.81
45	10	200	25	20	347.29		362.29	366.33		4.32		5.48	34.16	361.68	363.95		4.14		4.80	29.84
46	10	200	50	20	217.46		217.46	222.57		0.00		2.35	28.56	220.42	221.31		1.36		1.77	28.26
47	10	200	25	40	102.93		107.08	111.00		4.03		7.84	45.47	102.93	104.65		0.00		1.67	42.48
48	10	200	50	40	55.05		57.50	58.87		4.44		6.93	17.61	55.94	58.58		1.62		6.41	19.20
49	20	100	25	5	520.24		532.31	537.86		2.32		3.39	12.91	525.31	530.10		0.97		1.89	12.72
50	20	100	50	5	420.64		434.06	439.94		3.19		4.59	7.26	432.80	434.60		2.89		3.32	6.94
51	20	100	25	10	422.21		436.38	441.24		3.36		4.51	11.90	441.18	441.93		4.49		4.67	12.12
52	20	100	50	10	360.86		366.82	378.96		1.65		5.01	12.83	367.74	368.62		1.91		2.15	10.01
53	20	100	25	20	245.17		245.17	254.97		0.00		4.00	13.30	248.94	251.19		1.54		2.46	13.05
54	20	100	50	20	185.06		188.01	189.17		1.59		2.22	9.28	187.85	191.11		1.51		3.27	9.05
55	20	100	25	40	52.52		54.95	55.42		4.62		5.53	14.36	52.56	54.57		0.08		3.90	14.37
56	20	100	50	40	19.05		19.05	19.94		0.00		4.66	31.09	19.71	21.13		3.46		10.92	31.74

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
57	20	200	25	5	903.84		924.69	931.30		2.31		3.04	14.82	903.84	939.95		0.00		4.00	15.83
58	20	200	50	5	485.65		**484.10**	511.97		-0.32		5.42	36.11	503.90	506.89		3.76		4.37	35.11
59	20	200	25	10	616.93		633.70	635.86		2.72		3.07	40.05	639.18	642.18		3.61		4.09	36.18
60	20	200	50	10	462.31		478.54	484.18		3.51		4.73	25.65	478.29	483.44		3.46		4.57	25.38
61	20	200	25	20	373.21		377.16	385.99		1.06		3.43	67.24	380.56	386.28		1.97		3.50	64.27
62	20	200	50	20	250.75		251.83	255.98		0.43		2.08	35.86	251.02	256.01		0.11		2.10	32.95
63	20	200	25	40	93.01		96.02	97.38		3.23		4.70	92.91	97.71	98.46		5.05		5.86	84.08
64	20	200	50	40	45.40		46.99	47.09		3.50		3.71	45.98	46.38	48.42		2.17		6.64	42.77
65	20	400	25	5	1323.35		1384.95	1397.52		4.65		5.60	338.89	1338.09	1351.25		1.11		2.11	317.69
66	20	400	50	5	733.54		750.21	757.58		2.27		3.28	51.07	754.94	758.31		2.92		3.38	46.68
67	20	400	25	10	975.12		1006.38	1008.45		3.21		3.42	228.05	980.21	981.84		0.52		0.69	220.42
68	20	400	50	10	614.67		637.59	644.94		3.73		4.92	93.12	638.81	642.49		3.93		4.53	92.05
69	20	400	25	20	763.76		790.89	802.65		3.55		5.09	203.74	776.50	804.78		1.67		5.37	194.47
70	20	400	50	20	298.47		310.40	314.77		4.00		5.46	123.10	314.40	319.68		5.34		7.11	117.46
71	20	400	25	40	239.58		239.58	240.81		0.00		0.51	394.51	245.63	247.01		2.52		3.10	396.14
72	20	400	50	40	84.49		87.14	87.74		3.14		3.85	180.23	87.55	88.63		3.62		4.90	158.62
73	40	200	25	5	831.94		848.74	850.94		2.02		2.28	101.58	873.95	875.42		5.05		5.23	89.47
74	40	200	50	5	593.35		613.33	616.61		3.37		3.92	62.98	593.68	599.20		0.05		0.99	62.00
75	40	200	25	10	728.44		749.62	760.52		2.91		4.40	1091.78	728.44	744.41		0.00		2.19	1014.25
76	40	200	50	10	481.05		505.59	511.52		5.10		6.33	157.61	516.95	518.26		7.46		7.74	144.57
77	40	200	25	20	339.75		342.11	355.38		0.69		4.60	236.35	350.19	353.94		3.07		4.18	228.41
78	40	200	50	20	273.88		282.03	282.23		2.98		3.05	69.95	280.21	281.57		2.31		2.81	66.47
79	40	200	25	40	76.77		78.81	79.14		2.66		3.08	176.84	78.35	79.49		2.06		3.54	165.47
80	40	200	50	40	58.46		59.67	61.29		2.07		4.85	114.82	60.44	60.95		3.38		4.26	100.03
81	40	400	25	5	1407.05		1484.04	1490.99		5.47		5.97	464.42	1466.60	1481.56		4.23		5.30	443.75
82	40	400	50	5	858.80		888.07	897.51		3.41		4.51	880.34	883.15	888.65		2.84		3.48	843.86
83	40	400	25	10	891.02		898.80	902.16		0.87		1.25	673.48	920.98	927.86		3.36		4.13	652.17
84	40	400	50	10	757.42		787.19	791.97		3.93		4.56	536.36	797.91	801.48		5.35		5.82	518.48

ID	stop	stud	cap	wd	BKS(MH)	BKS(exact)	LNS-4							LNS-5						
							cost						Time(s)	cost						Time(s)
							Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time	Best sol	Avg sol	%Best Gap (exact)	%Best Gap(MH)	%Avg Gap (exact)	% Avg Gap(MH)	Avg time
85	40	400	25	20	586.29		601.47	606.03		2.59		3.37	1034.19	611.49	615.90		4.30		5.05	987.46
86	40	400	50	20	395.95		403.47	408.00		1.90		3.04	381.56	405.23	407.47		2.34		2.91	365.48
87	40	400	25	40	195.33		203.37	206.76		4.12		5.85	1775.91	201.93	205.51		3.38		5.21	1694.91
88	40	400	50	40	70.77		73.45	75.20		3.79		6.27	788.07	73.01	74.26		3.17		4.93	755.08
89	40	800	25	5	2900.14		2999.43	3041.80		3.42		4.88	4650.40	3028.92	3034.37		4.44		4.63	4411.96
90	40	800	50	5	1345.70		1345.64	1388.19		0.00		3.16	2135.60	1393.84	1400.01		3.58		4.04	2047.09
91	40	800	25	10	2200.57		2272.58	2288.36		3.27		3.99	5658.80	2200.57	2295.55		0.00		4.32	5217.78
92	40	800	50	10	1025.16		1087.79	1089.05		6.11		6.23	5808.31	1051.21	1055.33		2.54		2.94	5391.43
93	40	800	25	20	1404.16		1451.01	1464.80		3.34		4.32	4951.05	1436.22	1484.96		2.28		5.75	4635.86
94	40	800	50	20	616.58		634.36	639.47		2.88		3.71	4889.83	639.82	645.08		3.77		4.62	4629.46
95	40	800	25	40	396.92		405.10	411.36		2.06		3.64	5169.93	408.47	417.79		2.91		5.26	5000.96
96	40	800	50	40	200.94		207.25	209.02		3.14		4.02	4634.26	207.00	208.50		3.02		3.76	4312.12
97	80	400	25	5	1546.23		1546.23	1589.06		0.00		2.77	1362.64	1591.34	1602.19		2.92		3.62	1304.00
98	80	400	50	5	1048.56		1100.97	1112.34		5.00		6.08	763.05	1083.69	1098.19		3.35		4.73	709.76
99	80	400	25	10	1216.74		1280.54	1288.73		5.24		5.92	2704.81	1279.75	1297.65		5.18		6.65	2569.52
100	80	400	50	10	760.61		782.41	784.51		2.87		3.14	878.33	783.67	784.61		3.03		3.16	746.19
101	80	400	25	20	565.49		582.96	587.58		3.09		3.91	2009.07	589.39	599.75		4.23		6.06	1953.19
102	80	400	50	20	372.05		385.59	389.08		3.64		4.58	1678.28	385.10	388.11		3.51		4.32	1376.61
103	80	400	25	40	131.75		136.38	139.29		3.52		5.72	1763.71	135.70	138.80		3.00		5.35	1725.10
104	80	400	50	40	95.84		98.85	100.59		3.14		4.96	5049.86	99.86	104.57		4.19		9.11	4689.32

