International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 58, No. 2, February, 2025, pp. 191-203
Journal homepage: http://iieta.org/journals/jesa

A Hybrid Memetic and Set Partitioning Optimization Framework for Decision Support in Industrial Transportation: A Case Study of Employee Shuttle Routing

Hajar Bideq[1*], Khaoula Ouaddi[2], Rachid Ellaia[1]

[1] LERMA Laboratory, Mohammadia School of Engineers, Mohammed V University, Rabat 10000, Morocco
[2] Center for Doctoral Studies in Information Technology and Engineering Sciences, National Higher School for Computer Science and Systems Analysis, Mohammed V University, Rabat 10000, Morocco

Corresponding Author Email: hajar.bideq@gmail.com

https://doi.org/10.18280/jesa.580201

Received: 8 December 2024
Revised: 10 February 2025
Accepted: 19 February 2025
Available online: 28 February 2025

Keywords:
employee bus routing problem, matheuristic, Memetic Algorithm (MA), mixed loads, set partitioning

ABSTRACT

Designing cost-effective shuttle services for large-scale industrial companies presents a significant challenge in the transportation industry. This challenge arises from the need to balance high-quality service with cost-effectiveness while considering various practical constraints. In this context, we introduce a novel approach to help decision-makers address Employee Shuttle Bus Routing Problems (ESBRP). Our method combines the Memetic Algorithm (MA), a metaheuristic, with the Set Partitioning Problem (SPP) model, an exact algorithm. The proposed framework consists of two phases: (1) generating routes that adhere to the real-world constraints of the ESBRP using the MA, and (2) allocating these routes to a heterogeneous fleet of vehicles by optimally solving the SPP Model. A unique feature of our approach is the extension of the framework to enable the transition from addressing the single-load scenario of the ESBRP problem to solving the mixed-load scenario. This transition is achieved by implementing the Single to Mixed Loads Heuristic (SMH). This paper presents the results of thorough computational tests conducted on multiple data instances of varying sizes. Additionally, we develop a mixed-integer programming (MIP) model for the ESBRP to compare and evaluate the results of the proposed framework. By assessing solution quality and execution times on small and moderate-sized data instances, the experiments demonstrate that the proposed approach is efficient and often generates near-optimal solutions.

1. INTRODUCTION

Within the current economic context, the role of transportation systems emerges as decisive in the daily life of industrial societies, notably affecting their employees' mobility. Businesses encounter the issue of developing and implementing a reliable transportation system to carry employees to and from their workplaces, daily. In particular, using conventional methods to provide shuttle services that balance cost-effectiveness and employee satisfaction remains challenging and unattainable. Hence, designing reliable algorithms that ensure the generation of the most optimal route planning is needed.

In general, this class of problem is one of the Vehicle Routing Problem (VRP) variants. VRP is a well-studied NP-hard combinatorial problem aiming to design the "best" routes for a fleet of vehicles to serve a specific group of clients as demonstrated by Kumar and Panneerselvam [1] and Luo and Fu [2]. This problem finds applications in numerous industries and has many variants, such as bus school routing, public transportation scheduling, and split delivery VRP [3].

This paper addresses the problem of employee transportation for a large industrial group. Our goal is to plan the most efficient routes for a heterogeneous fleet of buses to pick up employees from their respective bus stops and transport them to their workplaces, all while minimizing total costs. A route planning in our context involves generating routes, assigning them to buses, and determining which employees to pick up on each route/bus. The planned routes should satisfy several constraints such as seat capacity and arrival time requirements, while minimizing total cost and distance. This problem is referred to in the literature as the Employee Shuttle Bus Routing Problem (ESBRP) [4, 5].

The ESBRP is classified as a School Bus Routing Problem (SBRP) [6]. SBRP comprises five sub-problems: data preparation, route scheduling, route generation, selection of bus stops, and adjustment of school bell time adjustment. For this study, we focus on generating and scheduling the routes based on predetermined arrival times. The data collection and selection of bus stops (pickup points) are managed by our data provider, a large Moroccan industrial group. The school bell time adjustment is not relevant to our current problem.

Moreover, all buses start by leaving the depot and then visit at least one bus stop before proceeding to the workplace(s). Unlike a typical VRP, the buses in our problem do not necessarily return to the depot during their journey since they can have multiple tours throughout the day. However, for this research paper, we only permit single tours per vehicle, with each tour beginning at the depot and ending at a workplace. The routes whereby employees are transported from workplaces to bus stops are generated by reversing the bus stops and workplaces.

Additionally, the ESBRP we are addressing includes multiple workplaces. This presents two options: single load and mixed load. A single-load ESBRP means each bus can only transport employees from the same workplace. On the other hand, a mixed-load ESBRP allows a bus to transport employees from different workplaces. Adopting mixed load restrictions would be more advantageous by default. However, due to the social policies and regulations of the industrial company, certain groups of employees from different workplaces may not be allowed to be simultaneously transported on the same bus. In contrast, others may have this permission. This is where versatile load planning becomes essential. However, the ESBRP with versatile load planning is further complicated by the constraint on arrival time.

Therefore, we proceed in two phases to resolve the ESBRP. In the first phase, we implement a MA to find the "best" routes in terms of minimum total distance for a single-load ESBRP with time arrival restrictions. The second phase involves passing the best solutions obtained by the MA through a Set Partitioning Problem (SPP) model, which aims to assign these routes to a fleet of heterogeneous buses while minimizing the total cost. To measure the generated solutions' quality, we have developed a mixed integer programming model (MIP) for the single-load ESBRP with time restrictions. Since ESBRP is an NP-hard problem, we could not test our MIP on large-size instances. However, the proposed matheuristic provided a fast and accurate approximation of the optimal solution for small and medium-sized instances. Finally, to obtain mixed-load solutions, we have extended our framework by developing a heuristic that seamlessly allows for this transition.

While the idea behind the proposed matheuristic is not entirely new, no studies have been conducted on integrating MA and SPP to solve the ESBRP with time constraints. For instance, Alvarenga et al. [7] have used a GA and SPP-based matheuristic to solve a VRP with time windows, but our current proposal introduces novel contributions. In the previously mentioned work [7], only one objective was considered. However, in our case, we have two complementary objectives. Firstly, we focus on reducing the total distance then minimizing the total cost. It is important to note that minimizing the total distance does not guarantee minimizing the total cost. This is because we have a heterogeneous fleet of buses with varying capacities and usage costs. If we randomly assign routes with the minimum total distance to buses, we may favor expensive buses. Therefore, the SPP model in this paper is not simply used for minor improvements but rather to make the best decision in terms of vehicle assignment. Additionally, it is widely acknowledged that parameterizing metaheuristics, such as MA, can be a challenging task. The solution quality and the time convergence time are directly influenced by the parameterization. Even with a suboptimal parameterization in the first phase of the MSP framework (the MA), we achieve superior solutions in terms of total costs and execution time compared to a basic MA. This improvement is due to enhancements made by the second phase of our proposed MSP framework, the SPP. Also, since this study is based on a real-world problem, the solution approaches presented in this paper can serve as a useful guide for tackling similar practical issues.

The rest of the paper is as follows: Section 2 presents the relevant literature on the ESBRP with time constraints. Section 3 provides a comprehensive explanation of our ESBRP problem. Section 4 presents the mathematical model formulated for the ESBRP with arrival time restriction. Section 5 describes the matheuristic we propose to solve the ESBRP. Section 6 presents the results we obtained from conducting various tests and provides an overview of the data sets used. Finally, in Section 7, we present our concluding remarks, as well as the challenges and opportunities for future work.

