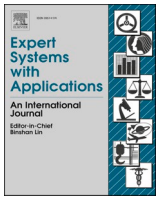
Expert Systems With Applications 263 (2025) 125720

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Bus scheduling with heterogeneous fleets: Formulation and hybrid metaheuristic algorithms

Mohammad Sadrani [a,*], Alejandro Tirachini [b,c], Constantinos Antoniou [a]

[a] Chair of Transportation Systems Engineering, TUM School of Engineering and Design, Technical University of Munich, Munich 80333, Germany
[b] Department of Civil Engineering and Management, University of Twente, Enschede, the Netherlands
[c] Department of Civil Engineering, Universidad de Chile, Santiago, Chile

ARTICLE INFO

Keywords:
Public transport
Optimization
Mixed-fleet bus scheduling
Resource allocation
Trip comfort
Hybrid metaheuristics

ABSTRACT

This paper focuses on optimizing mixed-fleet bus scheduling (MFBS) with vehicles of different sizes in public transport systems. We develop a novel mixed-integer nonlinear programming (MINLP) model to address the MFBS problem by optimizing vehicle assignment and dispatching programs. The model considers user costs, operator costs, and the crowding inconvenience of standing and sitting passengers. To tackle the complexity of the MFBS problem, we employ Genetic Algorithm (GA) and Grey Wolf Optimizer (GWO). Besides, we develop two hybrid metaheuristics, including GA-SA [a combination of GA and Simulated Annealing (SA)] and GWO-SA (a combination of GWO and SA), to improve optimization capabilities for the MFBS problem. We also employ a Taguchi approach to fine-tune the metaheuristics' parameters. We widely examine and compare the metaheuristics' performance across various-sized samples (small, medium, and large), considering solution quality, computational time, and the result stability of each algorithm. We also compare the metaheuristics' solutions with the optimal solutions acquired by GAMS software in small and medium-scale samples. Our findings show that the GWO-SA outperforms the other metaheuristics. Applying our model to a real bus corridor in Santiago, Chile, we find that precise dispatching plans generated by more sophisticated/advanced algorithms (GA-SA and GWO-SA) lead to larger cost savings and improved performance compared to simpler algorithms (GA and GWO). Interestingly, utilizing more advanced algorithms makes a difference in terms of fleet planning in crowded scenarios, whereas for low and medium-demand cases, simpler dispatching algorithms could be used without a drop in accuracy.

1. Introduction

1.1. Research motivation

Operational scheduling is a crucial stage in the planning process for public transport (PT) systems, encompassing tasks such as vehicle scheduling, crew scheduling, and real-time control strategies (Gkiotsalitis et al., 2022; Sadrani et al., 2022a). The operational planning stage enables transport operators to improve the level of service offered to travelers through more effective resource allocation. These operational planning problems, associated with short-term plans, should be solved periodically during operations, on a weekly basis, daily basis, or multiple times per day at various planning intervals. This periodicity ensures that the most up-to-date operating conditions, such as changes in passenger demand over time and space and weather conditions, are taken into account. Therefore, in addition to the quality and accuracy of planning, the processing time required to solve these problems is of practical significance for policymakers. However, the computational complexity of operational bus scheduling problems poses challenges for exact solution methods in addressing these problems efficiently within reasonable time frames (Perumal et al., 2022). This issue underlines a need for the design and selection of more efficient solution algorithms for problems of this nature, such as metaheuristic algorithms, which can offer the advantage of discovering practically good (not necessarily exact) solutions within rational computational (CPU) times (Ge et al., 2022; Tang et al., 2022). Besides, this computational difficulty will increase even more when planners aim to incorporate more practical elements into the operational scheduling of bus services, such as mixed fleet configurations, users' trip comfort, and driving time uncertainty.

Several studies have employed metaheuristics to address various PT

* Corresponding author.
E-mail addresses: m.sadrani@tum.de (M. Sadrani), Alejandro.tirachini@utwente.nl (A. Tirachini), c.antoniou@tum.de (C. Antoniou).

https://doi.org/10.1016/j.eswa.2024.125720
Received 14 December 2023; Received in revised form 21 September 2024; Accepted 4 November 2024
Available online 7 November 2024

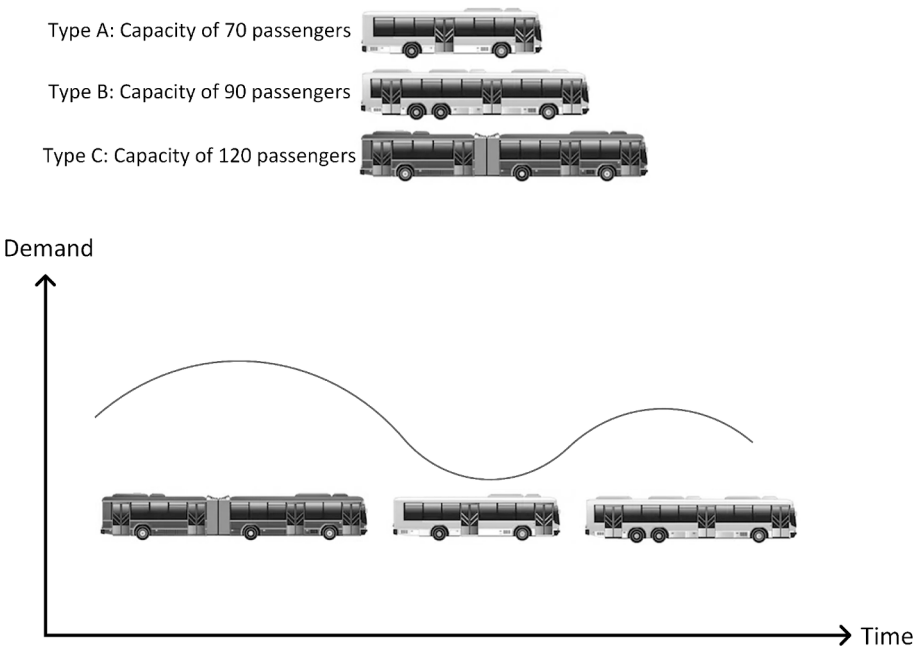


Fig. 1. Optimization concept of vehicle dispatching sequence in mixed-fleet deployments (allowing for better management of vehicle capacity in response to dynamic changes in passenger demand). For instance, consider a mixed fleet of 15 buses for operations: {A, A, A, A, A, A, B, B, B, B, B, C, C, C, C}, containing 6, 5, and 4 buses of type A, type B, and type C, respectively. Overall, $\frac{15!}{6! \times 5! \times 4!} = 630,630$ distinct ways can be proposed for the dispatching sequence of buses from the first station.

planning problems, such as vehicle dispatching setting problems (Zhang et al., 2018; Sadrani et al., 2022a), delay and disruption management problems (Wang et al., 2019; Xu et al., 2018), and stopping pattern determination problems (Chen et al., 2015; Mou et al., 2020). However, most existing research focuses on homogeneous fleets, in which the operation of vehicles of the same size is optimized. In contrast, this study addresses a bus dispatching problem within a heterogeneous operating environment, developing efficient modeling and solution tools to optimize dispatching programs for mixed fleets with buses of various sizes and capacities to cover passenger flows (see Fig. 1).

The practical relevance of this problem setting lies on the fact that operation with heterogeneous fleets is usual in PT systems, particularly in large and complex cities. For instance, the bus network in Santiago, Chile, is composed of 317 routes, out of which 116 routes (37 %) are operated with heterogeneous fleets during the morning peak period, either combining small (8-meter long) with standard (12-meter long) buses, or standard with articulated (18-meter long) buses in one single route (Sadrani et el., 2022a).

1.2. Related literature

We review three streams of studies to better contextualize our work within the existing literature: (i) homogeneous-fleet bus dispatching problems, (ii) mixed-fleet bus dispatching problems, and (iii) crowding discomfort in PT systems. In Section 1.2.1, we review studies addressing homogeneous-fleet bus dispatching problems, a topic that has been extensively studied in the bus scheduling literature. In Section 1.2.2, we review mixed-fleet bus dispatching problems, highlighting the limited research in this area compared to the extensive literature on homogeneous fleets. In Section 1.2.3, we review the literature on crowding in PT systems. We also highlight that, unlike the well-studied context of homogeneous-fleet scheduling, the inconvenience experienced by sitting and standing passengers due to crowding inside vehicles has not yet been investigated in the context of mixed-fleet bus scheduling.

1.2.1. Homogeneous-fleet bus dispatching problems

In this part, we review the stream of studies addressing the optimization of dispatching plans for homogeneous fleets, where all buses have the same type and capacity. This subject has been widely explored in the PT literature, as evidenced by several studies (e.g., Berrebi et al., 2015; Gkiotsalitis, 2020; Luo et al., 2019; Gkiotsalitis and Alesiani, 2019; Zhang and Liu, 2019; Gkiotsalitis and Liu, 2022). The primary objective of these efforts is to enhance the quality of service offered to passengers by optimizing the dispatching times of vehicles in response to temporal fluctuations in passenger demand. For instance, Gkiotsalitis and Alesiani (2019) proposed a bus dispatching problem, which considers uncertain passenger demand and driving times, to create a timetable with reliable dispatching times. They developed a Genetic Algorithm (GA) to address the problem and achieved a 5 % improvement in service regularity during operations based on an application to a bus line in Singapore. Similarly, Luo et al. (2019) addressed a dynamic bus dispatching problem to minimize users' waiting times, while accounting for dynamic changes in passenger demand and road congestion.