Appendix 5

Instance size	Problem characteristics				results			
	no of school	no of stop	capacity	Walking distance	Exact solution	Best sol meta	Avg time (ms)	%Best gap
1	1	10	25	5	135.43	135.43	35	0.00
2	1	10	50	5	242.62	247.56	97	2.04
3	1	10	25	10	151.34	155.19	71	2.54
4	1	10	50	10	224.68	224.68	46	0.00
5	1	10	25	15	219.45	223.12	71	1.67
6	1	10	50	15	177.50	177.50	50	0.00
7	1	10	25	20	103.45	105.19	62	1.68
8	1	10	50	20	155.47	155.47	69	0.00
9	1	10	25	25	136.35	138.90	65	1.87
10	1	10	50	25	65.98	68.34	113	3.58
11	2	20	25	5	161.45	161.45	186	0.00
12	2	20	50	5	267.21	272.12	214	1.84
13	2	20	25	10	226.06	228.90	242	1.26
14	2	20	50	10	326.34	330.23	252	1.19
15	2	20	25	15	244.49	244.50	260	0.00
16	2	20	50	15	234.68	243.12	244	3.60
17	2	20	25	20	247.42	250.21	220	1.13
18	2	20	50	20	171.99	178.23	212	3.63
19	2	20	25	25	167.22	169.12	174	1.14
20	2	20	50	25	187.82	190.45	272	1.40
21	3	30	25	5		794.80	383	
22	3	30	50	5		1012.46	345	
23	3	30	25	10		958.37	390	
24	3	30	50	10		1306.17	442	
25	3	30	25	15		1243.76	347	
26	3	30	50	15		1105.14	440	
27	3	30	25	20		863.20	428	

Instance size	Problem characteristics					results		
	no of school	no of stop	capacity	Walking distance	Exact solution	Best sol meta	Avg time (ms)	%Best gap
28	3	30	50	20		894.34	395	
29	3	30	25	25		835.51	314	
30	3	30	50	25		321.89	464	
31	4	40	25	5		1760.12	891	
32	4	40	50	5		2872.21	852	
33	4	40	25	10		1919.67	967	
34	4	40	50	10		1807.95	987	
35	4	40	25	15		2456.78	659	
36	4	40	50	15		2035.56	869	
37	4	40	25	20		1169.80	1,141	
38	4	40	50	20		1351.13	1,082	
39	4	40	25	25		1431.12	659	
40	4	40	50	25		912.34	904	
41	5	50	25	5		1897.23	3,016	
42	5	50	50	5		2182.69	3,000	
43	5	50	25	10		1987.59	2,667	
44	5	50	50	10		2878.45	2,723	
45	5	50	25	15		3114.09	2,104	
46	5	50	50	15		2304.29	2,396	
47	5	50	25	20		1455.34	3,903	
48	5	50	50	20		1678.32	3,345	
49	5	50	25	25		1790.32	1,785	
50	5	50	50	25		1903.34	2,585	
51	6	60	25	5		1900.34	11,912	
52	6	60	50	5		2234.19	14,565	
53	6	60	25	10		2543.19	8,142	
54	6	60	50	10		2664.45	9,970	
55	6	60	25	15		2732.21	7,805	
56	6	60	50	15		2021.45	9,801	
57	6	60	25	20		1450.39	15,648	

Instance size	Problem characteristics					results		
	no of school	no of stop	capacity	Walking distance	Exact solution	Best sol meta	Avg time (ms)	%Best gap
58	6	60	50	20		2097.34	13,211	
59	6	60	25	25		1891.13	7,219	
60	6	60	50	25		2793.12	9,466	
61	7	70	25	5		2121.34	43,367	
62	7	70	50	5		3043.12	63,767	
63	7	70	25	10		2570.57	38,718	
64	7	70	50	10		4096.50	42,285	
65	7	70	25	15		3362.22	30,057	
66	7	70	50	15		3021.54	34,506	
67	7	70	25	20		3098.45	69,988	
68	7	70	50	20		2560.87	62,995	
69	7	70	25	25		2272.12	32,626	
70	7	70	50	25		2341.34	45,017	
71	8	80	25	5		3023.32	208,465	
72	8	80	50	5		3957.43	330,594	
73	8	80	25	10		3652.32	164,202	
74	8	80	50	10		4432.12	178,967	
75	8	80	25	15		5034.21	127,729	
76	8	80	50	15		4321.14	146,634	
77	8	80	25	20		4567.23	347,841	
78	8	80	50	20		3457.21	309,300	
79	8	80	25	25		3094.32	155,156	
80	8	80	50	25		3101.15	240,337	
81	9	90	25	5		3987.21	763,417	
82	9	90	50	5		4976.54	1,056,142	
83	9	90	25	10		4674.23	601,323	
84	9	90	50	10		4867.12	692,230	
85	9	90	25	15		4523.13	523,622	
86	9	90	50	15		3987.12	614,957	
87	9	90	25	20		4231.23	1,418,512	

Instance size	Problem characteristics				results			
	no of school	no of stop	capacity	Walking distance	Exact solution	Best sol meta	Avg time (ms)	%Best gap
88	9	90	50	20		4578.20	1,328,981	
89	9	90	25	25		4309.21	673,320	
90	9	90	50	25		3211.98	1,039,881	
91	10	100	25	5		4219.20	2,396,318	
92	10	100	50	5		5396.95	4,116,779	
93	10	100	25	10		4748.18	1,949,401	
94	10	100	50	10		5763.43	2,938,713	
95	10	100	25	15		5626.40	1,821,001	
96	10	100	50	15		4473.65	2,114,905	
97	10	100	25	20		4514.68	4,276,218	
98	10	100	50	20		4352.37	4,091,803	
99	10	100	25	25		4228.80	2,408,016	
100	10	100	50	25		4589.78	3,968,672	