2. RELEVANT LITERATURE

The ESBRP has undergone significant changes over the years to reflect the increasing complexity of transportation needs and the growing demand for efficient employee transportation solutions. Initially, the problem was closely associated with the SBRP, as both involve scheduling and routing of vehicles to pick up individuals from different locations. School bus routing originated in the 1960s [8], with initial studies focusing on improving student transportation. Subsequently, employee bus routing gained importance as companies sought to streamline their employees' commuting efficiency. However, the literature often combines information from both domains due to limited research exclusively dedicated to employee bus routing. The similarities in problem structure and objectives allow for the application of methodologies and the acquisition of valuable insights across both areas.

As mentioned earlier, ESBRP is similar to SBRP. When it comes to the algorithms proposed for SBRP, a wide range of exact approaches, heuristics, and metaheuristics were explored. Various research studies in the SBRP field discuss the use of exact methods such as integer programming models (MIP), cutting plane techniques, and column generation procedures. For example, Li et al. [9] solved delivery problems using a branch-and-price-and-cut technique. Schittekat et al. [10] implemented an integer programming model for problems such as route generation and bus stop selection. Kumar and Jain [11] utilized a branch and bound algorithm to solve an assignment-based bus route generation problem.

Even though these methods can solve routing problems optimally, the size and complexity of the SBRP significantly increase execution times. This complexity justifies researchers' preference for heuristics and metaheuristics to streamline the search for high-quality solutions. The most commonly used ones are Ant Colony Optimization (ACO) [12, 13], Greedy Random Adaptive Search Procedure (GRASP) [10, 14, 15], and Genetic Algorithm (GA). Sghaier et al. [16] used a GA to solve a case study with capacity and riding time constraints. Oluwadare et al. [17] implemented a GA in a Nigerian school district to minimize the number of used buses. Díaz-Parra et al. [18] combined GA with a k-means algorithm to solve a bus route generation problem with 200 stops. Chalkia et al. [19] designed a GA to ensure the selection of safe routes for a capacitated SBRP with maximum riding time. For the multiple applications of the SBRP, the reviews of Ellegood et al. [20] and Park and Kim [21] are highly recommended.

Similarly, the ESBRP focuses on providing efficient commuting services for employees within an industrial context, often considering challenging factors such as remote locations and work hours. One interesting approach in ESBRP was developed by Hart et al. [22], which involved the creation of a hyper-heuristic method for handling scheduling and routing problems. Leksakul et al. [23] implemented Machine learning-based algorithms such as K-means and Competitive Learning to solve real instances of ESBRP. Purba et al. [24] compared their Tabu Search-based approach with previous methods used by the company. For readers interested in the ESBRP, the review by Peker and Eliiyi [25], which maps most of the interesting papers published in the last decade, is recommended.

In Section 3, we thoroughly describe the ESBRP with time constraints.

3. PROBLEM DESCRIPTION

In this section, we provide a comprehensive overview of the ESBRP with time constraints. The ESBRP can be described as follows: Buses must navigate city streets, transport employees from designated bus stops to their respective workplaces, such as offices or factories, all while adhering to strict time and capacity limits. When several employees are gathered at the same stop and must arrive at their workplace at the same time, they are referred to as a "group of employees". These groups are characterized by three key elements: the workplace destination, the arrival time, and the number of people within the group. Moreover, a departure depot serves as the starting point for buses' daily journeys. After leaving the depot, the buses visit the planned bus stops before making their way to the workplace(s).

As previously mentioned, there are various workplaces within the same industrial group. Due to social concerns from labor unions and complaints from employees, the company occasionally needs to refrain from transporting groups of employees who work in different locations on the same bus. Therefore, the industrial group must have the flexibility to decide whether to allow or prohibit the transportation of these groups based on its needs. If the company permits groups of employees from the same workplace to share a bus, then the scenario we address is the ESBRP with mixed loads. However, if this is not allowed, then we are dealing with the ESBRP with a single load.

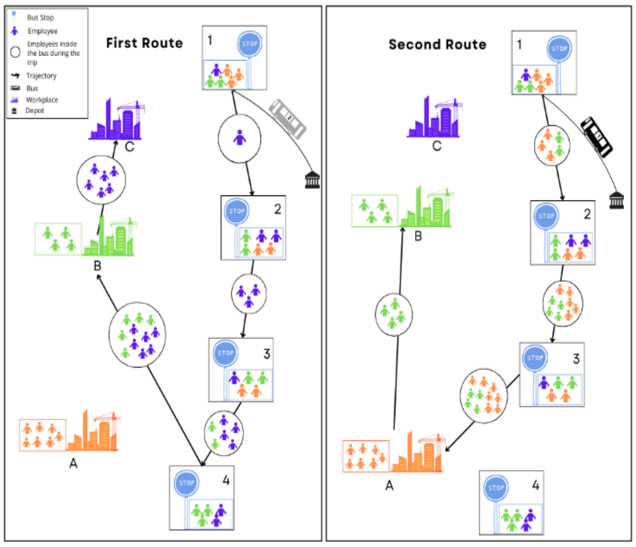

Figure 1. ESBRP with mixed loads

In Figure 1, we present an example of the ESBRP with mixed loads scenario. The diagram illustrates the network of bus stops, the workplaces, and the groups of employees. Although bus1 can pick up all the employee groups, we had to create two routes. This is because a single route would not be feasible.

In contrast, Figure 2 presents the scenario of the ESBRP with single load. In this scenario, buses only carry groups of employees with the same workplace destination and arrival time.

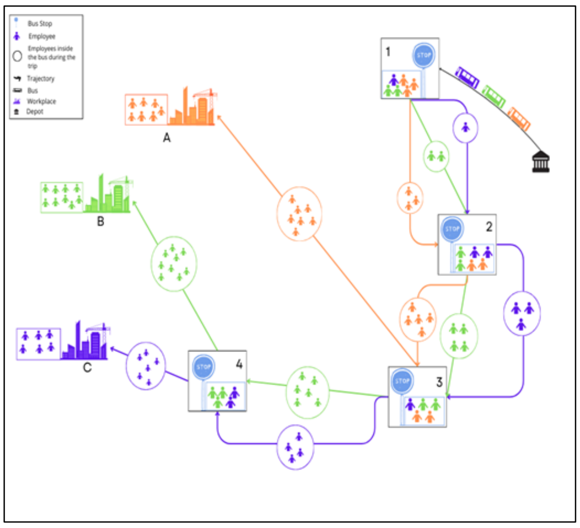

Figure 2. ESBRP with single loads

After describing the problem, we are addressing in this paper, the subsequent section provides a mathematical formulation of the ESBRP with arrival time constraints.

4. MATHEMATICAL FORMULATION

We formulate the ESBRP with arrival time constraints by adapting the model from our previous work [26] to incorporate and address the time requirements. We present this model in this section.

4.1 Indices

i and j: represent nodes that are either depots, bus stops, or workplaces.

t: represents an arrival time.

k: represents a bus.

(m,n,t): employees waiting at stop m that should be transported from m to n, and who need to arrive to n at time t.

0: represents the depot.

4.2 Data parameters

p: total stops.

w: total work sites.

b: total vehicles.

s: total timeslots.

q_k: capacity of a bus k.

c_k: using cost of a bus k.

d_{ij}: distance between i and j.

e_{mnt}: the number of employees number waiting at stop m heading to workplace n with an arrival time of t.

4.3 Data collections

$P = \{1, 2, 3, 4, ..., p\}$ selection of stops.
$W = \{p+1, p+2, p+3, ..., p+w\}$ selection of work sites.
$N = \{0, 1, 2, ..., p, p+1, p+2, .., p+w\}$ collection of nodes beginning with the depot, then stops and work sites.
$K = \{1, 2, 3, 4, ..., b\}$ selection of buses.
$T = \{1, 2, ..., s\}$ selection of times.
E: selection of employees groups (m,n,t), $m \in P$, $n \in W$, $t \in T$.