1.2.2. Mixed-fleet bus dispatching problems

In this part, we review studies focusing on mixed-fleet bus dispatching problems. Compared to research on homogeneous fleets, there is limited literature addressing planning problems for heterogeneous bus fleets (e.g., Ceder et al., 2013; Dai et al., 2020; Duran-Micco et al., 2020; Sadrani et al., 2022a). However, due to supply-and-demand dynamics and resource conditions in reality, fleets may consist of various bus sizes, requiring operators to deploy a mixed fleet to cover passenger volumes on a given route.

For instance, Ceder et al. (2013) proposed a bi-objective model to design timetables for deploying buses of different sizes, aiming to minimize variations in headway plans and vehicle load levels. Duran-Micco et al. (2020) addressed the determination of frequencies for mixed fleets in a bus network using a heuristic memetic algorithm based on the NSGA-II concept. The results demonstrated that mixed fleet deployments contribute to reduced greenhouse gas emissions and users' trip times. Dai et al. (2020) developed a real-time dispatching framework using an integer nonlinear programming formulation to assess the benefits of incorporating modular automated vehicle pods with varying

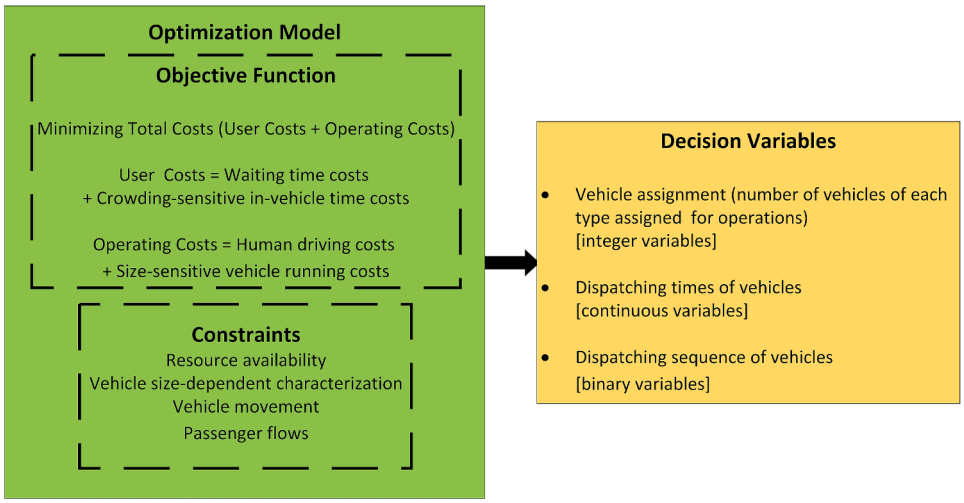


Fig. 2. Overview of the optimization model framework for the proposed mixed-fleet bus scheduling (MFBS) problem.

sizes alongside human-driven buses. They employed a dynamic programming algorithm but highlighted the need for more effective solution techniques, particularly in handling performance degradation with an increasing rolling horizon.

Li et al. (2019) proposed a hybrid simulated annealing heuristic for multistage heterogeneous fleet scheduling with fleet sizing decisions. Their model addressed vehicle allocation for various vehicle types and was formulated as a mixed integer programming problem. Rinaldi et al. (2020) developed a mixed integer linear programming model for scheduling mixed fleets of electric and hybrid buses. Their model included an ad-hoc decomposition scheme to improve scalability and was validated using case studies from Luxembourg City. The study demonstrated that the optimal handling of mixed-fleet conditions can significantly reduce operational costs in PT systems. Moreover, Frieß and Pferschy (2024) developed an optimization model for planning zero-emission mixed-fleet public bus systems with minimal life cycle costs. They evaluated overnight charging, opportunity charging, and hydrogen fuel cells, using an integer linear programming model to optimize technology combination and operational scheduling. Their findings indicate that mixed fleets can significantly reduce costs.

Our present model is based on our earlier work in Sadrani et al. (2022a), which developed a mixed-integer nonlinear programming (MINLP) model to address the dispatching problem of mixed bus fleets, aiming to minimize users' waiting times as demand profiles change over time. The NP-hardness of the problem was demonstrated due to its combinatorial nature, and a Simulated Annealing (SA) algorithm was proposed as a solution approach. However, that study has three important limitations that need to be addressed for a more comprehensive understanding of mixed-fleet operations.

First, Sadrani et al. (2022a) focused solely on minimizing passengers' waiting times, overlooking other crucial elements such as in-vehicle trip times, users' discomfort from in to vehicle crowding, and operator costs.

Second, the model simply assumed a fixed number of vehicles, disregarding the optimization of vehicle assignment programs that determine the number and type of vehicles required for mixed-fleet operations. In essence, the absence of a total cost objective function (including both user and operator costs) and resource constraints prevented the optimization of the allocated fleet for efficient operations. Specifically, in a mixed operating system, the running costs of vehicles are influenced by their size, with larger buses (e.g., 18-m long) incurring higher costs compared to smaller ones (e.g., 12-m long). However, larger buses offer the benefit of reducing crowding inconvenience for passengers, and also reduce the operator cost per passenger transported, due to the economics of scale present in bus transport. Thus, optimizing vehicle assignment programs requires a total cost analysis that accounts for both user and operator costs (Mohring, 1972; Jara-Díaz and Gschwender, 2003).

Third, that study only utilized the SA algorithm as a solution approach, without exploring other advanced algorithms. Given the high combinatorial complexity of the mixed-fleet bus scheduling problem, it is crucial to evaluate the performance of various algorithms to ensure robustness and effectiveness. Our research highlights the significance of utilizing more advanced algorithms to tackle the challenges of vehicle assignment optimization in crowded scenarios.

Overall, to bridge these gaps, the present research proposes a novel mixed-fleet bus scheduling (MFBS) problem that considers more realistic components, including in-vehicle trip times, crowding discomfort levels for both sitting and standing passengers, resource constraints, and operator costs (see Fig. 2). Moreover, our mathematical programming model goes beyond optimizing vehicle dispatching plans by also addressing the optimization of vehicle assignment programs, determining the number and type of vehicles assigned for mixed-fleet operations, thereby enabling efficient resource utilization. We also develop two hybrid metaheuristic algorithms to enhance optimization capabilities and solution quality for the MFBS problem.

Public transport planning typically includes strategic, tactical, and operational levels, which correspond to long-term, medium-term, and short-term planning decisions, respectively (Desaulniers and Hickman, 2007; Liu et al., 2021). In this work, we integrate tactical (vehicle assignment and dispatching time determination) and operational (dispatching sequence determination) levels in the context of mixed-fleet bus scheduling.

By integrating both levels, we capture the interactions between vehicle assignments (tactical) and dispatching sequences (operational) in mixed-fleet operations, an interdependency that, as far as we are aware of, has been explored for the first time in the literature. For example, our results show that different vehicle assignments require different dispatching sequences in mixed-fleet operations. This approach allows for more accurate matching of supply (vehicle capacity) to time-varying passenger demand, reducing in-vehicle crowding and improving overall service efficiency.

1.2.3. Crowding discomfort in public transport systems

In this part, we review the literature on crowding in PT systems. It is well known that PT users' satisfaction can be significantly impacted by in-vehicle comfort, as traveling in crowded vehicles (with high occupancy levels) raises the disutility of trip times for users (Wardman and Whelan, 2011; Tirachini et al., 2013; Hörcher et al., 2017; Drabicki et al., 2023). Passengers report increased levels of anxiety and stress, physical and mental discomfort, lack of personal space, and reduced

security if they are forced to travel in crowded vehicles (for a review, see Tirachini et al., 2013).

Moreover, it has been reported that the crowding discomfort experienced by sitting and standing passengers inside vehicles is not similar. Standing passengers tend to be more affected by in-vehicle crowding than sitting passengers, resulting in higher perceived crowding costs for those standing (Wardman and Whelan, 2011; Tirachini et al., 2017, 2016; Jenelius, 2020).

Several researchers have attempted to optimize the scheduling of homogeneous bus fleets while considering the effects of in-vehicle crowding on passenger discomfort (e.g., Tirachini et al., 2014; Shang et al., 2019; Suman and Bolia, 2019; Agrawal et al., 2020; Sadrani et al., 2022b, 2023). Nonetheless, to the best of our knowledge, Shang et al. (2023) is the only study considering crowding discomfort costs in the bus scheduling problem with multiple vehicle types. However, the difference in crowding discomfort levels perceived by standing and sitting passengers has not been addressed.

1.2.4. Summary of research gaps

The research gaps identified in the literature are summarized here. First, no research in the literature has tackled the joint optimization of vehicle assignment programs and vehicle dispatching programs for mixed-fleet operations. Second, considering the difference in crowding discomfort levels experienced by standing and sitting passengers and incorporating this factor into the scheduling of mixed bus fleets is another research gap in the existing literature. Third, incorporating such realistic elements into a comprehensive programming model increases the solution complexity of the MFBS problem. Thus, there is a need to develop more reliable and advanced solution algorithms (such as novel hybrid metaheuristics) to address these challenges effectively.

1.3. Synthesis and contribution statement

To fill the identified research gaps, we propose a MFBS problem formulation using a novel MINLP model. Our model optimizes the number and type of vehicles assigned for operations, as well as their dispatching order and times. We explicitly consider the discomfort caused by crowding for both standing and sitting passengers. Furthermore, we propose a novel framework to manage the size-sensitive characteristics of vehicles throughout operations, such as their in-vehicle capacity and running costs.

To solve the MFBS problem, we employ GA and Grey Wolf Optimizer (GWO). Moreover, we develop two hybrid metaheuristics by integrating the Simulated Annealing (SA) strategy into these algorithms (GA-SA and GWO-SA) to enhance their optimization capabilities. Several numerical experiments are conducted to evaluate the performance of these metaheuristics in terms of CPU time and solution quality.