Appendix 6

Problem characteristics					Metaheuristic		Deviation			Time (ms)	
Instance size	*No of school*	*No of stop*	*Capacity*	*Exact method*	*First scenario*	*Second scenario*	*Deviation-first scenario from exact*	*Deviation-second scenario from exact*	*Percentage Gap between two scenarios*	*Time first scenario*	*Time Second scenario*
1	1	10	20	508.48	517.42	508.48	1.76%	0.00%	1.759%	233.07	234.62
2	1	10	25	472.94	495.66	483.27	4.80%	2.19%	2.563%	223.18	227.13
3	1	10	30	441.75	457.57	441.75	3.58%	0.00%	3.581%	222.71	221.19
4	1	10	35	407.14	415.27	413.82	2.00%	1.64%	0.349%	212.31	218.00
5	1	10	40	379.90	379.94	379.90	0.01%	0.00%	0.011%	201.54	205.35
6	1	10	45	354.21	362.05	361.74	2.21%	2.13%	0.085%	168.47	171.12
7	1	10	50	346.14	365.20	369.63	5.51%	6.78%	-1.198%	151.59	153.60
8	2	20	20	516.30	533.67	525.03	3.37%	1.69%	1.646%	801.90	811.63
9	2	20	25	517.36	540.07	540.36	4.39%	4.44%	-0.054%	813.76	831.76
10	2	20	30	465.12	471.75	465.12	1.43%	0.00%	1.427%	786.90	818.00
11	2	20	35	394.29	404.64	400.03	2.62%	1.46%	1.152%	754.51	767.96
12	2	20	40	387.42	387.42	387.42	0.00%	0.00%	0.000%	768.45	777.83
13	2	20	45	365.36	370.57	365.36	1.43%	0.00%	1.427%	749.73	759.53
14	2	20	50	523.77	554.60	549.53	5.89%	4.92%	0.923%	777.55	792.77
15	3	30	20	2690.49	2791.86	2,770.75	3.77%	2.98%	0.762%	1,882.94	1,903.46
16	3	30	25	2815.25	2910.42	2,828.38	3.38%	0.47%	2.901%	1,920.68	1,929.92
17	3	30	30		2813.94	2,685.32	2.88%	1.79%	4.789%	1,813.77	1,875.44
18	3	30	35		2576.69	2,478.97			3.942%	1,760.28	1,830.40
19	3	30	40		2574.07	2,442.15			5.402%	1,838.89	1,809.79
20	3	30	45		2652.20	2,540.88			4.381%	2,140.86	2,117.35
21	3	30	50		2753.47	2,568.37			7.207%	2,445.68	2,489.19
22	4	40	20		6187.74	5,933.82			4.279%	2,208.62	2,264.60
23	4	40	25		5397.82	5,380.35			0.325%	2,076.96	2,101.22
24	4	40	30		4748.77	4,665.43			1.786%	1,942.84	1,966.11
25	4	40	35		4627.31	4,492.84			2.993%	1,986.20	2,072.31
26	4	40	40		2094.04	2,004.55			4.464%	1,381.60	1,398.64

Problem characteristics					Metaheuristic		Deviation			Time (ms)	
Instance size	***No of school***	***No of stop***	***Capacity***	***Exact method***	***First scenario***	***Second scenario***	***Deviation-first scenario from exact***	***Deviation-second scenario from exact***	***Percentage Gap between two scenarios***	***Time first scenario***	***Time Second scenario***
27	4	40	45		4961.57	4,869.46			1.892%	2,579.74	2,580.26
28	4	40	50		5398.40	5,434.95			-0.672%	3,181.44	3,188.88
29	5	50	20		7650.01	7,645.89			0.054%	6,469.21	6,692.40
30	5	50	25		7220.51	7,198.33			0.308%	6,959.11	6,968.47
31	5	50	30		7189.75	7,168.76			0.293%	7,556.36	7,557.80
32	5	50	35		6215.73	5,941.68			4.612%	7,206.24	7,154.60
33	5	50	40		5895.65	5,694.89			3.525%	6,997.28	7,092.17
34	5	50	45		4877.70	4,668.33			4.485%	6,482.34	6,675.51
35	5	50	50		4527.52	4,355.11			3.959%	6,829.01	6,923.93
36	6	60	20		4281.40	4,053.72			5.616%	22,836.83	23,232.36
37	6	60	25		3407.83	3,374.28			0.994%	20,944.32	21,457.46
38	6	60	30		5367.18	5,342.48			0.462%	28,614.99	28,968.10
39	6	60	35		4894.48	4,855.67			0.799%	30,852.85	31,574.81
40	6	60	40		2487.37	2,356.20			5.567%	22,513.96	23,267.05
41	6	60	45		5029.85	4,764.60			5.567%	37,205.38	37,686.81
42	6	60	50		4919.48	4,824.20			1.975%	39,855.45	40,825.53
43	7	70	20		8672.19	8,214.87			5.567%	115,021.60	121,796.37
44	7	70	25		6495.58	6,388.21			1.681%	101,513.44	103,726.19
45	7	70	30		6554.89	6,275.99			4.444%	107,115.53	109,779.50
46	7	70	35		3253.67	3,091.06			5.261%	79,129.12	78,767.66
47	7	70	40		2991.91	2,834.13			5.567%	77,436.38	77,255.18
48	7	70	45		3124.97	2,959.61			5.587%	78,531.93	80,374.29
49	7	70	50		7796.07	7,361.94			5.897%	138,612.74	141,107.77
50	8	80	20		13128.58	12,565.06			4.485%	496,285.01	505,163.56
51	8	80	25		13363.42	12,328.68			8.393%	515,016.36	524,235.16
52	8	80	30		12754.39	12,081.79			5.567%	525,407.46	543,810.66
53	8	80	35		12188.42	11,774.55			3.515%	545,230.65	560,308.24
54	8	80	40		11610.93	10,998.63			5.567%	538,866.86	552,613.97

Problem characteristics					Metaheuristic		Deviation			Time (ms)	
Instance size	*No of school*	*No of stop*	*Capacity*	*Exact method*	*First scenario*	*Second scenario*	*Deviation-first scenario from exact*	*Deviation-second scenario from exact*	*Percentage Gap between two scenarios*	*Time first scenario*	*Time Second scenario*
55	8	80	45		11228.50	10,667.32			5.261%	549,781.88	563,989.93
56	8	80	50		11012.74	10,150.45			8.495%	567,089.27	584,385.59
57	9	90	20		10858.43	10,396.41			4.444%	1,910,241.48	1,957,455.98
58	9	90	25		10715.12	10,172.80			5.331%	1,971,801.67	2,024,524.25
59	9	90	30		10167.99	10,121.21			0.462%	2,052,379.64	2,104,965.26
60	9	90	35		9427.20	9,548.79			-1.273%	2,020,067.42	2,187,338.56
61	9	90	40		9154.39	9,215.28			-0.661%	2,202,195.97	2,252,750.15
62	9	90	45		9630.85	9,248.19			4.138%	2,296,793.04	2,350,050.17
63	9	90	50		9365.56	8,917.47			5.025%	2,293,937.55	2,368,517.81
64	10	100	20		12369.85	11,890.01			4.036%	2,379,043.61	2,454,715.75
65	10	100	25		14139.01	13,568.03			4.208%	3,875,459.70	4,016,668.85
66	10	100	30		12801.97	12,126.86			5.567%	3,952,160.86	4,085,025.95
67	10	100	35		11905.98	11,399.39			4.444%	3,921,046.40	4,057,558.85
68	10	100	40		11920.24	11,291.63			5.567%	4,260,672.42	4,366,624.21
69	10	100	45		10257.58	9,716.65			5.567%	4,277,208.42	4,379,699.39
70	10	100	50		11013.81	10,899.98			1.044%	4,518,653.71	4,691,370.90
71	11	110	20		15480.36	14,821.69			4.444%	2,932,599.24	3,078,296.25
72	11	110	25		15592.69	14,907.37			4.597%	7,594,535.59	7,969,096.28
73	11	110	30		12913.37	12,363.92			4.444%	7,728,092.23	7,927,654.38
74	11	110	35		13614.00	13,129.71			3.689%	7,281,692.03	7,526,856.65
75	11	110	40		13864.98	13,342.59			3.915%	8,307,852.73	8,464,057.57
76	11	110	45		11123.89	10,520.99			5.730%	8,115,809.07	8,162,279.65
77	11	110	50		10932.81	10,340.27			5.730%	7,268,654.66	7,433,834.20