4.4 Variables

$x_{ijk} = 1$ if a bus k goes from i to j, and 0 if not.
$y_{mntk} = 1$ if a bus k transports (m,n,t), and 0 if not.
v_{ik}: it determines the visiting order of i by the bus k.

4.5 Objective function

Min: $\sum_{i \in N} \sum_{j \in N-\{i\}} \sum_{k \in K} x_{ijk} \times c_k \times d_{ij}$
In this paper, we aim to reduce total operational costs by calculating the cost per kilometer for each bus in operation.

4.6 Constraints

This model adheres to the constraints below:

$$\sum_{(m,n,t) \in E} y_{mntk} \times e_{mnt} \le q_k \quad \forall k \in K \tag{1}$$

$$\sum_{k \in K} y_{mntk} = 1 \quad \forall (m,n,t) \in E \tag{2}$$

$$y_{mntk} \le \sum_{j \in N-\{0,m\}} x_{mjk} \; \forall k \in K \quad \forall (m,n,t) \in E \tag{3}$$

$$y_{mntk} \le \sum_{i \in (P\cup\{0\})-\{m\}} x_{imk}, \forall k \in K, \forall (m,n,t) \in E \tag{4}$$

$$y_{mntk} \le \sum_{i \in N-\{0,n\}} x_{ink}, \forall k \in K, \forall (m,n,t) \in E \tag{5}$$

$$x_{0jk} = \sum_{r \in N-\{0,j\}} x_{jrk}, \forall j \in P, \forall k \in K \tag{6}$$

$$\sum_{i \in W} \sum_{j \in W-\{i\}} x_{ijk} = 0 \; \forall k \in K \tag{7}$$

$$v_{jk} \ge v_{ik} + 1 - card(N) \times (1 - x_{ijk}) \; \forall (i,j) \in A \quad \forall k \in K \tag{8}$$

$$\sum_{j \in N-\{0,i\}} x_{ijk} \le 1 \quad \forall i \in N - \{0\} \, \forall k \in K \tag{9}$$

$$\sum_{i \in N-\{j\}} x_{ijk} \le 1 \quad \forall \text{j} \in N - \{0\} \, \forall k \in K \tag{10}$$

$$x_{ijk} = \sum_{r \in (P\cup\{0\})-\{i\}} x_{rik}, \forall k \in K, \forall i \in P, \forall j \in N - \{0, i\} \tag{11}$$

$$x_{ijk} = \sum_{r \in P-\{i\}} x_{rik} \; \forall k \in K, \forall i \in P, \forall j \in W - \{i\} \tag{12}$$

$$\sum_{j \in N-\{0\}} x_{0jk} \le 1 \quad \forall k \in K \tag{13}$$

$$\sum_{j \in N-\{0\}} x_{mjk} \le \sum_{n \in W} y_{mntk} \; \forall k \in K, \forall m \in P, \forall t \in T \tag{14}$$

$$\sum_{j \in P\cup\{0\}} x_{jmk} \le \sum_{n \in W} y_{mntk} \; \forall k \in K, \forall m \in P, \forall t \in T \tag{15}$$

$$\sum_{i \in N-\{0\}} x_{ink} \le \sum_{m \in P} y_{mntk} \quad \forall k \in K \; \forall n \in W \; \forall t \in T \tag{16}$$

$$\sum_{i \in W} x_{nik} \le \sum_{m \in P} y_{mntk} \quad \forall k \in K \; \forall n \in W \; \forall t \in T \tag{17}$$

$$\sum_{i \in N} x_{ijk} = \sum_{r \in N} x_{jrk} \quad \forall k \in K \; \forall j \in N \tag{18}$$

$$\sum_{m \in P} \sum_{n \in W} \sum_{t \in T} y_{mntk} \times \sum_{m \in P} \sum_{n \in W} y_{mnfk} = 0 \quad \forall k \in K, \quad \forall f \in T - \{t\} \tag{19}$$

Eq. (1) defines the capacity requirement for each vehicle k. Eq. (2) ensures that all employee groups are picked up, with each group being picked up only once by a single vehicle. Eqs. (3), (4), and (5) stipulate that if a bus k transports (m,n,t), it must visit both m and n then leave them. If a bus k visits the arc (0,j) then vehicle k should leave the node j (Eq. (6)). Eq. (7) restricts vehicle k from going from one workplace to another workplace. Eq. (8) introduces the concept of visit order to the model to prevent any potential sub-tours.

Eqs. (9) and (10) specify that a vehicle k can visit a node i either once or not at all. If a bus k goes from i to j and i refers to a bus stop, it means that the previous node visited by the vehicle k should be a bus stop or the depot (Eq. (11)). Eq. (12) is to make sure that if a vehicle visits the arc (i,j) with i a workplace then the previous node visited by the vehicle should be a workplace or a bus stop.

No multi-tours are allowed; only one route is assigned to a vehicle (Eq. (13)). If a bus does not transport a group (m,n,t), it is forbidden to go to m (Eqs. (14) and (15)). Furthermore, if a bus does not pick up a group (m,n,t) and has not picked up any other group going to n, then the vehicle k should not visit n (Eqs. (16) and (17)). Eq. (18) is for the flow conservation.

Finally, if a bus k transports (m,n,t), it is not allowed to pick up any group with the same workplace and different arrival times (Eq. (19)).

The following section presents the proposed MSP framework.

5. THE PROPOSED MATHEURISTIC

To address the ESBRP with arrival time constraints, we propose a hybrid approach called MSP. MSP hybridizes a MA with a SPP formulation, considering capacity, time, and single/mixed loads. Although evolutionary-based metaheuristic algorithms are effective in optimizing complex combinatorial problems like the ESBRP, they have limitations. A significant concern is early convergence, where the algorithm settles on suboptimal solutions too soon in the search process. This can occur due to inadequate diversity maintenance, improper selection methods, or incorrect parameter settings. Similarly, while mathematical programming approaches can help find optimal solutions, they also have drawbacks. The computational complexity of solving large-scale instances with mathematical programming techniques can lead to long execution times and potentially infeasible solutions. Therefore, this paper suggests leveraging the benefits of mathematical programming to achieve optimal solutions while harnessing the efficacy and adaptability of metaheuristic algorithms.

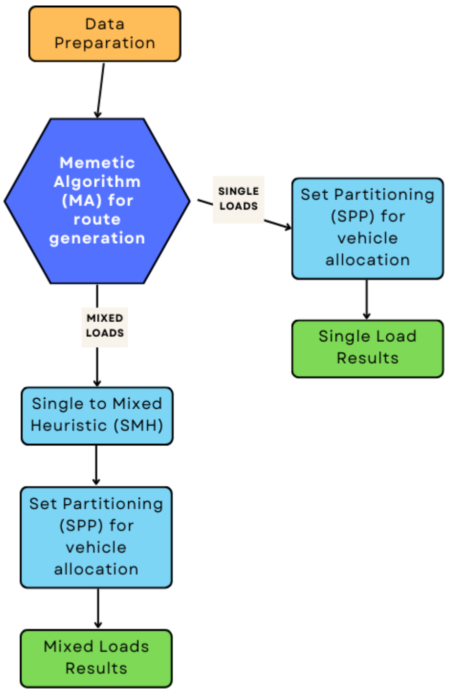
Data Preparation
Memetic Algorithm (MA) for route generation
SINGLE LOADS
Set Partitioning (SPP) for vehicle allocation
Single Load Results
MIXED LOADS
Single to Mixed Heuristic (SMH)
Set Partitioning (SPP) for vehicle allocation
Mixed Loads Results

Figure 3. The MSP framework structure

As shown in Figure 3, the MSP framework we use to solve the ESBRP with arrival time constraints consists of three phases:

(1) The main objective of the initial phase is to generate routes that adhere to the real-life constraints of the ESBRP while minimizing the total distance. We obtain these routes using an efficient MA, which is a hybridization of a GA and Local Search operators [27]. Further details regarding the MA are provided later in this section.