To validate the practical applicability of our research, we apply the developed model and metaheuristics to a real-world bus corridor in Santiago, Chile, characterized by high passenger crowding levels during the morning peak period. We conduct sensitivity analyses to evaluate the sensitivity of mixed-fleet deployment programs to factors such as demand levels, solution techniques, crowding inconvenience valuations, and uncertain driving times. Our real-world application shows that the choice of metaheuristic for the MFBS is not innocuous, as the most sophisticated algorithms perform better in terms of improving trip comfort precisely when crowding levels are high. Through this empirical application, we offer valuable managerial insights for planning mixed bus fleets, thus improving service efficiency and passenger comfort.

The rest of this paper is structured as follows. Section 2 presents a mathematical model for the MFBS problem, followed by Section 3, including the solution algorithms designed to address the problem. In Section 4, various test samples are solved to analyze the performance of the metaheuristics. Section 5 assesses the model's and metaheuristics' applications in a real case study. Ultimately, conclusions are offered in Section 6, along with recommendations for more research.

2. Problem formulation

In this section, we design and present a MINLP model for the MFBS problem. The timetable is determined by optimizing the dispatching times of vehicles, which are defined as decision variables in our model. Our integrated approach focuses on tactical decisions (deciding how many vehicles of each type should be assigned for operations and their dispatching times in the timetable) and operational decisions (determining the dispatching sequence of those vehicles). This problem is particularly relevant for operators managing heterogeneous bus fleets, which may arise due to supply–demand dynamics, resource constraints, and historical reasons where buses of different types were purchased at different times through different contracts. In such cases, it is reasonable to assume that the total fleet size per vehicle type (the number of buses of each type) has already been determined during prior strategic planning phases, and our task is to optimize their assignment and dispatch to meet demand more efficiently. The notations utilized in our model presentation are defined as follows.

Indices:

r Index of dispatched bus service (each dispatched vehicle is a bus service).

y Index of vehicle type.

s, s' Index of station (origin s to destination s').

Sets:

R Set of bus services, $r \in R$

Y Set of vehicle types, $y \in Y$

S Set of stations, $s, s' \in S,\ s < s'$

Parameters:

c_y Total capacity of a vehicle of type y [pax/veh].

a_y Standing floor area inside a vehicle of type y [m^2].

q_y Number of seats inside a vehicle of type y.

λ^w Waiting time cost [€/h].

λ^v In-vehicle time cost [€/h].

λ^d Driver cost [€/veh-h].

λ^u_y Running cost of a vehicle of type y [€/veh-h].

$F_{s,s'}(t)$ Destination distribution function (exhibiting what percentage of commuters, who enter station s at time t, intend to perform a journey from s to s') [%].

$\psi_s(t)$ Traveler arrival rate function in station s at time t [pax/min].

U_y Maximum resource availability on vehicles of type y [veh].

$\delta_{\min}$ Minimum allowable dispatching headway [min].

$\delta_{\max}$ Maximum allowable dispatching headway [min].

$[\ell_b, \ell_e]$ Planning period beginning and ending at time points of ℓ_b and ℓ_e.

κ Time for opening and closing of doors [s].

ξ_a Average time for alighting (unloading) action [s/pax].

ξ_b Average time for boarding (loading) action [s/pax].

$M^n_{r,s}$ Running time required for bus service r to traverse the distance between station $s-1$ and station s [min].

$M^w_{r,s}$ Stopping time of bus service r at station s for loading and unloading of passengers [min].

$G^w_{r,s,s'}$ Number of commuters (who want to perform a journey from s to s' by means of service r) entering station s during the headway duration between service $r-1$ and service r [pax].

$G^w_{r,s}$ Number of waiting commuters for bus service r at station s [pax].

$G^b_{r,s}$ Loading (boarding) demand for bus service r at station s [pax].

$G^a_{r,s}$ Unloading (alighting) demand for bus service r at station s [pax].

$G^{in}_{r,s}$ Total number of commuters inside bus service r during the journey between station $s-1$ and station s [pax].

Auxiliary variables:

C_r Capacity of bus service r [pax/veh].

A^{snd}_r Standing floor area inside bus service r [m^2].

E^{sit}_r Number of seats inside bus service r.

λ^u_r Running cost of bus service r [€/veh-h].

Table 1
Crowding multiplier values (Source: Tirachini et al., 2017).

Standing density (pax/m^2)	Sitting multiplier	Standing multiplier
0	1.00	1.12
1	1.11	1.25
2	1.23	1.39
3	1.34	1.53
4	1.46	1.66
5	1.57	1.80
6	1.69	1.93

$\pi^{sit}_{r,s}$ Crowding multiplier for sitting passengers inside bus service r when traveling between station $s-1$ and station s.

$\pi^{snd}_{r,s}$ Crowding multiplier for standing passengers inside bus service r when traveling between station $s-1$ and station s.

$M^{a}_{r,s}$ Entrance time of bus service r at station s [min].

$M^{d}_{r,s}$ Leaving time of bus service r from station s, $s \geqslant 2$ [min].

$M^{h}_{r,s}$ Headway duration between service $r-1$ and service r at station s [min].

$G^{sit}_{r,s}$ Number of sitting commuters inside bus service r during the journey between station $s-1$ and station s [pax].

$G^{snd}_{r,s}$ Number of standing commuters inside bus service r during the journey between station $s-1$ and station s [pax].

$G^{v}_{r,s}$ Remaining (active) capacity of service r at station s after performing the alighting phase [pax].

$D^{snd}_{r,s}$ Density of standing passengers inside bus service r during the journey between station $s-1$ and station s [pax/m^2].

Decision variables:

x_{ry} A binary variable that is 1 if the r-th bus service is performed by means of a vehicle of type y; otherwise 0.

B_y Number of type-y vehicles assigned for operations [veh].

$M^{d}_{r,1}$ Dispatching time of bus service r from the starting terminal [min].

Here, we outline the main assumptions and modeling attributes used to formulate the MFBS problem:

- The entrance of commuters at stations is assumed as random, which is a usual and rational assumption in PT systems served by high-frequency services (Dakic et al., 2021; Gkiotsalitis and Cats, 2018; Sadrani et al., 2023).
- There is sufficient capacity available to cover the total passenger demand, which is a pervasive assumption in the stage of operational planning (Vuchic, 2017; Gkiotsalitis and Alesiani, 2019; Gkiotsalitis, 2020; Sadrani et al., 2022a).
- The maximum resource constraints for each vehicle type are specified by service providers based on depot inventories.
- We consider the scheduling of vehicles within a designated planning (simulation) period for a two-directional bus line.
- We distinguish between passengers' time valuations when waiting at stations and when riding inside vehicles.
- We distinguish between standing and sitting passengers' perceptions of crowding disutility (discomfort).

Our objective function is defined for the minimization of total costs, consisting of user costs and operating costs, as presented in Eq. (1). User costs are introduced in the first and second terms, which reflect waiting time costs and crowding-sensitive in-vehicle time costs (as sensitive to on-board comfort levels for users), respectively. Operating costs are introduced in the third and fourth terms, which represent human driver costs and size-sensitive vehicle running costs, respectively. Next, we discuss the functional form of the cost terms and problem constraints.

$$Min\ Z = \sum_{r\in R}\sum_{s\in S}\lambda^{w} G^{w}_{r,s}\frac{M^{h}_{r,s}}{2} + \sum_{r\in R}\sum_{s\in S}\lambda^{v} M^{n}_{r,s}\left(\pi^{sit}_{r,s} G^{sit}_{r,s} + \pi^{snd}_{r,s} G^{snd}_{r,s}\right) + \sum_{r\in R}\sum_{s\in S}\lambda^{d}\left(M^{n}_{r,s}+M^{w}_{r,s}\right) + \sum_{r\in R}\sum_{s\in S}\lambda^{u}_{r}\left(M^{n}_{r,s}+M^{w}_{r,s}\right) \quad (1)$$

s.t.

$$\sum\nolimits_{y\in Y} x_{ry} = 1 \quad \forall r \in R \quad (2)$$

$$\sum\nolimits_{r\in R} x_{ry} = B_y \quad \forall y \in Y \quad (3)$$

$$C_r = \sum\nolimits_{y\in Y} x_{ry}\, c_y \quad \forall r \in R \quad (4)$$

$$A^{snd}_{r} = \sum\nolimits_{y\in Y} x_{ry}\, a_y \quad \forall r \in R \quad (5)$$

$$E^{sit}_{r} = \sum\nolimits_{y\in Y} x_{ry}\, q_y \quad \forall r \in R \quad (6)$$

$$\lambda^{u}_{r} = \sum\nolimits_{y\in Y} x_{ry}\, \lambda^{u}_{y} \quad \forall r \in R \quad (7)$$

$$\delta_{\min} \leqslant M^{d}_{r,1} - M^{d}_{r-1,1} \leqslant \delta_{\max} \quad \forall r \in R-\{1\} \quad (8)$$

$$M^{a}_{r,s} = M^{d}_{r,s-1} + M^{n}_{r,s} \quad \forall r \in R,\ s \in S-\{1\} \quad (9)$$

$$M^{d}_{r,s} = M^{a}_{r,s} + M^{w}_{r,s} \quad \forall r \in R,\ s \in S-\{1\} \quad (10)$$

$$M^{w}_{r,s} = \kappa + G^{a}_{r,s}\,\xi_a + G^{b}_{r,s}\,\xi_b \quad \forall r \in R,\ s \in S \quad (11)$$

$$M^{h}_{r,s} = M^{d}_{r,s} - M^{d}_{r-1,s} \quad \forall r \in R-\{1\},\ s \in S \quad (12)$$

$$G^{w}_{r,s,s'} = \int_{M^{d}_{r-1,s}}^{M^{d}_{r,s}} \psi_s(t).\ F_{s,s'}(t).\ dt \quad \forall r \in R,\ s,s' \in S\ \&\ s' > s \quad (13)$$