Appendix 7 (District 12)

Problem characteristics				Cost		Problem characteristics				Cost	
Instance size	*school no of*	*no of stop*	*Capacity*	*First scenario*	*Second scenario*	*size Instance*	*school no of*	*no of stop*	*Capacity*	*First scenario*	*Second scenario*
1	5	50	20	2,678	2,689	29	20	200	20	7,751	7,589
2	5	50	25	2,477	2,469	30	20	200	25	10,153	9,887
3	5	50	30	2,265	2,179	31	20	200	30	5,675	5,556
4	5	50	35	1,890	1,771	32	20	200	35	7,615	7,539
5	5	50	40	1,713	1,656	33	20	200	40	8,347	8,371
6	5	50	45	1,366	1,317	34	20	200	45	3,543	3,414
7	5	50	50	1,236	1,225	35	20	200	50	3,273	3,126
8	10	100	20	3,191	3,151	36	25	250	20	10,418	9,879
9	10	100	25	3,560	3,401	37	25	250	25	14,503	14,516
10	10	100	30	3,042	2,932	38	25	250	30	6,841	6,511
11	10	100	35	2,764	2,688	39	25	250	35	10,309	10,058
12	10	100	40	2,737	2,660	40	25	250	40	11,827	11,322
13	10	100	45	2,302	2,229	41	25	250	45	4,074	3,923
14	10	100	50	2,495	2,392	42	25	250	50	3,574	3,435
15	12	120	20	4,290	4,346	43	30	300	20	13,581	13,170
16	12	120	25	5,036	4,854	44	30	300	25	20,610	19,508
17	12	120	30	3,754	3,578	45	30	300	30	8,081	7,619
18	12	120	35	3,878	3,722	46	30	300	35	12,562	11,840
19	12	120	40	4,002	3,850	47	30	300	40	16,544	15,608
20	12	120	45	2,659	2,478	48	30	300	45	4,611	4,437
21	12	120	50	2,743	2,744	49	30	300	50	4,021	3,855
22	15	150	20	5,710	5,567	50	35	350	20	17,553	17,246
23	15	150	25	7,108	6,561	51	35	350	25	28,450	27,384

24	15	150	30	4,588	4,542	52	35	350	30	9,272	8,842
25	15	150	35	5,420	5,362	53	35	350	35	13,776	13,212
26	15	150	40	5,819	5,670	54	35	350	40	23,038	21,858
27	15	150	45	3,050	2,970	55	35	350	45	5,069	5,087
28	15	150	50	2,996	2,777	56	35	350	50	4,690	4,511

Appendix 7 (District 13)

Problem characteristics				Cost	
Instance size	*no of school*	*no of stop*	*capacity*	*first scenario*	*second scenario*
1	5	50	20	4150.1	4054.4
2	5	50	25	3454.9	3320.2
3	5	50	30	3054	2964
4	5	50	35	2723.9	2705.3
5	5	50	40	2628.1	2637.1
6	5	50	45	2053.4	1967.6
7	50	50	50	1980.5	1920.3
8	10	100	20	5168.6	4972
9	10	100	25	5535.4	5356.3
10	10	100	30	5517.7	5302.6
11	10	100	35	3895.2	3780.4
12	10	100	40	3843.3	3693.4
13	10	100	45	3210.5	3109.7
14	10	100	50	3515.3	3378.2
15	12	120	20	6253.7	6004
16	12	120	25	7642.4	7351.7
17	12	120	30	6538.2	6314.3
18	12	120	35	5415.1	5183.7
19	12	120	40	5507	5266.5
20	12	120	45	3261.4	3200.5
21	12	120	50	3884.8	3722.4

Problem characteristics				Cost	
Instance size	*no of school*	*no of stop*	*capacity*	*first scenario*	*second scenario*
29	20	120	0	11087.68	10991.24
30	20	120	0	14254.07	14026.52
31	20	120	0	9942.52	9541.432
32	20	120	0	10567.07	10233.98
33	20	120	0	11578.82	11213.84
34	20	120	0	4100.551	3971.294
35	20	120	0	4650.066	4566.663
36	25	120	0	14428.85	14144.54
37	25	120	0	20195.93	19754.52
38	25	120	0	12251.32	11995.52
39	25	120	0	13925.71	13513.45
40	25	120	0	15495.06	14888.92
41	25	120	0	4443.228	4227.969
42	25	120	0	5186.524	4864.414
43	30	120	0	19487.09	18724.8
44	30	120	0	27661.75	26842.85
45	30	120	0	15062.43	14587.63
46	30	120	0	19036.29	18472.74
47	30	120	0	22413.5	22189.37
48	30	120	0	6455.694	6456.353
49	30	120	0	5721.992	5746.029

22	15	120	0	8356.704	7951.85	50	35	120	0	25885.62	25069.66
23	15	120	0	10357.54	9942.634	51	35	120	0	36685.09	35514.68
24	15	120	0	8038.648	7982.45	52	35	120	0	17635.46	16562.36
25	15	120	0	7672.582	7595.856	53	35	120	0	25167.14	25167.14
26	15	120	0	7921.494	7671.794	54	35	120	0	31099.61	30791.7
27	15	120	0	3779.57	3659.347	55	35	120	0	6994.879	6761.027
28	15	120	0	4380.103	4236.174	56	35	120	0	6211.032	6015.249

Appendix 8

Type of Hazard	Severity	Probability			North	South	Center	Avg. Score
		North	South	Center	Rank1	Rank2	Rank3	
Safety risk in area								
The route leading to bus stop	4	5	6	6	20	24	24	22.7
Pedestrian crossing	3	4	4	5	12	12	15	13.0
The location of bus stop	5	4	4	6	20	20	30	23.3
Size of waiting area (bus stop)	7	5	6	5	35	42	35	37.3
Quality of place in waiting area	6	2	3	3	12	18	18	16.0
Healthy and safety risk								
Density of population	7	3	6	5	21	42	35	32.7
Prevalence to corona virus	8	3	6	5	24	48	40	37.3
Household Income	4	1	7	6	4	28	24	18.7
Traffic and road condition								
Complex intersection	4	3	4	4	12	16	16	14.7
High traffic volume	6	4	6	5	24	36	30	30.0
Traffic speed	7	2	3	5	14	21	35	23.3
Highway area	5	2	5	4	10	25	20	18.3