(2) The second phase involves assigning the generated routes to a fleet of heterogeneous buses using a SPP-based formulation (SPP). Unlike the first phase, the goal here is to minimize the total cost. Minimizing the total travel distance does not necessarily reduce the cost since the vehicles fleet is heterogeneous. Detailed information concerning this phase is presented later in this section.

(3) The third step is to expand the MSP framework to incorporate mixed loads by developing an efficient Single to Mixed loads Heuristic (SMH). Further details regarding this heuristic are provided later in this section.

The following subsections present the three steps of the MSP framework.

5.1 Memetic Algorithm

A MA is a hybrid optimization technique that integrates evolutionary algorithms with local search methods to improve solution quality and convergence quality. In this paper, we combine a Genetic Algorithm (GA) with a Local Search method (LS), which we refer to as MA.

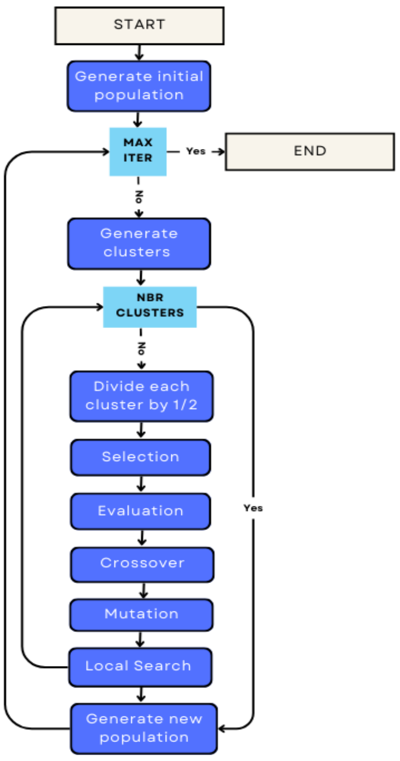

Figure 4. The first phase of the MSP framework - the MA

This section presents the initial step of our MSP framework. The primary goal is to generate solutions that minimize total travel distance while adhering to capacity and time constraints. In this stage, we only consider the single-load scenario. Initially, we implemented a traditional MA and conducted tests. Still, we quickly discovered that conventional selection operators, which we will discuss later, led to premature convergence as the algorithm settled into local optima in the first few iterations. Therefore, we developed a customized MA, drawing inspiration from the work of Malik and Wadhwa [28]. As shown in Figure 4, this personalized MA operates as follows:

- Generate the initial population of candidate solutions (individuals). The initial population is generated using an advanced heuristic that we explain later in this paper.
- Divide the population into clusters of equal size. The number of clusters is calculated using log2(n) and n is the population size.
- Split each cluster into two equal parts.
- Select the highest-performing chromosomes from the first

and second halves of each cluster.

• Perform crossover between the selected chromosomes to generate two child chromosomes, based on the crossover probability.

• Select the best chromosome from the first half, second half, and child chromosomes to retain it in the next generation.

• Apply LS to the best chromosome and child chromosomes to improve their quality and incorporate them into the next generation.

• Perform mutation on the child's chromosomes, depending on the mutation probability, and add them to the next generation.

• Apply LS to the result of the mutation and add it to the next generation.

• Repeat steps 2 to 9 for a determined number of iterations.

The following subsections present the different operators of the MA.

5.1.1 Chromosome encoding

Before discussing our encoding approach, it is essential to note that in this paper:

• A chromosome represents a complete and feasible solution.

• A population is a collection of chromosomes.

• A chromosome comprises a number of mini-chromosomes. Each mini-chromosome is a route which is assigned to a bus.

• A viable and complete solution must meet all constraints and ensure that all groups of employees are picked up and dropped off as required.

As illustrated in Figure 5, a chromosome consists of v mini-chromosomes, where v is the number of available buses. Each mini-chromosome represents the route assigned to a bus. The representation of the solution's elements is real-coded and not binary to generate chromosomes that efficiently reflect the essential information deciders need. The components of the chromosomes are encoded as follows:

• A bus stop is assigned a unique identifier i, ranging from 1 to p, where p is the total number of bus stops.

• A workplace is assigned a unique integer identifier j, ranging from p+1 to $p + w$, and w represents the number of workplaces.

• The five different types of buses are from type1 to type5, and each bus is assigned a unique identifier, k, ranging from 1 to v, and v represents the number of vehicles.

• The depot is represented by 0.

• A group of employees' representation is as follows: a tetrad (i, j, t, e), and i is the index of the bus stop where the group is waiting, j is the index of the workplace, t is the group's required arrival time at their workplace, and e represents the number of employees in that group.

Furthermore, as shown in Figure 5, the encoding of each mini-chromosome consists of three layers:

• A sequence of integers representing the bus stops and workplaces in the correct order of their visit, with the corresponding arrival time.

• The index of the bus/vehicle associated with the mini-chromosome.

The groups of employees assigned to that mini-chromosome/route.

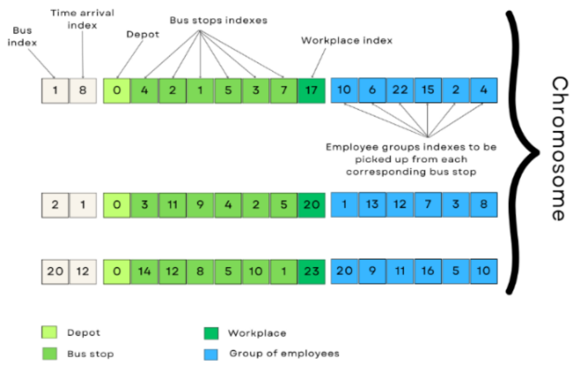

Figure 5. The chromosome encoding of the proposed MA

5.1.2 Initial population

Algorithm 1. Creating initial population

```
Procedure CREATE CHROMOSOME
  Groups: groups of employees
  Mini-chromosome: route of a vehicle
  Used groups: groups already picked up
  Used stops: groups at stops already visited by a vehicle
  Initialize capacity chromosome with 0
  Initialize a chromosome with an empty list
  Randomize the list of vehicles
  For vehicle in list of vehicles do
    Initialize a mini-chromosome with an empty list
    Initialize capacity mini-chromosome with 0
    While capacity mini-chromosome < capacity vehicle do
      M = Groups excluding the Used groups and Used stops
      If M is empty do
        break
      End if
Choose a random group g from Groups
Update capacity mini-chromosome with number of
people in group g
      If capacity mini-chromosome < capacity vehicle do
Add group g to mini-chromosome
Add group g to the Used groups
Add groups with time and destination different than group
g to Used stops
      End if
      Empty Used groups
      Add all picked up groups to Used groups
      Add the total number of people picked up so far to
capacity chromosome
      Add mini-chromosome to chromosome
    End while
  End for
  Return chromosome
End Procedure
Repeat Procedure until the desired size of population is
reached
```

In evolutionary metaheuristics, the initial population is typically generated randomly or through established heuristic methods. In vehicle routing problems (VRP), researchers employ various common heuristics to create initial populations, such as the "Push Forward Insertion Heuristic" (PFIH) introduced by Solomon [29]. However, due to the intricate nature and expansive solution space of our ESBRP, most of these well-known heuristics are unsuitable for our specific problem. Consequently, we have developed a customized heuristic to generate the initial population for our MA. This heuristic primarily relies on efficient insertion mechanisms that consider the diverse constraints of our ESBRP while incorporating a level of randomness during route generation. Algorithm 1 presents the pseudo-code outlining this heuristic.

5.1.3 Selection operator

There are several commonly used selection algorithms in MAs for selecting individuals for the mating pool. This study examines two selection operators:

• K-way tournament: This method selects individuals from a population based on their fitness values. It randomly picks k individuals, evaluates their fitness, and selects the fittest individual as a parent.