$$G^{w}_{r,s} = \sum\nolimits_{s'\in S,\, s'>s} G^{w}_{r,s,s'} \quad \forall r \in R,\ s \in S \quad (14)$$

$$G^{v}_{r,s} = C_r - G^{in}_{r,s} + G^{a}_{r,s} \quad \forall r \in R,\ s \in S \quad (15)$$

$$G^{b}_{r,s} = G^{w}_{r,s} \leqslant G^{v}_{r,s} \quad \forall r \in R,\ s \in S \quad (16)$$

$$G^{a}_{r,s} = \sum\nolimits_{\ell\in S,\, \ell<s} G^{w}_{r,\ell,s} \quad \forall r \in R,\ s \in S \quad (17)$$

$$G^{in}_{r,s} = G^{in}_{r,s-1} - G^{a}_{r,s-1} + G^{b}_{r,s-1} \quad \forall r \in R,\ s \in S \quad (18)$$

$$G^{sit}_{r,s} = \min\left\{G^{in}_{r,s},\ E^{sit}_{r}\right\} \quad \forall r \in R,\ s \in S \quad (19)$$

$$G^{snd}_{r,s} = \max\left\{G^{in}_{r,s} - E^{sit}_{r},\ 0\right\} \quad \forall r \in R,\ s \in S \quad (20)$$

$$D^{snd}_{r,s} = \frac{G^{snd}_{r,s}}{A^{snd}_{r}} \quad \forall r \in R,\ s \in S \quad (21)$$

$$\pi^{sit}_{r,s},\ \pi^{snd}_{r,s} = f\left(D^{snd}_{r,s}\right) \quad \forall r \in R,\ s \in S \quad (22)$$

$$B_y \in \mathbb{Z},\ \ 0 \leqslant B_y \leqslant U_y \quad \forall y \in Y \quad (23)$$

$$M^{d}_{r,1} \geqslant 0 \quad \forall r \in R \quad (24)$$

$$x_{ry} \in \{0,\ 1\} \quad \forall r \in R,\ y \in Y \quad (25)$$

Travelers' waiting times are estimated in the first term of (1), formulated based on the random entering of commuters at stations. In this situation, the average waiting time is approximated as half of the headway (Dakic et al., 2021; Gkiotsalitis and Cats, 2018; Sadrani et al.,

2023). Besides, parameter λ^w is the value of waiting time savings, used to translate time values into cost values. Since passengers perceive waiting at stations as less favorable than riding inside vehicles, the waiting time valuation (parameter λ^w) has been demonstrated to be higher than the in-vehicle time valuation (parameter λ^v), e.g., $\lambda^w = 2\lambda^v$ (Cats et al., 2016; Tirachini and Antoniou, 2020; Sadrani et al., 2022a).

The second term calculates users' in-vehicle time costs, while accounting for users' trip comfort. We discern between standing and sitting passengers' perceptions of crowding discomfort by means of crowding multipliers defined separately for standing and sitting cases (Table 1). As a noteworthy aspect of our model, a microscopic tracking of alighting and boarding volumes at each station enables us to update the occupancy level of each service at each part of a route. This precise framework allows us to evaluate the impacts of crowding on passengers' comfort during their trips.

The third term accounts for human driving costs. The number of driving hours determines the drivers' wages, regardless of the vehicle type being driven. By contrast, the fourth term calculates vehicle running costs (e.g., energy and upkeep expenses) depending on the size of each service (size-sensitive running costs), as operating larger vehicles (such as 18-m long buses) is more expensive than operating smaller ones (12-m long buses). However, larger vehicles offer the advantage of reducing passenger inconvenience caused by crowding. This aspect becomes crucial in optimizing vehicle assignment plans because, in addition to operators' costs, users' costs (such as waiting times and crowding costs) have a significant impact on supply decisions in PT services when minimizing a total cost function (Mohring, 1972; Jara-Díaz and Gschwender, 2003; Tirachini et al., 2014).

In the following, we explain the model's constraints, presented in four categories: (i) vehicle assignment [Eqs. (2) and (3)], (ii) vehicle size-dependent characterization [Eqs. (4)-(7)], (iii) vehicle movement planning [Eqs. (8)-(12)], and (iv) user trip flows [Eqs. (13)-(22)].

In Eq. (2), we determine the vehicle type assigned to each bus service (i.e., to each vehicle dispatching). For example, suppose there are three different vehicle types (A, B, and C), if the 5th bus service is carried out by means of a type-A vehicle, then x_{5A} equals 1 (in this case, Eq. (2) will hold as $\underbrace{x_{5A}}_{1} + \underbrace{x_{5B}}_{0} + \underbrace{x_{5C}}_{0} = 1$). Eq. (3) indicates the number of vehicles of each type assigned for operations.

Eqs. (4)-(7) are specifically designed to handle size-sensitive parameters on a service-to-service resolution. This is crucial because the MFBS problem involves services with varying sizes, where size-sensitive attributes (such as on-board capacity) differ for each service based on its size characteristics. For instance, the capacity of service r (performed by means of a 12-m long vehicle) differs from that of service $r + 1$ (performed by means of an 18-m long vehicle). Given the changing patterns of passenger flows over time and place, such modeling components allow for the introduction of best vehicle assignment and dispatching plans in the MFBS problem, resulting in a better capacity adjustment to passenger demand.

Constraint (8) ensures that dispatching headways remain within the permitted range established by policymakers. The dispatching times of services from the first terminal are defined as decision variables in our problem, allowing for the optimization of dispatching intervals between services according to temporal fluctuations in passenger requests. Besides, it is assumed that the dispatching of the first service is performed at the start of the planned period (i.e., $M^d_{1,1} = \ell_b$).

As presented in Eq. (9), to determine the entry time of a service at a station, the leaving time of that service from the earlier station would be added to the time spent riding between stations. Eq. (10) indicates that a service will leave a station after passengers' unloading and loading events at that station (dwelling time). As shown in Eq. (11), the dwell time is estimated in view of the time consumed for unloading and loading actions by passengers, as well as the time associated with opening and closing doors. In Eq. (12), the inter-departure headway is computed as the elapsed time between the leavings of two successive services from a designated station.

Considering time-varying arrival rates of travelers at stations, Eq. (13) computes the number of travelers (intending to carry out a journey from s to s' by means of service r) who enter station s during the elapsed headway between service $r - 1$ and service r. Given the notions of $G^w_{r,s,s'}$ and $G^w_{r,s}$, Eq. (14) becomes apparent in our model, meaning that the aggregate of all travelers waiting at station s, whose journey destinations can differ from each other (journey from s to s', $s' > s$), yields the total traveler volume waiting for service r at station s.

Eq. (15) updates the residual capacity inside each service, taking into account the passenger unloading process that occurs at each station. Constraint (16) ensures that the operational plans for service supply (including vehicle assignment and dispatching plans) are sufficient to satisfy passenger demands.

Eq. (17) calculates the unloading volume of travelers from service r at station s, considering the commuters who boarded service r at former stations in order to carry out a journey to station s. Considering passengers' unloading and loading events at each station, Eq. (18) is employed to update the passenger load (occupancy level) of each service in every section of the route. Eqs. (19) and (20) are used to compute the number of travelers sitting and standing inside service r during the journey between stations $s-1$ and s. Eq. (21) determines the density of standing passengers inside services, used as a proxy for representing the degree of crowding inside services (Tirachini et al., 2017). Accordingly, crowding multipliers for standing and sitting situations are determined as a function of the standing density, as stated in Eq. (22). In our study, crowding multipliers are given in Table 1, taken from Tirachini et al. (2017). For instance, a multiplier of 1.93 means that, on average, travel time savings when standing with a density of 6 standees per square meter are valued almost doubled than travel time savings when sitting without any passenger standing. Ultimately, constraints (23)-(25) exhibit the scope of the problem's decision variables.

As depicted in Fig. 2, our proposed problem aims to optimize three decision variables in mixed-fleet bus operations: (i) vehicle assignment (integer decision variable), (ii) dispatching sequence of vehicles (binary decision variable), and (iii) dispatching times of vehicles (continuous decision variable). In vehicle assignment optimization, the problem determines the number of vehicles of each type assigned for mixed-fleet operations, considering resource constraints on vehicle availability. Specifically, in a mixed-fleet bus operation, the running costs of buses are influenced by their size, with larger buses (e.g., 18-m long) incurring higher costs than smaller ones (e.g., 12-m long). However, larger buses (with larger capacities) can reduce in-vehicle crowding discomfort for passengers. Therefore, the proper choice of vehicle types in the fleet requires an in-depth cost analysis encompassing both user (demand) and operator (supply) aspects. In vehicle dispatching optimization, the problem determines both the best dispatching sequence and dispatching time for each vehicle.

The presence of both types of integer and continuous decision variables, along with the nonlinear complexity within the problem formulation, results in a MINLP model, which encompasses the complex aspects of optimizing integer variables and handling nonlinear terms. Additionally, the high combinatorial complexity of the problem regarding vehicle dispatching sequence solutions, which represent permutation-based solutions, is further discussed in Section 3.