Appendix 9

ID	School	stop	cap	wd	P-HA			
					Exact solution	Metaheuristic	Exact (Gap)	Time (ms)
1	1	10	25	5	168.69	170.64	1.2%	183.83
2	1	10	50	5	316.58	319.30	0.9%	476.81
3	1	10	25	10	195.29	199.42	2.1%	341.61
4	1	10	50	10	265.09	265.09	0.0%	203.80
5	1	10	25	15	281.130	281.13	0.0%	321.13
6	1	10	50	15	218.27	220.48	1.0%	218.18
7	1	10	25	20	127.83	130.96	2.4%	246.94
8	1	10	50	20	189.67	189.67	0.0%	281.83
9	1	10	25	25	166.72	168.84	1.3%	243.37
10	1	10	50	25	86.14	88.16	2.3%	458.24
11	2	20	25	5	201.64	201.64	0.0%	677.96
12	2	20	50	5	348.27	365.22	4.9%	950.51
13	2	20	25	10	277.29	286.13	3.2%	1,084.98
14	2	20	50	10	365.11	386.37	5.8%	1,265.42
15	2	20	25	15	303.18	309.48	2.1%	1,367.90
16	2	20	50	15	310.17	321.24	3.6%	1,517.23
17	2	20	25	20	317.20	332.78	4.9%	1,272.31
18	2	20	50	20	217.03	228.64	5.3%	1,330.56
19	2	20	25	25		229.16		1,072.35
20	2	20	50	25		262.82		1,637.99
21	3	30	25	5		988.69		2,298.73
22	3	30	50	5		1254.93		2,014.19
23	3	30	25	10		1216.25		2,317.61

ID	School	stop	cap	wd	P-HA			
					Exact solution	Metaheuristic	Exact (Gap)	Time (ms)
24	3	30	50	10		1626.29		3,052.50
25	3	30	25	15		1544.17		2,351.55
26	3	30	50	15		1401.59		2,996.71
27	3	30	25	20		1089.76		3,025.66
28	3	30	50	20		1148.67		2,878.71
29	3	30	25	25		1078.93		2,270.02
30	3	30	50	25		406.73		3,186.63
31	4	40	25	5		2216.34		6,087.58
32	4	40	50	5		3618.56		4,928.99
33	4	40	25	10		2423.53		5,526.24
34	4	40	50	10		2301.20		4,896.02
35	4	40	25	15		3055.99		3,260.82
36	4	40	50	15		2571.68		3,745.01
37	4	40	25	20		1480.59		5,149.40
38	4	40	50	20		1707.33		4,652.23
39	4	40	25	25		1755.35		2,704.40
40	4	40	50	25		1152.69		3,979.90
41	5	50	25	5		2333.12		16,074.33
42	5	50	50	5		4503.13		19,772.68
43	5	50	25	10		2484.24		17,052.59
44	5	50	50	10		3650.59		14,684.70
45	5	50	25	15		3999.66		12,339.82
46	5	50	50	15		2929.55		15,259.26
47	5	50	25	20		1805.20		22,875.76
48	5	50	50	20		2119.03		20,265.91

ID	School	stop	cap	wd	P-HA			
					Exact solution	Metaheuristic	Exact (Gap)	Time (ms)
49	5	50	25	25		2237.47		8,599.10
50	5	50	50	25		2363.21		11,657.29
51	6	60	25	5		2441.54		52,070.96
52	6	60	50	5		3126.39		56,262.28
53	6	60	25	10		3265.08		25,415.23
54	6	60	50	10		3452.92		34,632.04
55	6	60	25	15		3509.18		24,369.98
56	6	60	50	15		2828.69		32,595.97
57	6	60	25	20		1975.11		57,051.02
58	6	60	50	20		2739.81		44,847.21
59	6	60	25	25		2646.33		23,934.19
60	6	60	50	25		3862.02		34,479.00
61	7	70	25	5		2724.59		125,310.30
62	7	70	50	5		3908.50		176,424.21
63	7	70	25	10		3302.91		94,039.44
64	7	70	50	10		5476.67		97,789.45
65	7	70	25	15		4687.39		72,966.97
66	7	70	50	15		3879.22		82,139.38
67	7	70	25	20		3911.55		151,041.96
68	7	70	50	20		3304.83		135,447.48
69	7	70	25	25		2907.61		78,552.63
70	7	70	50	25		3005.94		104,867.55
71	8	80	25	5		4214.92		480,711.82
72	8	80	50	5		5105.46		844,852.86
73	8	80	25	10		4350.86		408,832.66

ID	School	stop	cap	wd	P-HA			
					Exact solution	Metaheuristic	Exact (Gap)	Time (ms)
74	8	80	50	10		5671.75		446,486.49
75	8	80	25	15		6232.73		323,359.92
76	8	80	50	15		5147.59		396,422.09
77	8	80	25	20		5863.65		727,727.52
78	8	80	50	20		4529.16		637,288.66
79	8	80	25	25		4416.60		320,934.84
80	8	80	50	25		3731.95		487,288.66
81	9	90	25	5		5134.21		1,451,588.66
82	9	90	50	5		5938.65		1,949,344.91
83	9	90	25	10		5428.59		1,124,575.45
84	9	90	50	10		5859.72		1,317,101.91
85	9	90	25	15		5976.94		966,449.96
86	9	90	50	15		5544.09		1,095,494.84
87	9	90	25	20		5895.65		2,425,230.52
88	9	90	50	20		6147.85		2,254,604.73
89	9	90	25	25		5645.33		1,272,487.93
90	9	90	50	25		4584.54		1,812,233.96
91	10	100	25	5		5568.00		4,259,078.90
92	10	100	50	5		7133.91		7,143,027.31
93	10	100	25	10		6224.44		2,868,644.26
94	10	100	50	10		7006.72		4,284,048.45
95	10	100	25	15		7370.62		2,570,007.29
96	10	100	50	15		5860.76		2,817,816.07
97	10	100	25	20		6440.56		5,722,787.89
98	10	100	50	20		5563.31		5,520,410.63

ID	School	stop	cap	wd	P-HA			
					Exact solution	Metaheuristic	Exact (Gap)	Time (ms)
99	10	100	25	25		6037.21		3,303,408.12
100	10	100	50	25		6353.84		5,005,692.82