• Roulette wheel: This method calculates a probability of being selected for all the individuals in the solution. It is calculated based on the fitness value. This process guarantees that individuals with higher fitness values have more chances to be selected as parents, as their corresponding slices are proportionally larger. It also allows individuals with lower fitness values to be chosen, preserving diversity in the population and preventing premature convergence.

Section 6 of this article provides detailed information about the tests conducted using these two selection operators and their efficiency in the context of our ESBRP.

5.1.4 Elitism

Elitism involves selecting a number of the highest-performing individuals to keep them in the next generation. This approach prevents the loss of valuable solutions through the different iterations of the algorithm. In our MA, we retain the best individual from each cluster for the next generation during each iteration.

5.1.5 Crossover

Crossover is a genetic operator essential for the interchange of genetic material between parent chromosomes, producing offspring solutions. By recombining genetic material from various parent solutions, crossover can yield better offspring solutions with improved properties. Nevertheless, finding the optimal crossing points can be fairly challenging, especially in our ESBRP, where complex solution representations are used. Therefore, we devised a customized crossover mechanism to preserve the quality and viability of the final solutions. Figure 6 illustrates an overview of our proposed crossover operator, which works as follows:

• A random mask is applied to Parent1. This mask is an array of 0s and 1s, and its length is equal to the number of routes/mini-chromosomes in Parent1.

• The routes/mini-chromosomes corresponding to 0 in Parent1 are removed, while those corresponding to 1 are retained.

• The elements of the routes remaining in Parent1 are deleted from Parent2.

• The new Parent1 and new Parent2 are then concatenated to produce Child1.

• Verification of the capacity constraint is applied to Child1. The time constraint is, by default, respected.

• If Child1 requires any modification, it undergoes a correction procedure, which we detail later in this section, to ensure it complies with capacity constraints.

The same process is used to obtain Child 2.

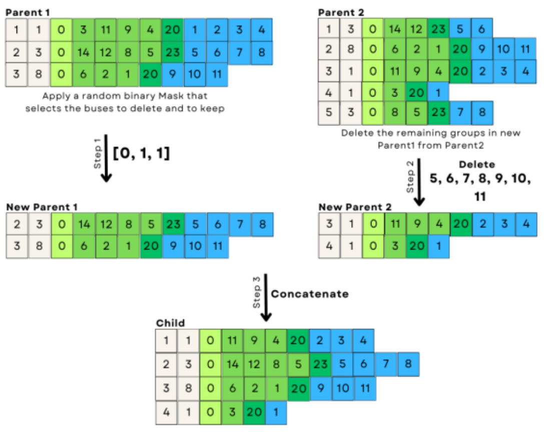
Parent 1
Parent 2
Apply a random binary Mask that selects the buses to delete and to keep
Delete the remaining groups in new Parent1 from Parent2
Step 1
[0, 1, 1]
Step 2
Delete 5, 6, 7, 8, 9, 10, 11
New Parent 1
New Parent 2
Step 3
Concatenate
Child

Figure 6. The crossover operator of the proposed MA

5.1.6 Mutation

Making small, haphazard mutations to the solutions helps maintain genetic diversity in the MA. Through mutation, specific genes or parameters are randomly selected and altered within an individual's chromosome representation based on a predetermined probability. However, identifying the appropriate mutation operator for our ESBRP is not straightforward. Therefore, we implemented a personalized method, as illustrated in Figure 7.

• We randomly select a set of routes or mini-chromosomes from the chromosome.

• We randomly choose two nodes and exchange them within each of the selected routes or mini-chromosomes. These nodes must be bus stops, as the workplace in a single-load scenario is always the last node to be visited in a route.

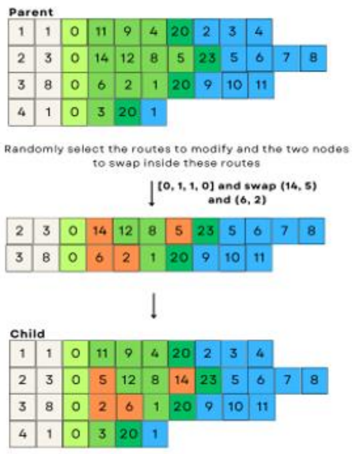
Parent
Randomly select the routes to modify and the two nodes to swap inside these routes
[0, 1, 1, 0] and swap (14, 5) and (6, 2)
Child

Figure 7. The mutation procedure of the MA

5.1.7 Correction procedure

After the crossover and mutation operators, a correction procedure is essential to address any violations of the ESBRP constraints. In our case, this correction procedure begins by identifying the problematic routes/mini-chromosomes in a solution and grouping the various sets that need to be reinserted into an array. Next, we attempt to insert these groups into existing routes based on the remaining capacity of each route, ensuring that the group aligns with the route's destination and arrival time. For any remaining groups that cannot be inserted into existing routes/mini-chromosomes, we create new routes.

5.1.8 Local search

The local search used in our MA focuses on finding the best permutation of each route's nodes by iteratively exploring neighboring options. It starts with a random permutation and evaluates several neighboring permutations to select the one with the best fitness value. This process continues until no further improvements can be made.

The following subsection presents the second step of the MSP framework.

5.1.9 Mutation

Making small, haphazard mutations to the solutions helps maintain genetic diversity in the MA. Through mutation, specific genes or parameters.

5.2 Set partitioning

After generating solutions with the minimum distance traveled using MA, the second step of our MSP framework involves assigning these solutions to the heterogeneous fleet. To accomplish this, we formulated the ESBRP as a SPP with the objective of minimizing the total cost. The formulation is as follows:

Sets and Data

$R = \{1, 2, ..., z\}$ set of feasible solutions generated by the MA where z is the total number of possible solutions.

$G = \{1, 2, ..., l\}$ set of employees' groups where l is the number of groups.

$K = \{1, 2, ..., b\}$ set of buses where b is the number of buses.

r = is a route of the set of routes R.

g = is a group of the set of employee groups G.

d_r = the total travel distance of a route r.

e_g = the number of people in a group g.

q_k = capacity of a vehicle k.

c_k = the cost of using a vehicle k.

Variables

x_{rk} = 1 if the route r is considered in the solution and assigned to vehicle k, and 0 otherwise.

y_{gr} = 1 if a group g is in route r, and 0 otherwise.

Objective

Minimize

$$\sum_{r \in R} \sum_{k \in K} d_r \times c_k \times x_{rk}$$

Constraints

$$\sum_{r \in R} \sum_{k \in K} y_{gr} \times x_{rk} = 1 \quad \forall g \in G \tag{20}$$

$$\sum_{g \in G} (y_{gr} \times e_g) \times x_{rk} \leq q_k \quad \forall r \in R \ \forall k \in K \tag{21}$$

$$\sum_{r \in R} x_{rk} \leq 1 \quad \forall k \in K \tag{22}$$

Constraint (20) specifies that each group of employees must covered by one route. Constraint (21) requires that the capacity of each vehicle k must be respected. Constraint (22) precises that it is forbidden to assign more than one route to a bus k.

5.3 Single to mixed loads heuristic

The SMH facilitates the transition from single-load routes, where each bus serves only one workplace, to a more efficient mixed-load configuration. In this scenario, buses can transport groups working in different workplaces. Compared to single-load routes, this approach reduces the number of vehicles needed, thereby lowering operational costs. However, the challenges of transitioning from a single-load route to a mixed-load route, while adhering to capacity and time constraints, is not straightforward and requires going through multiple steps. Algorithm 2 outlines the procedure for the SMH. This heuristic operates as follows:

- In this section, 'workplace/time' refers to a workplace with its associated arrival time, and 'SL solution' means a solution generated in the single-load scenario.

- The first step in SMH is to choose an SL solution and calculate the remaining capacity in each bus used in this solution.