Our formulation contains nonlinear terms in both the objective function and constraints to achieve a more realistic modeling and representation of real-world conditions in urban bus operations. Specifically, the nonlinearity stems from the product of variables needed for modeling crowding discomfort effects, a notable contribution to the literature, as this is the first time these effects are modeled in the scheduling of mixed-fleet bus operations. Therefore, despite the need to address nonlinear complexity, we decided to keep the nonlinear nature of our problem formulation because it ensures a more accurate reflection

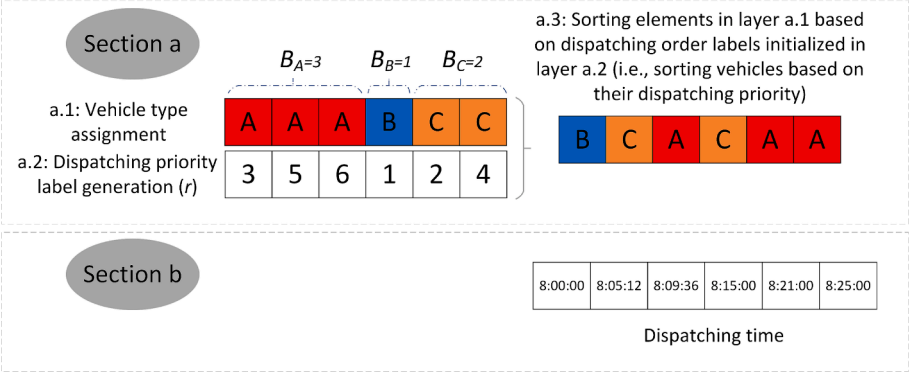


Fig. 3. Solution encoding example.

Initialize population size (N_{pop}), maximum number of iterations ($MaxIt$), crossover rate (P_c), mutation rate (P_m)

Generate an initial population of N_{pop} chromosomes
Evaluate the fitness of each chromosome in the population by Eq. (26)
Set iteration counter $t = 1$
while ($t < MaxIt$)
  Select a pair of chromosomes (parents) from the current population using roulette wheel selection
  Apply crossover to the selected parents to create new offspring
  Select a chromosome using a random selection scheme
  Apply mutation to the selected chromosome to create new mutant offspring
  Evaluate the fitness of the generated offspring by Eq. (26)
  Merge the current population and offspring population (obtained from crossover and mutation)
  Sort the individuals based on their fitness
  Select the best (top) N_{pop} individuals for the next generation
  Increase the current iteration by 1, $t = t + 1$.
end

Fig. 4. Pseudo code of the GA.

of practical scenarios.

For example, the second term of the objective function in Eq. (1) (a nonlinear term) models crowding discomfort costs for passengers while distinguishing between the perceptions of sitting and standing passengers, as it has been shown that standing passengers feel more dissatisfied with crowding conditions inside vehicles than sitting passengers (Wardman and Whelan, 2011; Tirachini et al., 2017). Another example is the nonlinear terms in constraints (19)-(21), which are required for the microscopic (high-resolution) modeling and updating of the number of sitting and standing passengers and crowding levels inside each bus at each trip segment (between every two consecutive stations).

3. Solution algorithms

Optimizing vehicle dispatching sequences in MFBS problems is notably challenging due to the combinatorial nature of these sequence-based solutions. Even in their simplest forms, these problems have been demonstrated to be NP-hard (Sadrani et al., 2022a). As the number of vehicles increases, the computational requirements of solving this sequence-dependent combinatorial optimization problem grow significantly. This increased complexity stems from the need to generate distinct dispatching sequences while accounting for the similarity among buses of the same type, resulting in permutations with repetition.

It is evident that exhaustively evaluating all potential solutions through full enumeration becomes computationally infeasible for large instances. In this context, employing metaheuristics can present a promising approach to effectively address the MFBS problem. Next, we describe our proposed metaheuristics, including GA, GWO, GA-SA, and GWO-SA.

3.1. Solution representation

The process of introducing solutions into algorithms is made possible by solution representation, or encoding, which is a critical step in the

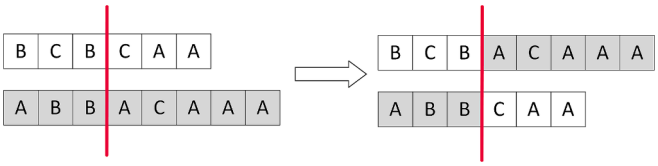


Fig. 5. Example of single-point crossover.

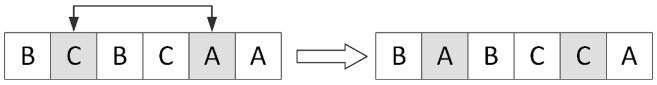


Fig. 6. Example of swapping mutation operator.

Initialize the grey wolf population X_i $(i = 1, 2, \ldots, n)$
Initialize a, A, and C
Evaluate the fitness of each search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
while $(t < MaxIt)$
  for each search agent
    Update the position of the current search agent by Eq. (34)
  end
  Update a, A, and C
  Evaluate the fitness of all search agents
  Update X_α, X_β, and X_δ
  $t = t + 1$
end
Return X_α

Fig. 7. Pseudo code of the GWO algorithm (Mirjalili et al., 2014).

Vehicle ID	A	A	A	B	C	C
Random numbers *U [0, 1]*	0.36	0.22	0.72	0.91	0.14	0.59

Sorted dispatching sequence	C	A	A	C	A	B
Sorted numbers	0.14	0.22	0.36	0.59	0.72	0.91

Fig. 8. Example of the random-key (RK) encoding method.

application of metaheuristics. We offer a two-section structure for the solution box representation for handling the MFBS problem. This structure, illustrated in Fig. 3, consists of an integer-coded segment (section a) and a continuous (real)-coded section (section b).

The first encoding segment (section a) characterizes vehicle assignment plans. It consists of two sublayers: a.1, representing the number of vehicles of each type assigned for operations, and a.2, representing the dispatching sequence labels of the vehicles. Specifically, in sublayer a.1, non-negative integer values are randomly generated for each vehicle type, considering the maximum availability of each vehicle type. To assign a service priority label to each vehicle, a random permutation of integers from 1 to the total number of assigned vehicles is created in sublayer a.2. This permutation represents the dispatching sequence of the vehicles from the first terminal. The combination of sublayers a.1 and a.2 results in layer a.3, which presents the sorted dispatching sequence of the vehicles. In the second encoding segment (section b), we present the encoding of dispatching times using real-coded values.

3.2. Genetic algorithm (GA)

The GA is a popular evolutionary algorithm, in which a population of solutions is handled and improved based on the ideas of evolution and natural selection (Boussaïd et al., 2013; Katoch et al., 2021). Particularly, the GA has demonstrated a highly prominent performance in solving a broad spectrum of complex combinatorial optimization problems (for a review of metaheuristic applications in the realm of combinatorial problems, see Peres and Castelli, 2021). Each potential solution in the GA is represented as a chromosome (individual) with a number of genes describing the decision variables of the problem. The GA employs three main operators (selection, crossover, and mutation) to create new chromosomes (offspring). The search process conducted by the GA is depicted in Fig. 4.

To measure the fitness (desirability) of solutions in the evaluation phase, we consider a fitness function as follows:

$$F(x) = \frac{\text{Zmax} - Z(x)}{\text{Zmax} - \text{Zmin}} \tag{26}$$

where $F(x)$ represents the scaled fitness score of individual x ,$Z(x)$ is the objective function value (OFV) of individual x , as well as Zmax and Zmin are the OFVs of the worst and best solutions in the current population, respectively. According to Eq. (25), in our study, which involves a minimization problem, solutions with lower cost values are considered more favorable and receive higher fitness scores.

To boost population diversity, new chromosomes (offspring) are created by means of crossover and mutation operators applied to a number of chromosomes being selected as parents. For the parent selection phase, we employ a roulette wheel selection scheme, under which superior solutions (with higher fitness values) have more chances of being chosen as parents (Akbari et al., 2017; Dulebenets et al., 2018; Goli et al., 2021).

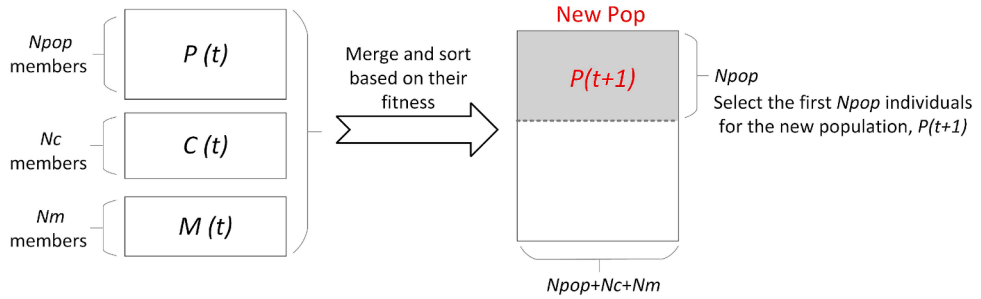
The crossover application leads to the exchange of genetic material between two parents. As shown in Fig. 5, we utilize a single-point crossover operator for the integer-coded sections of vehicle assignment solutions. However, for the vehicles' dispatching times in section b (Fig. 3), we employ an arithmetic crossover operator, which is a popular operator for increasing the diversity of real-coded values in the GA (Mirjalili et al., 2020; Katoch et al., 2021):

$$\begin{aligned} \text{Child1} &= m * \text{Parent1} + (1 - m) * \text{Parent2} \\ \text{Child2} &= (1 - m) * \text{Parent1} + m * \text{Parent2} \end{aligned} \tag{27}$$

where m is a random weighting vector generated for each crossover operation.