ID	School	stop	cap	wd	M-VND			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
1	1	10	25	5	170.64	1.16%	0.00%	248.40
2	1	10	50	5	325.34	2.77%	1.89%	675.99
3	1	10	25	10	201.41	3.14%	1.00%	599.39
4	1	10	50	10	270.13	1.90%	1.90%	352.12
5	1	10	25	15	281.13	0.00%	0.00%	542.68
6	1	10	50	15	219.21	0.43%	-2.00%	386.50
7	1	10	25	20	133.03	4.07%	1.58%	352.37
8	1	10	50	20	193.26	1.89%	1.89%	487.46
9	1	10	25	25	166.72	0.00%	-1.26%	357.72
10	1	10	50	25	90.32	4.85%	2.45%	824.88
11	2	20	25	5	201.64	0.00%	0.00%	1,069.55
12	2	20	50	5	370.33	6.33%	1.40%	1,461.97
13	2	20	25	10	288.70	4.11%	0.90%	1,797.99
14	2	20	50	10	386.54	5.87%	0.05%	2,226.67
15	2	20	25	15	311.28	2.67%	0.58%	2,429.82
16	2	20	50	15	325.73	5.02%	1.40%	2,532.54
17	2	20	25	20	341.30	7.60%	2.56%	1,837.82
18	2	20	50	20	230.92	6.40%	1.00%	2,402.80

ID	School	stop	cap	wd	M-VND			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
19	2	20	25	25	233.74		2.00%	1,725.79
20	2	20	50	25	268.87		2.30%	3,000.98
21	3	30	25	5	1018.35		3.00%	4,484.77
22	3	30	50	5	1269.99		1.20%	3,756.37
23	3	30	25	10	1244.23		2.30%	3,941.08
24	3	30	50	10	1735.25		6.70%	5,502.45
25	3	30	25	15	1579.68		2.30%	4,166.03
26	3	30	50	15	1428.22		1.90%	5,267.20
27	3	30	25	20	1067.97		-2.00%	4,649.58
28	3	30	50	20	1162.45		1.20%	4,882.85
29	3	30	25	25	1111.30		3.00%	3,897.77
30	3	30	50	25	416.08		2.30%	5,324.58
31	4	40	25	5	2282.83		3.00%	10,355.97
32	4	40	50	5	3734.71		3.21%	9,700.71
33	4	40	25	10	2479.28		2.30%	9,716.72
34	4	40	50	10	2354.13		2.30%	9,110.06
35	4	40	25	15	3260.75		6.70%	5,732.78
36	4	40	50	15	2725.98		6.00%	5,415.59
37	4	40	25	20	1511.68		2.10%	7,904.67
38	4	40	50	20	1859.28		8.90%	7,358.24
39	4	40	25	25	1839.61		4.80%	4,235.45
40	4	40	50	25	1235.68		7.20%	6,348.47
41	5	50	25	5	2393.32		2.58%	26,136.62
42	5	50	50	5	4667.50		3.65%	33,014.95
43	5	50	25	10	2583.11		3.98%	25,944.19

ID	School	stop	cap	wd	M-VND			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
44	5	50	50	10	3694.39		1.20%	22,384.11
45	5	50	25	15	4102.05		2.56%	19,874.24
46	5	50	50	15	3003.97		2.54%	23,663.18
47	5	50	25	20	1863.87		3.25%	38,200.11
48	5	50	50	20	2197.22		3.69%	31,119.58
49	5	50	25	25	2317.79		3.59%	14,618.79
50	5	50	50	25	2422.29		2.50%	21,027.15
51	6	60	25	5	2578.26		5.60%	92,423.12
52	6	60	50	5	3263.64		4.39%	98,920.74
53	6	60	25	10	3340.18		2.30%	39,799.07
54	6	60	50	10	3562.31		3.17%	55,761.28
55	6	60	25	15	3575.85		1.90%	43,237.99
56	6	60	50	15	2859.44		1.09%	55,913.45
57	6	60	25	20	2020.54		2.30%	98,683.62
58	6	60	50	20	2812.96		2.67%	73,555.03
59	6	60	25	25	2691.32		1.70%	42,462.32
60	6	60	50	25	3961.28		2.57%	60,652.01
61	7	70	25	5	2811.78		3.20%	213,009.19
62	7	70	50	5	3982.77		1.90%	328,799.25
63	7	70	25	10	3408.60		3.20%	166,854.67
64	7	70	50	10	5706.69		4.20%	180,794.83
65	7	70	25	15	4958.32		5.78%	121,861.24
66	7	70	50	15	3968.44		2.30%	144,536.52
67	7	70	25	20	4087.57		4.50%	252,239.05
68	7	70	50	20	3380.85		2.30%	230,219.55

ID	School	stop	cap	wd	M-VND			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
69	7	70	25	25	2954.14		1.60%	135,875.18
70	7	70	50	25	3078.08		2.40%	195,456.25
71	8	80	25	5	4349.80		3.20%	795,621.78
72	8	80	50	5	5161.62		1.10%	1,360,502.91
73	8	80	25	10	4550.12		4.58%	640,027.18
74	8	80	50	10	5793.69		2.15%	705,642.24
75	8	80	25	15	6310.64		1.25%	501,441.47
76	8	80	50	15	5258.05		2.15%	656,095.54
77	8	80	25	20	5746.38		-2.00%	1,052,554.60
78	8	80	50	20	4646.92		2.60%	969,365.45
79	8	80	25	25	4528.78		2.54%	464,149.32
80	8	80	50	25	3820.22		2.36%	806,550.99
81	9	90	25	5	5283.82		2.91%	2,424,068.65
82	9	90	50	5	6132.13		3.26%	2,964,723.48
83	9	90	25	10	5627.16		3.66%	1,777,427.94
84	9	90	50	10	6087.78		3.89%	2,238,857.38
85	9	90	25	15	6223.19		4.12%	1,469,879.60
86	9	90	50	15	5663.84		2.16%	1,682,469.44
87	9	90	25	20	6303.45		6.92%	3,760,843.58
88	9	90	50	20	6344.58		3.20%	3,563,433.94
89	9	90	25	25	5900.95		4.53%	1,840,467.01
90	9	90	50	25	4790.84		4.50%	2,567,097.36
91	10	100	25	5	6236.16		12.00%	6,609,813.87
92	10	100	50	5	7376.47		3.40%	10,324,797.60
93	10	100	25	10	6504.54		4.50%	4,451,933.59

ID	School	stop	cap	wd	M-VND			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
94	10	100	50	10	7216.92		3.00%	6,322,677.41
95	10	100	25	15	7812.86		6.00%	3,988,500.38
96	10	100	50	15	6095.19		4.00%	4,373,085.06
97	10	100	25	20	6646.66		3.20%	8,881,430.70
98	10	100	50	20	5680.14		2.10%	8,567,353.08
99	10	100	25	25	6308.89		4.50%	5,126,695.40
100	10	100	50	25	6569.87		3.40%	7,768,541.28

ID	School	stop	cap	wd	m-HA			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
1	1	10	25	5	170.6	1.16%	0.00%	427
2	1	10	50	5	330.3	4.34%	3.45%	1,144
3	1	10	25	10	204.3	4.63%	2.46%	773.40
4	1	10	50	10	270.0	1.87%	1.87%	508.90
5	1	10	25	15	281.1	0.00%	0.00%	791.13
6	1	10	50	15	222.6	2.00%	0.98%	562.14
7	1	10	25	20	133.4	4.37%	1.87%	557.67
8	1	10	50	20	198.7	4.74%	4.74%	704.32
9	1	10	25	25	182.2	9.28%	7.91%	560.49
10	1	10	50	25	94.3	9.43%	6.92%	1,136.17
11	2	20	25	5	204.6	1.48%	1.47%	1,660.19