- The second step is to generate clusters of workplace/time that can be added to each bus route of the SL solution:

• Why do we need to create clusters? Since we have capacity and arrival time constraints that need to be respected, we cannot randomly combine groups in one route. Therefore, based on the list of workplace/time, we need to define which ones can be combined.

• To know how many workplace/time will be combined, we define the number of workplaces that a bus can visit. For example, if we choose to have buses visiting three workplaces, then the clusters we form are composed of three groups. If we want the bus to visit five workplaces, we create clusters of five groups.

• As mentioned earlier, the clustering is based on the number of workplaces we want to have in a bus route and the feasibility concerning arrival times. For example, to cluster two workplace/time, the travel time between the two workplaces must be less than the absolute difference between the arrival times. For clusters containing more than two groups, this condition must be verified among all groups within the cluster.

• Another important criterion in the clustering process is that it is not permissible to group workplace/time that have the same workplace but different arrival times, or workplace/time that have different workplaces but the same arrival time.

• The third step is to generate the clusters of workplace/time for each route of the SL solution. Then, from the workplace/time clusters of the route, we determine all the possible employee groups that can be inserted into it. The output of this procedure is a set of new mixed-load routes. The insertion process is explained in Algorithm 2.

• The fourth step is to repeat the second and third steps for each route of the SL solution. Then repeat the whole process for other SL solutions.

• The final step is to input the set of new mixed-load routes (generated for all the routes in the SL solution) into the SPP presented earlier to obtain the final mixed-load solution.

• One last point regarding the clustering is to justify the choice of having clusters of different sizes (grouping two workplace/time, three workplace/time or more). The variety in the size of clusters is essential to our framework, as the SPP will process the mixed-load solutions. Providing only clusters of a single size significantly increases the likelihood that the SPP will fail to find feasible solutions.

The following section presents the results obtained for the proposed MSP framework compared to a simple MA and the MIP.

Algorithm 2. Single to Mixed Loads Heuristic

```
Procedure Create Clusters - grouping two entities
 Groups: groups of employees
 $g_1$ and $g_2$: groups in Groups
 Condition1: destination of $g_1$ different than destination of $g_2$
 Condition2: arrival time of $g_1$ different than arrival time of $g_2$
 Condition3: the difference between arrival time of g1 and arrival time of g2 superior than travel time between g1 and g2
 For g in Groups do
   If Condition1 and Condition2 and Condition3 are TRUE do
     Create clusters of two groups for g
   End if
 End for
End Procedure Create Clusters - grouping two entities

Procedure Insertion of groups in a route
 Groups: groups of employees
 For g in Groups do
   Create Clusters for group g
   For element in clusters of g do
     If number of people in route + number of people in element < capacity vehicle do
       Insert element in route based on best distance
     End if
   End for
   Return set of new mixed loads routes
End Procedure Insertion of groups in a route
Repeat Procedure Insertion of groups in a route for all routes in chromosome
Pass the new generated routes into SPP for vehicle affectation
```

6. COMPUTATIONAL RESULTS

This section exhibits and analyzes the outcomes of various tests conducted using the proposed MSP framework. The first subsection provides insights into the dataset utilized for the tests. The second subsection outlines the comparative procedure employed to determine the near-optimal parameters for the MSP framework, including population size, crossover probability, and mutation rate. The third subsection thoroughly analyzes the results of the MSP framework compared to a simple MA and the MIP's solution. We also examine the impact of single-load and mixed-load scenarios on riding time and cost efficiency.

6.1 Data

This study utilizes real-world data from a large Moroccan industrial group that is present in multiple geographical regions across Morocco. In this work, we solely focus on testing MSP in the largest region. The company shared with us data that comprises the following information:

- As shown in Table 1, the vehicle fleet includes five types, distinguished by capacity, number of vehicles, and cost of using each type.
- A matrix detailing the distances between different nodes (the depot, bus stops, and workplaces), along with their GPS locations.
- A list of the various timeslots scheduled throughout the day. By "timeslot," we refer to a time interval designated for transporting specific groups of employees between their bus stops and workplaces.
- A table outlining the groups of employees waiting at each bus stop, including their sizes, destinations, and arrival times. Table 2 displays an example of employee distribution within a one-hour timeslot.

Furthermore, the bus stops where employees await pick-up are fixed and predefined. Most workplaces are located outside the city in rural areas. While the dataset is generally stable, changes may occur if employees modify their addresses, impacting bus stops. Other modifications may arise due to strikes or technical issues related to the vehicle fleet.

For data manipulation, we processed the raw data using Python libraries to prepare it for input into our MSP framework and organized it into Excel files.

Table 1. A presentation of the vehicle fleet of our ESBRP problem

Types of Buses	1	2	3	4	5
Number of Buses per Type	26	4	20	10	4
Capacity of Each Bus Type	48	15	48	17	28
Cost of Use per km	3.5	2.5	9.4	4.8	6.27

Table 2. An example of the number of employees per stop in one of the company's areas from 6 am to 7 am

Bus Stops	1	2	3	4	5	6	7	8	9	10	11
Number of Employees	49	31	31	22	25	24	33	43	20	39	27

6.2 Parameters and testing environment

This subsection outlines the process we followed to determine the most suitable values for the parameters of the proposed MSP framework: the size of the population, the number of iterations, the crossover rate, and the mutation rate.

Our first set of preliminary tests aimed to identify the optimal population size and number of iterations. In these tests, we fixed the values for crossover and mutation rates while systematically varying the population size and the number of iterations. We assessed the quality of the solution based on the total distance. We observed no significant improvements in solution quality with a population size exceeding 200. For the number of iterations, we chose the value at which the MA converged to the best solution among those already generated. Specifically, we found that 150 iterations and a population size of 200 were sufficient to converge to a satisfactory solution in terms of total distance. Table 3 summarizes the results.

Table 3. The best results obtained in terms of total distance when testing the proposed MA based on the number of iterations and the population size

Number of Iterations \ Population Size	100	150	200	300	400
50	906.50	884.42	884.02	884.04	884.01
100	902.98	875.68	872.46	872.43	872.43
150	895.49	870.19	**865.31**	865.31	865.31
200	890.12	887.04	872.59	872.58	872.59

Table 4. The results of the proposed MA in terms of total distance when varying the crossover and mutation rates

Mutation Rate \ Crossover Rate	0.01	0.04	0.08	0.1	0.3	0.4
0.6	918.29	910.35	891.71	913.90	888.41	917.25
0.7	922.36	886.39	891.18	916.67	927.09	903.77
0.8	903.99	906.49	893.68	894.74	907.72	904.94
0.85	915.57	**872.75**	901.72	895.46	923.37	874.58
0.9	921.12	890.87	887.94	885.38	925.59	913.67
0.95	911.79	886.52	878.74	881.46	888.63	886.16

The second set of tests focused on determining the best combination of crossover and mutation rates. In this phase, we fixed the population size and the number of iterations while varying the crossover and mutation rates. Our decision was based on the quality of the solution obtained at the final iteration and the convergence of the framework with these values. The results are also highlighted in Table 4. Consequently, we adopted mutation and crossover rates of 0.04 and 0.85, respectively.

Additionally, our testing environment included a laptop with an Intel i7 12th generation processor and 32 GB of RAM, and we utilized Python as the programming language.

6.3 Single load

As previously mentioned, the ESBRP is an NP-hard problem, which makes it very complex to solve optimally. The difficulty arises from factors such as the number of nodes, the complexity of the problem, and the constraints involved. In this subsection, we highlight the results obtained from the MSP framework, comparing it to a simple MA and the MIP solved using CPLEX. In these tests, the goal is to minimize the operational cost in a Single Load scenario. We conducted experiments using the dataset provided by the company, as well as instances we created based on that data. These instances vary in size (number of nodes), the number of employees, and the number of buses.