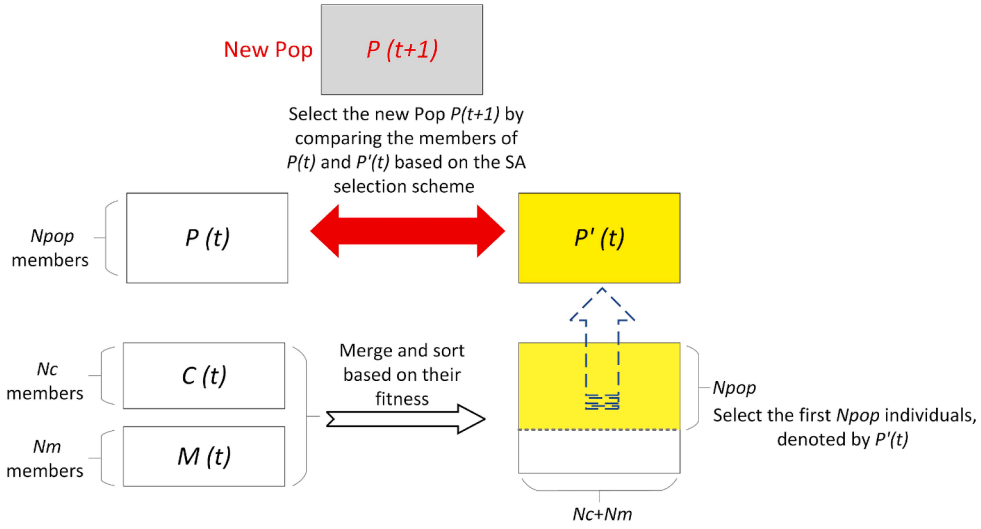
Notably, our crossover strategy allows for gene exchange between parents of different sizes (lengths) (Fig. 5), facilitating the exploration of a wider search space. This approach enhances solutions' diversity regarding vehicle assignment (resource allocation) and dispatching order in the MFBS problem. The length of a solution depends on the total number of vehicles assigned for operations. During the crossover process, the feasibility of the resulting offspring is checked regarding resource constraints, ensuring that the number of vehicles of each type assigned for operational activities does not exceed the available resources. If an offspring is found to be infeasible, a modification process is executed to correct it. In such cases, random corrections are made to the older parts of the offspring, while the genetic material moved through the crossover remains unchanged. Besides, Fig. 6 depicts an example of a swapping mutation, which involves randomly selecting two elements and interchanging their positions.

As for the next generation, the members of the current population and new offspring (obtained from crossover and mutation operators) are



* *P(t)*: Current population, *C(t)*: Crossover offspring population, *M(t)*: Mutation offspring population

(a) GA



(b) GA-SA

Fig. 9. Selection of individuals for the next generation in the GA vs. GA-SA.

Initialize population size (N_{pop}), maximum number of iterations ($MaxIt$), maximum number of sub-iterations ($MaxsubIt$), crossover rate (P_c), mutation rate (P_m), initial temperature (T_0), cooling rate (α)

Generate the initial population of chromosomes Y_i $(i = 1, 2, \ldots, N_{pop})$
Set iteration counter $t = 1$
Evaluate the fitness value of each chromosome

while $(t < MaxIt)$
  while $(Cool\_iteration < MaxSubIt)$

    Select parents and apply crossover and mutation phases (in the same manner as the GA)

    Merge offspring created from crossover and mutation phases (offspring population)

    Create the next generation by performing pairwise comparisons between the members in the offspring population (y') and those in the current population (y) based on a SA rule (see Fig. 9 for a better view of this step):

    For each pairwise comparison using a SA strategy:

    Compute Δ= Cost (y') − Cost (y)
      If $\Delta < 0$ **then**
        $y = y'$
      Else
        Compute $P = e^{\frac{-\Delta}{T}}$
        If r = random (0,1) $\leq P$ **then**
          $y = y'$
        end
      end
    Update and store the cost of the best solution
    $Cool\_iteration = Cool\_iteration + 1$
      Reduce the temperature $T = T \times \alpha$
  end
  $t = t + 1$
end

Fig. 10. Pseudo code of the GA-SA.

merged and arranged according to their fitness scores. Then, the top N_{pop} individuals (equivalent to the population size) are chosen for the next generation.

3.3. Grey wolf Optimizer (GWO)

The GWO, introduced by Mirjalili et al. (2014), is a nature-inspired swarm intelligence algorithm that mimics the social hierarchy and hunting behavior of grey wolves. It considers four kinds of grey wolves for the simulation of leadership hierarchy: alpha, beta, delta, and omega. In essence, the alpha wolf (α) refers to the best (most important) individual in a grey wolf pack, known as the group leader. Besides, the second and third top individuals are known as beta (β) and delta wolves (δ), respectively. Other individuals are known as omega (ω) wolves, essentially steered by the α, β, and δ wolves during the search for prey. The hunting phase consists of three steps: encircling, hunting, and attacking steps.

In the encircling phase, the GWO simulates the motions of grey wolves encircling prey at the beginning of the hunting process. Mathematically speaking, such motions are modeled using the following equations:

$$\vec{D} = \left| \vec{C} \,.\, \vec{X}_p\,(t) - \vec{X}(t) \right| \tag{28}$$

$$\vec{X}(t+1) = \vec{X}_p\,(t) - \vec{A} \,.\, \vec{D} \tag{29}$$

$$\vec{A} = 2\,\vec{a} \,.\, \vec{r_1} - \vec{a} \tag{30}$$

$$\vec{C} = 2 \,.\, \vec{r_2} \tag{31}$$

where t refers to the current iteration, $\vec{X}_p$ refers to the location of the prey, and $\vec{X}$ refers to the location of a wolf. $\vec{A}$ and $\vec{C}$ are coefficient vectors allowing for the movement (relocation) of wolves at different positions around the prey. In essence, D reflects the distance between a wolf and a prey. Elements $\vec{a}$ will be reduced linearly from 2 to 0 as iteration progresses. Besides, $\vec{r_1}$ and $\vec{r_2}$ refer to random vectors within the range of [0, 1] (Mirjalili et al., 2014).

In the hunting phase, the locations of omega (ω) wolves will be updated based on the directions of the α, β, and δ wolves (which are the three best solutions identified so far). In essence, this idea stems from the fact that the three best search agents have a better overview (knowledge) about the location of the prey, and therefore other search agents (other solutions) should update their locations accordingly (see Fig. 7 for more details on the pseudo code of the GWO). Such a concept is mathematically modeled as follows:

Initialize the grey wolf population X_i $(i = 1, 2, \ldots, n)$
Initialize a, A, and C
Evaluate the fitness of each search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent

while $(t < MaxIt)$
  while $(Cool\_iteration < MaxSubIt)$
    Update the position of each omega search agent by Eq. (34)
    Update a, A, and C
    Evaluate the fitness of all omega search agents and select the three best ones in the omega population
    Update X_α, X_β, and X_δ based on a SA strategy, as described below:
    Comparing the current main members (alpha, beta, and delta) (denoted by y) with the three best agents identified in the omega population (denoted by y') based on a SA rule:
    For each pairwise comparison using a SA strategy:
    Compute Δ= Cost (y') − Cost (y)
      If $\Delta < 0$ **then**
        $y = y'$
      Else
        Compute $P = e^{\frac{-\Delta}{T}}$
        If r = random (0,1) $\leq P$ **then**
          $y = y'$
        end
      end
    Update and store the best solution
    $Cool\_iteration = Cool\_iteration + 1$
    Reduce the temperature $T = T \times \alpha$
  end
  $t = t + 1$
end

Fig. 11. Pseudo code of the GWO-SA.

$$\begin{aligned}\overrightarrow{D_\alpha} &= \left|\overrightarrow{C}_1 . \overrightarrow{X}_\alpha - \overrightarrow{X}\right|,\\ \overrightarrow{D_\beta} &= \left|\overrightarrow{C}_2 . \overrightarrow{X}_\beta - \overrightarrow{X}\right|,\\ \overrightarrow{D_\delta} &= \left|\overrightarrow{C}_3 . \overrightarrow{X}_\delta - \overrightarrow{X}\right|\end{aligned} \tag{32}$$

$$\overrightarrow{X}_1 = \overrightarrow{X}_\alpha - \overrightarrow{A}_1 . \overrightarrow{D_\alpha}, \quad \overrightarrow{X}_2 = \overrightarrow{X}_\beta - \overrightarrow{A}_2 . \overrightarrow{D_\beta}, \quad \overrightarrow{X}_3 = \overrightarrow{X}_\delta - \overrightarrow{A}_3 . \overrightarrow{D_\delta} \tag{33}$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X}_1 + \overrightarrow{X}_2 + \overrightarrow{X}_3}{3} \tag{34}$$

In the stage of attacking a prey, the GWO mimics the motions of grey wolves as they attempt to approach prey. This stage aims to create a better balance between the exploration and exploitation phases. Accordingly, when $\left|\overrightarrow{A}\right| > 1$, search agents diverge from each other to discover a better prey, thus enhancing the GWO's ability for exploration. By contrast, when $\left|\overrightarrow{A}\right| < 1$, the grey wolves are directed towards the prey, thus enhancing the GWO's ability for exploitation. It should be noted that $\overrightarrow{A}$ can vary within the range of [-2, 2], depending on parameter $\overrightarrow{a}$ reduced from 2 to 0.

Given that the GWO is a continuous algorithm, an appropriate encoding procedure is required to encode solutions in a manner that adapts this algorithm to the discrete search region of the proposed MFBS problem. In this regard, we utilize a Random-Key (RK) method as our chosen encoding scheme, widely recognized for its effectiveness in adapting continuous metaheuristics, such as GWO, to problems with discrete search spaces (Mirjalili and Lewis, 2013; Beheshti, 2021; Goodarzian et al., 2021; Nayeri et al., 2022). Specifically, the RK method has been broadly applied in permutation-based optimization problems, such as job shop scheduling and traveling salesman problems, demonstrating its successful applications in these domains (Fathollahi-Fard et al., 2020a, 2020b; Yu et al., 2020; Nayeri et al., 2022).