ID	School	stop	cap	wd	m-HA			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
12	2	20	50	5	369.0	5.95%	1.04%	2,408.54
13	2	20	25	10	285.8	3.06%	-0.12%	2,780.39
14	2	20	50	10	392.8	7.57%	1.65%	3,079.08
15	2	20	25	15	317.0	4.55%	2.42%	3,334.19
16	2	20	50	15	329.5	6.24%	2.58%	3,680.65
17	2	20	25	20	331.7	4.57%	-0.33%	3,205.58
18	2	20	50	20	224.6	3.50%	-1.75%	3,490.60
19	2	20	25	25	234.0		2.10%	2,556.24
20	2	20	50	25	269.4		2.50%	4,061.63
21	3	30	25	5	1,043.0		5.49%	5,606.58
22	3	30	50	5	1,278.1		1.85%	5,303.66
23	3	30	25	10	1,248.1		2.62%	5,171.04
24	3	30	50	10	1,682.7		3.47%	7,321.38
25	3	30	25	15	1,569.5		1.64%	5,715.56
26	3	30	50	15	1,437.3		2.55%	7,726.14
27	3	30	25	20	1,106.9		1.57%	7,425.19
28	3	30	50	20	1,178.7		2.61%	6,905.96
29	3	30	25	25	1,098.8		1.84%	5,735.91
30	3	30	50	25	412.2		1.35%	7,453.61
31	4	40	25	5	2,196.1		-0.91%	14,825.82
32	4	40	50	5	3,743.6		3.45%	12,832.41
33	4	40	25	10	2,612.8		7.81%	12,447.46
34	4	40	50	10	2,374.1		3.17%	11,741.68
35	4	40	25	15	3,222.4		5.44%	6,592.46

ID	School	stop	cap	wd	m-HA			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
36	4	40	50	15	2,652.6		3.15%	8,611.97
37	4	40	25	20	1,519.1		2.60%	12,205.15
38	4	40	50	20	1,840.7		7.81%	11,557.93
39	4	40	25	25	1,819.5		3.65%	6,643.44
40	4	40	50	25	1,170.4		1.53%	9,909.02
41	5	50	25	5	2,420.4		3.74%	42,113.56
42	5	50	50	5	4,964.1		10.24%	45,234.91
43	5	50	25	10	2,675.6		7.70%	42,767.52
44	5	50	50	10	3,658.7		0.22%	35,065.68
45	5	50	25	15	4,223.6		5.60%	30,118.44
46	5	50	50	15	3,035.0		3.60%	35,775.81
47	5	50	25	20	1,893.7		4.90%	53,918.10
48	5	50	50	20	2,208.0		4.20%	45,955.61
49	5	50	25	25	2,304.0		2.97%	20,562.16
50	5	50	50	25	2,517.3		6.52%	25,672.93
51	6	60	25	5	2,524.6		3.40%	135,310.95
52	6	60	50	5	3,335.9		6.70%	120,188.73
53	6	60	25	10	3,413.4		4.54%	59,220.10
54	6	60	50	10	3,686.0		6.75%	79,459.40
55	6	60	25	15	3,697.7		5.37%	56,966.52
56	6	60	50	15	2,872.1		1.53%	78,821.24
57	6	60	25	20	2,026.7		2.61%	118,045.91
58	6	60	50	20	2,781.8		1.53%	107,916.65
59	6	60	25	25	2,713.1		2.52%	57,878.50

ID	School	stop	cap	wd	m-HA			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
60	6	60	50	25	3,947.7		2.22%	82,302.51
61	7	70	25	5	2,818.7		3.45%	279,629.47
62	7	70	50	5	3,966.2		1.47%	426,455.76
63	7	70	25	10	3,335.3		0.98%	214,496.06
64	7	70	50	10	5,427.3		-0.90%	229,454.64
65	7	70	25	15	4,547.7		-2.98%	166,272.89
66	7	70	50	15	3,978.7		2.56%	196,443.54
67	7	70	25	20	4,023.5		2.86%	344,044.65
68	7	70	50	20	3,386.0		2.46%	309,714.64
69	7	70	25	25	2,939.6		1.10%	176,473.57
70	7	70	50	25	3,090.1		2.80%	240,051.63
71	8	80	25	5	4,303.4		2.10%	1,051,591.48
72	8	80	50	5	5,338.8		4.57%	1,904,080.52
73	8	80	25	10	4,742.4		9.00%	931,334.22
74	8	80	50	10	5,615.0		-1.00%	967,831.90
75	8	80	25	15	6,376.1		2.30%	632,774.04
76	8	80	50	15	5,325.2		3.45%	897,733.30
77	8	80	25	20	5,998.5		2.30%	1,659,572.91
78	8	80	50	20	4,773.7		5.40%	1,365,846.77
79	8	80	25	25	4,509.3		2.10%	725,301.18
80	8	80	50	25	3,817.8		2.30%	1,164,062.75
81	9	90	25	5	5,365.2		4.50%	3,190,822.35
82	9	90	50	5	6,064.0		2.11%	4,439,637.56
83	9	90	25	10	5,734.7		5.64%	2,562,119.75

ID	School	stop	cap	wd	m-HA			
					Metaheuristic	Exact (Gap)	Heuristic(Gap)	Time (ms)
84	9	90	50	10	6,018.9		2.72%	3,045,570.07
85	9	90	25	15	6,344.6		6.15%	1,901,562.25
86	9	90	50	15	5,666.7		2.21%	2,378,089.83
87	9	90	25	20	6,172.1		4.69%	4,999,142.70
88	9	90	50	20	6,347.0		3.24%	4,871,710.48
89	9	90	25	25	5,790.4		2.57%	2,722,695.01
90	9	90	50	25	4,666.6		1.79%	4,044,359.76
91	10	100	25	5	5,929.9		6.50%	9,385,382.65
92	10	100	50	5	7,611.9		6.70%	10,583,988.42
93	10	100	25	10	6,439.2		3.45%	5,900,888.41
94	10	100	50	10	7,348.6		4.88%	9,117,974.90
94	10	100	25	15	7,502.5		1.79%	5,886,409.78
96	10	100	50	15	6,226.4		6.24%	6,192,687.34
96	10	100	25	20	6,877.3		6.78%	9,436,397.53
96	10	100	50	20	5,988.5		7.64%	9,114,426.88
96	10	100	25	25	6,532.7		8.21%	7,259,868.32
96	10	100	50	25	6,508.6		2.44%	8,203,618.11