During our experimentation phase, we initially used the conventional k-tournament selection operator commonly employed in traditional evolutionary approaches. However, as shown in Figure 8, we observed that the MA converged too quickly after only a short period of iterations. To address this issue, we experimented with the roulette wheel selection technique, which showed some improvement but still tended toward premature convergence.

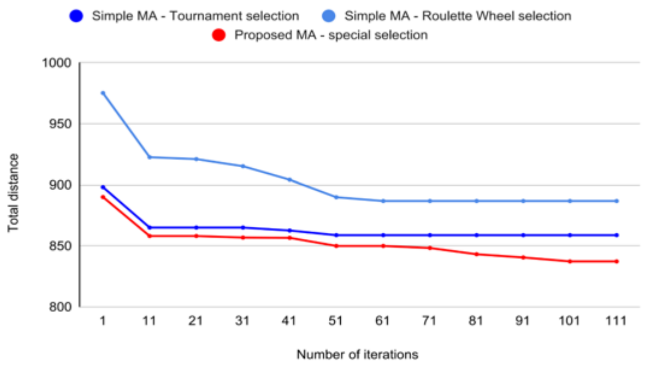

Figure 8. Comparison of the convergence pace of a simple MA and the proposed MA, based on the selection operator used

Recognizing the necessity of finding a solution to prevent premature convergence, we successfully adapted the approach proposed by Malik and Wadhwa [28]. This approach involved dividing the initial population into several clusters and applying the MA operators to each cluster. To analyze the efficiency of the proposed MA in this work, Figure 8 illustrates a comparison of the convergence pace for the following three approaches:

• Simple MA - Tournament Selection: a MA that uses the tournament selection defined in section 5.1.3.

• Simple MA - Roulette Wheel Selection: a MA that uses the roulette wheel selection defined in section 5.1.3.

• Proposed MA - Special Selection: the MA explained in section 5.1.

Figure 8 shows no premature convergence issue in the Proposed MA - Special Selection Approach, while the other two algorithms converge prematurely to a local optimum.

After deciding on the type of selection to adopt, we tested and compared three approaches—MIP, MSP, and MA—for the single load scenario, with the objective of minimizing total cost. As shown in Table 5, finding a solution using MIP becomes impossible for more than 25 nodes. For smaller instances, specifically between 6 and 14 nodes, the proposed MSP framework achieves the same optimal solution as MIP. For medium-sized instances, ranging from 18 to 25 nodes, we fixed a maximum execution time of 1 hour and a half for MIP. The results obtained by the MSP framework were superior to the feasible solutions obtained by MIP in significantly less time. For the final instance of 33 nodes, we could not compare the result of MSP with MIP, as no solution was found within that time.

Compared to MA, the MSP framework consistently demonstrated greater efficiency and provided better solutions in very short periods. As mentioned earlier, the choice of MA's parameters is crucial and significantly affects the algorithm's convergence. In our case, even if the MA does not converge to the optimal solution, we were able to utilize the pool of solutions it generated, which yielded better results in less time. This indicates that breaking down the decisions of the ESBRP problem into two distinct steps was more beneficial than treating it in one single phase.

Table 5. Optimal and feasible solutions found for 10 instances

Instances	Size of Instances		Exact Solution - Single Load		MA - Single Load		MSP - Single Load	
	Number of Vehicles	Number of Nodes	Best Found Total Cost	Execution Time in Second	Best Found Total Cost	Execution Time in Second	Best Found Total Cost	Execution Time in Second
1	8	6	462.249	0.27	463.278	30	462.249	30.02
2	8	10	1173.57	23.31	1182.94	62	1173.57	62.02
3	9	11	1118.93	58.27	1165.58	87	1118.93	87.03
4	12	12	849.873	72.14	972.18	149	849.873	149.02
5	14	13	1045.36	429.84	1177.73	236	1045.36	236.03
6	14	14	1031.72	2600	1343.29	876	1031.72	876.03
7	19	18	1416.23	4001,17	1778.823	1249	1343.9	1249.02
8	20	20	1614.42	5400	2053.47	1590	1538.31	1590.05
9	25	25	1589.88	5400	1481.04	1873	1383.61	1873.05
10	64	33	None	5400	3000.80	2460	2775.66	2460.11

6.4 Mixed load

In Figure 9, we present the results of the MSP framework in a scenario with mixed loads. One mixed-load solution can be generated from several single-load solutions. Therefore, the number of solutions in Figure 9 indicates how many single-load solutions were used to create the mixed-load solution. As shown, using more single-load solutions to create mixed-load solutions improves the quality of the result. However, we could only test up to 20 single-load solutions, as the execution time of the SPP increased exponentially.

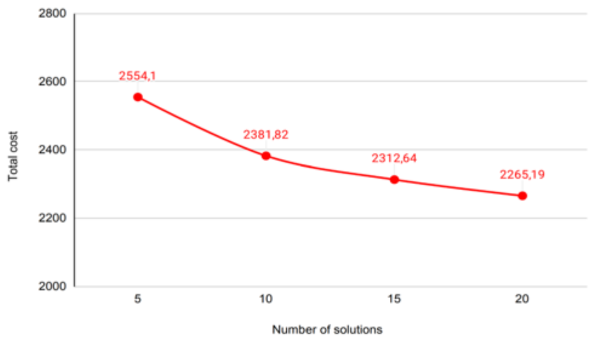

Figure 9. The results obtained for the MSP mixed loads in terms of total cost

In the single-load scenario, at least one bus per workplace is required, ultimately increasing the overall cost. Therefore, MSP ML is a more efficient approach than MSP SL regarding the total cost and the number of buses used. In contrast, after comparing the maximum riding time between MSP ML, MSP SL, and Simple MA SL, we determined that Simple MA SL is superior in this regard, despite being the least cost-efficient. The maximum riding-time is the amount of time a passenger spends on the bus before arriving at their workplace.

If the decision-maker prioritizes optimizing riding time, MSP ML may not be suitable, even though it offers the best solution regarding total distance, cost, and number of buses used. Minimizing riding time typically results in fewer stops per bus, necessitating additional buses to pick up all employee groups. Consequently, the best trade-off between cost-effectiveness and maximum riding time can be achieved using MSP SL. However, if enabling mixed loads is a requirement, an alternative approach should be considered.

6.5 Results analysis

Based on the earlier results for both SL and ML ESBRP, the proposed MSP framework achieved a significant cost reduction of approximately 4% for SL and 6% for ML compared to the traditional routing method used by the company. This improvement is primarily attributed to an average decrease of 5% in total travel distance for SL and a 6% reduction for ML, resulting in lower fuel consumption and reduced costs. Additionally, by optimizing route assignment to buses, we reduced the number of required buses by 3% for SL and 5% for ML, leading to substantial operational savings for the company.

The MSP framework also considered arrival times at workplaces, ensuring that all employees arrived on time; a guarantee not provided by the company's used method. This adjustment allows employees to spend less time commuting and more time at work. Therefore, these efficiency improvements positively impact both operational performance and employee satisfaction.

The results demonstrate that hybridizing a MA with set partitioning is effective not only in optimizing employee shuttle routes but also as a practical decision-support tool for transportation managers. The reduced costs, decreased commuting time, and lower number of vehicles used translate into financial and environmental benefits while maintaining a high level of efficiency and employee satisfaction.

7. CONCLUSIONS

This paper presents a matheuristic MSP to solve the transportation problem of a large industrial company. The business requires solutions that enable cost reduction while maintaining flexibility in allowing or prohibiting mixed loads. Consequently, we introduce the MSP framework with the SMH heuristic, which allows the company to seamlessly switch between single and mixed loads based on their needs. In this context, we address the ESBRP problem for both single and mixed-load scenarios using the MSP framework. Therefore, this paper presents two main contributions:

- To the best of our knowledge, no paper addressed the ESBRP with Time constraints using a matheuristic and in two steps, each with a different objective.
- The results of the MSP framework compared to MIP proved the efficiency of our approach and reached optimal solutions within very short execution times.