Our RK encoding scheme has two stages, as shown in Fig. 8. Initially, a vector with a length equal to the number of vehicles assigned for operations is created using a uniform distribution U [0, 1], i.e., each element in the vector represents a random number within the range [0, 1]. Subsequently, the random numbers in the vector are arranged in ascending order to introduce the new sequence in which the vehicles will be dispatched. For example, the encoded solution in Fig. 8 yields the dispatching sequence: C, A, A, C, A, B. By utilizing the RK method, we

Table 2
Parameter calibration of the proposed algorithms.

Algorithm	Parameters	Levels			Best levels (based on the results in Fig. 12)
		1	2	3	
GA	Maximum iterations (*MaxIt*)	100	150	200	1
	Population size (N_{pop})	30	50	70	3
	Crossover rate (P_c)	0.6	0.7	0.8	3
	Mutation rate (P_m)	0.2	0.3	0.4	3
GWO	Maximum iterations (*MaxIt*)	100	150	200	1
	Grey wolf pack size (N_{pop})	30	60	90	3
GA-SA	Maximum iterations (*MaxIt*)	100	150	200	2
	Population size (N_{pop})	30	50	70	2
	Crossover rate (P_c)	0.6	0.7	0.8	3
	Mutation rate (P_m)	0.2	0.3	0.4	2
	Cooling rate (*Alpha*)	0.90	0.95	0.99	1
	Initial temperature (T_0)	10	15	20	3
GWO-SA	Maximum iterations (*MaxIt*)	100	150	200	1
	Grey wolf pack size (N_{pop})	30	60	90	3
	Cooling rate (*Alpha*)	0.90	0.95	0.99	3
	Initial temperature (T_0)	10	15	20	3

adapt the discrete dispatching sequence into a continuous representation that can be processed by the GWO, enabling the optimization process within the MFBS problem.

3.4. Hybrid of GA and SA (GA-SA)

In this part, we introduce a hybrid algorithm, GA-SA, which amalgamates the GA and SA approaches. In essence, the hybridization involves incorporating the SA strategy into the GA's selection process for the next generation of individuals. Accordingly, the proposed GA-SA can provide an opportunity for the acceptance of non-improving solutions for the next generation based on the SA strategy, instead of relying merely on a ranking-based selection of the best solutions (see Fig. 9 for an illustration of this integration). The details of the GA-SA algorithm are shown in Fig. 10.

It is worth noting that there are other versions of GA, such as biased random-key GAs (BRKGAs), which incorporate diversity by allowing non-elite solutions to be selected as parents (for a recent review of BRKGAs, see Londe et al., 2024). BRKGAs maintain an elite set and a non-elite set, with a bias towards selecting parents from the elite set, and use a biased crossover with offspring inheriting more traits from the elite parent. However, BRKGAs focus on maintaining diversity and convergence without probabilistic acceptance of non-improving solutions.

In contrast, our GA-SA algorithm uses the SA strategy to probabilistically accept non-improving solutions based on temperature, promoting exploration in early iterations and shifting towards exploitation as temperature decreases. This adaptive acceptance helps escape local optima and enhances search capabilities. Moreover, BRKGAs represent solutions as random key vectors (real numbers between 0 and 1), decoded into feasible solutions using a decoding function, whereas our GA-SA directly encodes solutions as binary strings, permutations, or other problem-specific formats without requiring an additional decoding step.

3.5. Hybrid of GWO and SA (GWO-SA)

In this section, we develop a hybrid GWO-SA algorithm that combines the convergence capabilities of GWO with the diversity-maintaining capabilities of SA. Specifically, the hybrid algorithm uses the acceptance probability function of SA to update the positions of the alpha, beta, and delta wolves by comparing them with the best three omega search agents identified in each iteration. For example, if an omega wolf exhibits better fitness than the current alpha wolf, it replaces the alpha wolf; otherwise, the SA strategy determines whether the alpha wolf is replaced. The steps of the proposed GWO-SA algorithm are presented in Fig. 11.

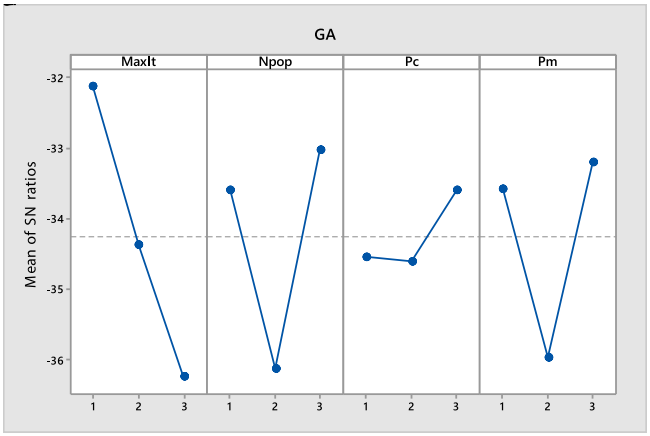
In this study, we leveraged the GWO and combined it with the SA algorithm (GWO-SA) to further enhance search power and exploitation capabilities for the proposed MFBS problem. Our intention was not to introduce a new metaheuristic—an issue highlighted by researchers against the introduction of excessive new metaphor-based metaheuristics without substantial innovation (see Aranha et al., 2022)—but to explore the performance of GWO within a new application context.

Moreover, a notable advantage of the GWO is its low number of controlling parameters (the number of iterations and grey wolf pack size), reducing the need for extensive parameter tuning (Faris et al., 2018; Wang et al., 2022). While the GWO has been successfully applied in diverse engineering fields (for a comprehensive review of GWO applications, see Sharma et al., 2022; Liu et al., 2024), its use in transport planning problems is scarce (for review papers on PT planning, see Liu et al., 2021; Gkiotsalitis et al., 2022; and Durán-Micco and Vansteenwegen, 2022). This motivated us to explore the GWO's performance in the context of mixed-fleet bus scheduling.

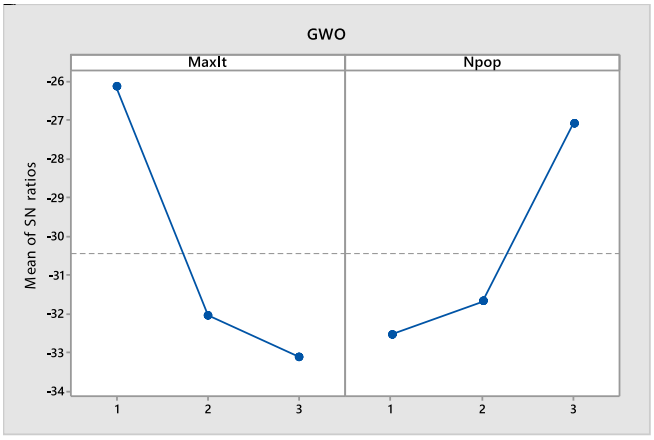
Our testing revealed that population-based algorithms, which offer greater exploration capabilities, outperformed single-solution algorithms for this problem. For example, in the GWO algorithm, the updating mechanism involves solutions being randomly moved towards three leading solutions (alpha, beta, and delta wolves), representing the best solutions in the grey wolf pack. This strategic guidance towards multiple leading solutions enhances the algorithm's capability to avoid local optima and improve solution diversity. For a comprehensive template for metaheuristic comparison, we refer to de Armas et al. (2022), proposing a methodology to compare metaheuristics by categorizing their components, aiming to identify similarities and novelties among them. Moreover, to further enhance exploitation capabilities during the search process, we integrated SA with GA and GWO, resulting in the development of GA-SA and GWO-SA hybrids, both of which showed superior solution quality than GWO.

4. Computational experiments

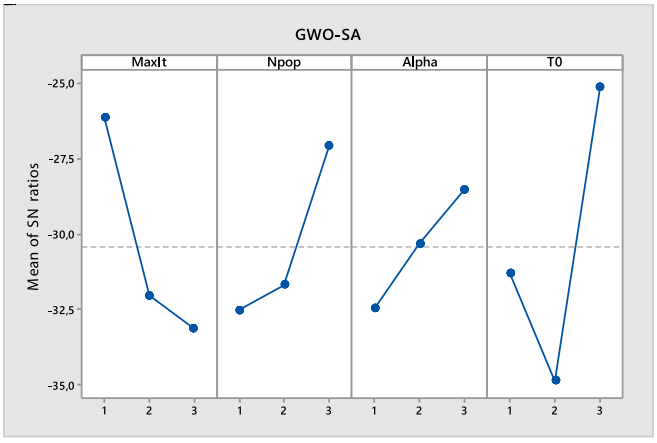
In this section, we suggest a Taguchi approach to fine-tune the parameters of the metaheuristics. We also perform several numerical experiments to compare the metaheuristics' capabilities based on two crucial metrics: solution quality and CPU time.



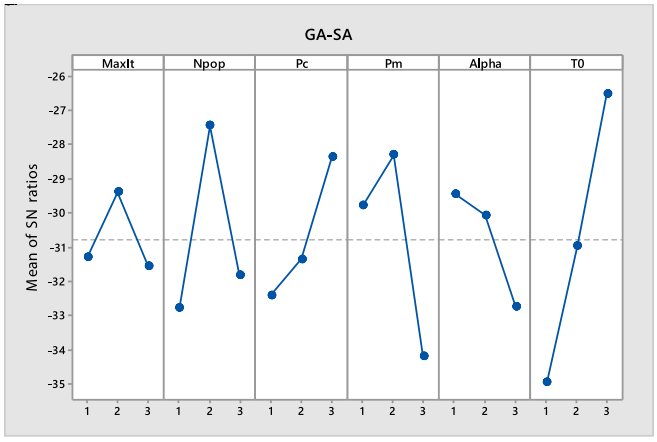
(a) GA



(b) GWO



(d) GWO-SA



(c) GA-SA

Fig. 12. Results of the Taguchi method for adjusting metaheuristics' parameters.

Table 3
Small (S) and medium (M) test instance results.