ID	School	stop	cap	wd	P-VND			
					Metaheuristic	Exact (Gap)	Heuristic (Gap)	Time (ms)
1	1	10	25	5	174.7	3.55%	2.37%	155.93
2	1	10	50	5	340.8	7.64%	6.72%	412.93
3	1	10	25	10	197.4	1.09%	-1.00%	295.22
4	1	10	50	10	268.5	1.28%	1.28%	182.16
5	1	10	25	15	281.1	0.00%	0.00%	267.10
6	1	10	50	15	233.6	7.00%	5.93%	183.15
7	1	10	25	20	134.6	5.28%	2.76%	224.04
8	1	10	50	20	192.2	1.32%	1.32%	246.60
9	1	10	25	25	180.2	8.08%	6.72%	227.80
10	1	10	50	25	90.3	4.87%	2.46%	371.41
11	2	20	25	5	203.4	0.88%	0.88%	624.23
12	2	20	50	5	374.6	7.55%	2.56%	883.46
13	2	20	25	10	299.4	7.98%	4.64%	1,008.63
14	2	20	50	10	395.9	8.43%	2.46%	1,095.22
15	2	20	25	15	314.0	3.59%	1.47%	1,196.91
16	2	20	50	15	325.3	4.89%	1.28%	1,311.67
17	2	20	25	20	343.1	8.16%	3.10%	1,202.26
18	2	20	50	20	237.1	9.23%	3.69%	1,259.28
19	2	20	25	25	237.1		3.45%	995.66
20	2	20	50	25	268.4		2.11%	1,505.28
21	3	30	25	5	1,033.9		4.58%	2,092.18
22	3	30	50	5	1,300.7		3.65%	1,834.35
23	3	30	25	10	1,264.0		3.93%	2,132.30
24	3	30	50	10	1,662.4		2.22%	2,778.46

ID	School	stop	cap	wd	P-VND			
					Metaheuristic	Exact (Gap)	Heuristic (Gap)	Time (ms)
25	3	30	25	15	1,615.9		4.65%	2,118.00
26	3	30	50	15	1,445.5		3.13%	2,764.83
27	3	30	25	20	1,140.4		4.65%	2,620.44
28	3	30	50	20	1,210.2		5.35%	2,439.05
29	3	30	25	25	1,129.1		4.65%	1,987.19
30	3	30	50	25	427.2		5.03%	3,129.72
31	4	40	25	5	2,293.6		3.48%	6,089.49
32	4	40	50	5	3,773.9		4.29%	4,311.45
33	4	40	25	10	2,635.8		8.76%	4,997.81
34	4	40	50	10	2,455.4		6.70%	4,975.74
35	4	40	25	15	3,288.2		7.60%	3,084.44
36	4	40	50	15	2,750.4		6.95%	3,544.09
37	4	40	25	20	1,532.4		3.50%	4,753.68
38	4	40	50	20	1,857.1		8.77%	4,487.85
39	4	40	25	25	1,799.2		2.50%	2,608.42
40	4	40	50	25	1,198.8		4.00%	3,766.76
41	5	50	25	5	2,416.2		3.56%	14,929.68
42	5	50	50	5	4,872.4		8.20%	18,178.93
43	5	50	25	10	2,586.1		4.10%	16,446.73
44	5	50	50	10	3,744.8		2.58%	14,689.89
45	5	50	25	15	4,147.6		3.70%	10,687.12
46	5	50	50	15	3,099.5		5.80%	12,262.55
47	5	50	25	20	1,871.1		3.65%	20,014.85
48	5	50	50	20	2,145.7		1.26%	17,186.92

ID	School	stop	cap	wd	P-VND			
					Metaheuristic	Exact (Gap)	Heuristic (Gap)	Time (ms)
49	5	50	25	25	2,291.2		2.40%	8,447.63
50	5	50	50	25	2,467.2		4.40%	10,824.89
51	6	60	25	5	2,632.0		7.80%	49,293.30
52	6	60	50	5	3,235.8		3.50%	56,234.04
53	6	60	25	10	3,384.3		3.65%	25,873.08
54	6	60	50	10	3,610.4		4.56%	34,942.56
55	6	60	25	15	3,785.0		7.86%	24,804.81
56	6	60	50	15	3,023.0		6.87%	30,564.74
57	6	60	25	20	2,082.0		5.41%	49,419.27
58	6	60	50	20	2,918.4		6.52%	39,235.84
59	6	60	25	25	2,818.3		6.50%	20,724.69
60	6	60	50	25	3,960.1		2.54%	29,239.31
61	7	70	25	5	2,825.1		3.69%	105,185.74
62	7	70	50	5	4,087.9		4.59%	152,773.23
63	7	70	25	10	3,388.8		2.60%	88,160.46
64	7	70	50	10	5,903.9		7.80%	91,677.61
65	7	70	25	15	4,952.2		5.65%	61,893.06
66	7	70	50	15	4,013.0		3.45%	74,811.58
67	7	70	25	20	3,833.3		-2.00%	151,049.02
68	7	70	50	20	3,400.7		2.90%	139,072.99
69	7	70	25	25	2,981.5		2.54%	73,643.92
70	7	70	50	25	3,115.7		3.65%	102,058.93
71	8	80	25	5	4,388.6		4.12%	484,993.29
72	8	80	50	5	5,340.3		4.60%	792,036.17

ID	School	stop	cap	wd	P-VND			
					Metaheuristic	Exact (Gap)	Heuristic (Gap)	Time (ms)
73	8	80	25	10	4,503.0		3.50%	401,523.27
74	8	80	50	10	5,930.9		4.57%	446,486.49
75	8	80	25	15	6,522.6		4.65%	326,244.70
76	8	80	50	15	5,441.0		5.70%	371,629.17
77	8	80	25	20	6,268.2		6.90%	688,725.71
78	8	80	50	20	4,583.5		1.20%	603,227.47
79	8	80	25	25	4,840.6		9.60%	297,993.49
80	8	80	50	25	3,825.3		2.50%	456,833.12
81	9	90	25	5	5,323.7		3.69%	1,451,557.07
82	9	90	50	5	5,793.7		-2.44%	1,966,736.46
83	9	90	25	10	5,589.3		2.96%	1,084,416.49
84	9	90	50	10	6,076.4		3.70%	1,234,787.86
85	9	90	25	15	6,245.9		4.50%	975,083.70
86	9	90	50	15	5,904.5		6.50%	1,115,031.31
87	9	90	25	20	6,150.6		4.33%	2,446,899.50
88	9	90	50	20	6,455.2		5.00%	2,214,308.74
89	9	90	25	25	6,198.6		9.80%	1,102,068.73
90	9	90	50	25	4,680.8		2.10%	1,569,549.02
91	10	100	25	5	6,111.4		9.76%	3,569,920.19
92	10	100	50	5	7,340.1		2.89%	6,052,282.12
93	10	100	25	10	6,822.0		9.60%	2,770,381.27
94	10	100	50	10	7,139.8		1.90%	4,176,337.76
95	10	100	25	15	7,554.9		2.50%	2,552,211.37
96	10	100	50	15	6,065.9		3.50%	2,798,304.21

ID	School	stop	cap	wd	P-VND			
					Metaheuristic	Exact (Gap)	Heuristic (Gap)	Time (ms)
97	10	100	25	20	6,730.4		4.50%	5,683,160.65
98	10	100	50	20	5,874.9		5.60%	5,482,184.74
99	10	100	25	25	6,564.9		8.74%	3,280,533.79
100	10	100	50	25	6,643.6		4.56%	4,971,031.08

ANT/OR - University of Antwerp Operations Research Group
Faculty of Applied Economic

Metaheuristics for school bus routing problem

Thesis submitted for the Degree of Doctor in Department of Engineering Management at the University of Antwerp to be defended by **Mohammad Saied Fallah Niasar**

Supervisors:
Prof. Kenneth Sörensen
Prof. dr. Mehdi Sajadifar

March 2024