Our analysis and results indicate that dividing the ESBRP problem into two sub-problems and hybridizing a metaheuristic with an exact method is more efficient than treating the problem in a single phase (Simple MA). The two sub-problems we adopted are as follows: first, generating routes that minimize overall distance using the proposed metaheuristic; and second, allocating these routes to buses using the SPP to minimize overall costs.

However, this study is limited in terms of the objectives considered. In practice, it is essential to reach a trade-off between transportation costs, vehicle fleet composition, and the riding time for passengers to reach their workplaces. Additionally, this work only considered data from a one-hour timeslot with one-way trips, whereas the real problem requires consideration of round trips and multi-tours per vehicle. These aspects will be addressed in our future work.

To achieve this, the next step is to explore multi-objective optimization techniques that can help find the optimal trade-off between costs and riding time. We also aim to optimize the company's vehicle fleet, which will necessitate a detailed analysis of the types of vehicles used and their associated costs.

Furthermore, one challenging aspect encountered when implementing the proposed MSP framework was setting the parameters of the metaheuristic. Despite conducting numerous tests to determine suitable parameter values, many factors need consideration to establish a stable choice. Therefore, we seek to integrate machine learning algorithms to predict parameter values that would be appropriate for our problem. This integration would further enhance the overall efficiency of the company's employee transportation system, resulting in a more sustainable and reliable mode of transportation for all passengers.

REFERENCES

[1] Kumar, S.N., Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. Intelligent Information Management, 4(3): 19355. https://doi.org/10.4236/iim.2012.43010

[2] Luo, W., Fu, Z. (2010). A variable neighborhood tabu search algorithm for the heterogeneous fleet vehicle routing problem with time windows. In 2010 International Conference on Logistics Engineering and Intelligent Transportation Systems, Wuhan, China, pp. 1-4. https://doi.org/10.1109/LEITS.2010.5665040

[3] Toth, P., Vigo, D. (2002). The Vehicle Routing Problem. Society for Industrial and Applied Mathematics.

[4] Yüceer, Ü. (2013). An employee transporting problem. Journal of Industrial Engineering International, 9(1): 31. https://doi.org/10.1186/2251-712X-9-31

[5] Peker, G., Eliiyi, D.T. (2023). Employee shuttle bus routing problem: A case study. Avrupa Bilim ve Teknoloji Dergisi, 46: 151-160. https://doi.org/10.31590/ejosat.1173057

[6] Park, J., Kim, B.I. (2010). The school bus routing problem: A review. European Journal of Operational Research, 202(2): 311-319. https://doi.org/10.1016/j.ejor.2009.05.017

[7] Alvarenga, G.B., Mateus, G.R., De Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. Computers & Operations Research, 34(6): 1561-1584. https://doi.org/10.1016/j.cor.2005.07.025

[8] Newton, R.M., Thomas, W.H. (1969). Design of school bus routes by computer. Socio-Economic Planning Sciences, 3(1): 75-85. https://doi.org/10.1016/0038-0121(69)90051-2

[9] Li, C., Gong, L., Luo, Z., Lim, A. (2019). A branch-and-price-and-cut algorithm for a pickup and delivery problem in retailing. Omega, 89: 71-91. https://doi.org/10.1016/j.omega.2018.09.014

[10] Schittekat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F., Springael, J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. European Journal of Operational Research, 229(2): 518-528. https://doi.org/10.1016/j.ejor.2013.02.025

[11] Kumar, Y., Jain, S. (2015). School bus routing based on branch and bound approach. In 2015 International Conference on Computer, Communication and Control (IC4), Indore, India, pp. 1-4. https://doi.org/10.1109/IC4.2015.7375684

[12] Arias-Rojas, J.S., Jiménez, J.F., Montoya-Torres, J.R. (2012). Solving of school bus routing problem by ant colony optimization. Revista EIA, (17): 193-208.

[13] Huo, L., Yan, G., Fan, B., Wang, H., Gao, W. (2014). School bus routing problem based on ant colony optimization algorithm. In 2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), Beijing, China, pp. 1-5. https://doi.org/10.1109/ITEC-AP.2014.6940973

[14] Euchi, J., 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. https://doi.org/10.1016/j.swevo.2011.10.002

[15] Faraj, M.F., Sarubbi, J.F., Silva, C.M., Porto, M.F., Nunes, N.T.R. (2014). A real geographical application

for the school bus routing problem. In 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, pp. 2762-2767. https://doi.org/10.1109/ITSC.2014.6958132

[16] Sghaier, S.B., Guedria, N.B., Mraihi, R. (2013). Solving school bus routing problem with genetic algorithm. In 2013 International Conference on Advanced Logistics and Transport, Sousse, Tunisia, pp. 7-12. https://doi.org/10.1109/ICAdLT.2013.6568426

[17] Oluwadare, S.A., Oguntuyi, I.P., Nwaiwu, J.C. (2018). Solving school bus routing problem using genetic algorithm-based model. International Journal of Intelligent Systems and Applications, 12(3): 50. https://doi.org/10.5815/ijisa.2018.03.06

[18] Díaz-Parra, O., Ruiz-Vanoye, J.A., de los Ángeles Buenabad-Arias, M., Saenz, A.C. (2013). Vertical transfer algorithm for the school bus routing problem. In Transactions on Computational Science XXI: Special Issue on Innovations in Nature-Inspired Computing and Applications, pp. 211-229. https://doi.org/10.1007/978-3-642-45318-2_9

[19] Chalkia, E., Salanova Grau, J.M., Bekiaris, E., Ayfandopoulou, G., Ferarini, C., Mitsakis, E. (2016). Safety bus routing for the transportation of pupils to school. Traffic Safety, 4: 283-299.

[20] Ellegood, W.A., Solomon, S., North, J., Campbell, J.F. (2020). School bus routing problem: Contemporary trends and research directions. Omega, 95: 102056. https://doi.org/10.1016/j.omega.2019.03.014

[21] Park, J., Kim, B.I. (2010). The school bus routing problem: A review. European Journal of Operational Research, 202(2): 311-319. https://doi.org/10.1016/j.ejor.2009.05.017

[22] Hart, E., Sim, K., Urquhart, N. (2014). A real-world employee scheduling and routing application. In Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, BC, Vancouver, Canada, pp. 1239-1242. https://doi.org/10.1145/2598394.260544

[23] Leksakul, K., Smutkupt, U., Jintawiwat, R., Phongmoo, S. (2017). Heuristic approach for solving employee bus routes in a large-scale industrial factory. Advanced Engineering Informatics, 32: 176-187. https://doi.org/10.1016/j.aei.2017.02.006

[24] Purba, A.P., Siswanto, N., Rusdiansyah, A. (2020). Routing and scheduling employee transportation using tabu search. AIP Conference Proceedings, 2217(1): 030143. https://doi.org/10.1063/5.0000766

[25] Peker, G., Eliiyi, D.T. (2022). Shuttle bus service routing: A systematic literature review. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 28(1): 160-172.

[26] Bideq, H., Ouaddi, K., Gorge, A., Ellaia, R. (2022). A real-world employee bus routing problem application with mixed loads. In 2022 IEEE 6th International Conference on Logistics Operations Management (GOL), Strasbourg, France, pp. 1-7. https://doi.org/10.1109/GOL53975.2022.9820604

[27] Neri, F., Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation, 2: 1-14. https://doi.org/10.1016/j.swevo.2011.11.003

[28] Malik, S., Wadhwa, S. (2014). Preventing premature convergence in genetic algorithm using DGCA and elitist technique. International Journal of Advanced Research in Computer Science and Software Engineering, 4(6): 410-418.

[29] Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2): 254-266. https://doi.org/10.1287/opre.35.2.254