Instance ID	GAMS		GA			GWO			GA-SA			GWO-SA			GAP (%) relative to GAMS			
	OFV	CPUT*	Avg. OFV	SD. OFV	CPUT	Avg. OFV	SD. OFV	CPUT	Avg. OFV	SD. OFV	CPUT	Avg. OFV	SD. OFV	CPUT	GA	GWO	GA-SA	GWO-SA
S1	270.69	532.06	270.69	0.00	21.24	270.69	0.00	20.63	270.69	0.00	28.87	270.69	0.00	26.48	0.00	0.00	0.00	0.00
S2	276.52	607.33	276.52	0.00	29.70	276.52	0.00	26.58	276.52	0.00	29.57	276.52	0.00	27.34	0.00	0.00	0.00	0.00
S3	356.64	1516.65	357.66	0.42	32.31	356.64	0.00	26.64	356.64	0.00	38.75	356.64	0.00	35.78	0.28	0.00	0.00	0.00
S4	370.90	1676.98	370.90	0.00	33.54	370.90	0.00	30.50	370.90	0.00	42.20	370.9	0.00	39.03	0.00	0.00	0.00	0.00
S5	405.94	3619.97	414.18	3.25	38.91	410.87	0.91	36.54	409.39	0.47	47.52	405.94	0.00	44.03	2.02	1.21	0.84	0.00
S6	424.80	4073.96	433.18	3.53	46.05	430.20	0.68	41.82	427.85	0.63	52.10	424.8	0.00	48.09	1.97	1.27	0.71	0.00
S7	440.27	5159.58	450.43	4.01	54.62	446.56	0.67	42.36	444.20	0.45	61.94	440.27	0.00	56.87	2.30	1.42	0.89	0.00
S8	468.42	6241.24	479.90	4.14	53.96	475.02	0.72	40.45	473.13	0.54	62.59	468.42	0.00	57.43	2.45	1.40	1.00	0.00
S9	484.99	7358.86	496.32	3.85	57.68	491.52	0.91	46.33	489.05	0.27	66.96	484.99	0.00	61.16	2.33	1.34	0.83	0.00
S10	518.98	8006.62	530.62	3.22	66.40	526.38	0.86	46.83	522.13	0.19	77.27	518.98	0.00	70.47	2.24	1.42	0.60	0.00
M1	539.40	9088.34	550.12	4.06	68.16	539.40	0.00	54.52	539.40	0.00	84.77	539.40	0.00	77.36	1.98	0.00	0.00	0.00
M2	561.97	10338.05	584.31	4.48	70.46	572.54	3.31	56.05	567.12	3.17	85.43	564.26	0.70	77.97	3.97	1.88	0.91	0.40
M3	590.05	— **	610.10	3.61	68.54	599.22	2.19	59.13	596.61	2.63	86.14	590.05	0.00	78.65	3.39	1.55	1.11	0.00
M4	619.57	—	639.24	3.67	73.21	629.19	0.00	60.95	619.57	3.28	88.28	619.57	0.00	80.97	3.17	1.55	0.00	0.00
M5	621.60	—	640.14	4.00	86.56	632.35	2.45	65.90	628.10	2.50	100.91	623.43	0.45	92.41	2.98	1.72	1.04	0.29
M6	660.51	—	680.41	3.46	88.41	670.20	2.34	69.75	668.22	2.07	104.09	662.12	0.73	95.26	3.01	1.46	1.16	0.24
M7	765.85	—	787.11	2.67	92.18	779.41	3.06	67.81	774.14	2.47	104.97	768.20	0.71	96.06	2.77	1.77	1.08	0.30
M8	755.14	—	774.26	2.81	100.10	763.34	3.14	79.27	763.52	1.87	119.46	758.26	0.87	109.01	2.53	1.08	1.10	0.41
M9	769.95	—	789.15	3.01	105.13	778.61	2.06	88.91	777.15	1.27	130.89	771.43	0.40	119.51	2.49	1.12	0.93	0.19
M10	788.07	—	808.30	2.71	118.27	798.10	2.31	89.87	796.56	2.54	137.46	790.37	0.51	125.32	2.56	1.27	1.07	0.29

* The acronym CPUT corresponds to CPU time, measured in seconds.
** A hyphen is used to indicate instances where GAMS's CPU times exceed 3 h (10800 s).

4.1. Parameter tuning

It is well known that the performance of metaheuristics is sensitive to the adjustment of their control parameters (Mirjalili et al., 2020). Various methods have been employed in the literature for fine-tuning the parameters of metaheuristics, such as the Response Surface Methodology (RSM) (Assadi and Bagheri, 2016; Nourmohammadzadeh and Voß, 2022; Sadrani et al., 2023), the Iterated Racing (Irace) method (López-Ibáñez et al., 2016; Leon-Blanco et al., 2022), and the Taguchi method (Fathollahi-Fard et al., 2021; Ala et al., 2022; Goli et al., 2023; Hashemi-Amiri et al., 2023; Rahmanifar et al., 2023; Nayeri et al., 2022; Tirkolaee et al., 2022b, 2023). In this work, we utilize the Taguchi method. By introducing orthogonal arrays, the Taguchi method reduces the number of trials required to identify the most favorable combination of parameter levels.

Two kinds of factors are taken into account by the Taguchi method: controllable and noise (uncontrollable) factors. It seeks to identify the best values of controllable factors while lessening the impact of noise sources. The Signal-to-Noise (S/N) ratio was proposed by Taguchi to evaluate the fluctuation in the response variable. In essence, signal (S) and noise (N) represent the response variable (desirable value) and standard deviation (undesirable value), respectively. Therefore, the objective is to maximize the S/N ratio:

$$\frac{S}{N} = -10 \times \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \tag{35}$$

where n and y_i reflect the number of orthogonal arrays and the response in replication i, respectively.

Table 2 presents the parameters for each algorithm, with three levels considered for each parameter. To implement the Taguchi designs, we utilize Minitab 17 software and adopt a three-level Taguchi strategy. For example, consider an algorithm with four parameters, e.g., the GA with the maximum number of iterations, population size, crossover rate, and mutation rate. Under a three-level Taguchi strategy, three different levels (values) are considered and tested for each parameter (see Table 2). In this case (four parameters at three levels), the Taguchi method offers the L9 orthogonal array to design the experiments, resulting in a total of 9 trials. Table A.1 (in Appendix A) details the L9 orthogonal array and parameter combinations for each trial. The results of the Taguchi method are depicted in Fig. 12, where the best level for each parameter is determined by the highest mean of the S/N ratio, as reported in Table 2.

4.2. Test samples for the assessment of metaheuristics

This section presents a computational experiment that employs random test instances of varying sizes to examine the validity and efficiency of the proposed metaheuristics. To this end, 30 randomly generated test problems, containing small, medium, and large samples, are designed (the characteristics of the test samples are given in Appendix B).

For small- and medium-scale instances, which can be solved by GAMS software optimally, we compare the results of the four metaheuristics (GA, GWO, GA-SA, and GWO-SA) with optimal results achieved by GAMS software using the BARON optimization solver. Besides, all computational programs have been conducted on a PC with an Intel (R) Core(TM) i5-6500 CPU operating at 3.20 GHz and 16.0 GB of RAM. To assess the solution quality of the proposed metaheuristics, we measure the gap as follows:

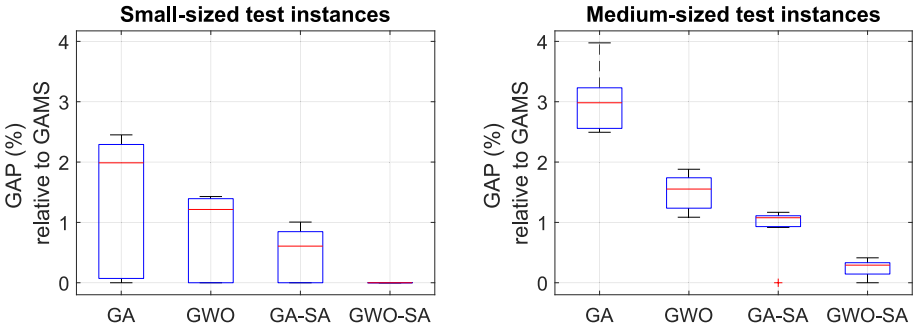
$$GAP = \frac{Heur_{sol} - GAMS_{sol}}{GAMS_{sol}} \times 100 \tag{36}$$

where $GAMS_{sol}$ is the optimal solution achieved by GAMS, and $Heur_{sol}$ is the solution found by the selected metaheuristic.

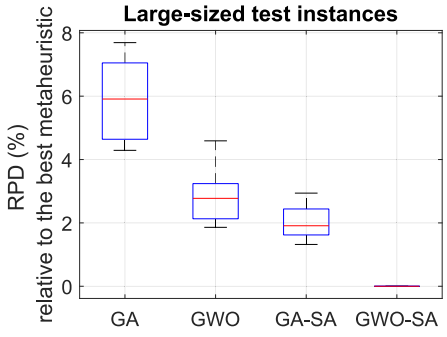
Each metaheuristic is executed in 10 independent runs. The results,

Table 4
Large (L) test instance results.

Instance ID	GA			GWO			GA-SA			GWO-SA			RPD (%) relative to the best metaheuristic			
	Avg. OFV	SD. OFV	CPUT	Avg. OFV	SD. OFV	CPUT	Avg. OFV	SD. OFV	CPUT	Avg. OFV	SD. OFV	CPUT	GA	GWO	GA-SA	GWO-SA
L1	917.33	9.34	122.81	907.52	7.42	92.73	902.84	3.67	144.02	879.59	0.00	131.13	4.29	3.17	1.61	0.00
L2	1016.21	7.62	123.17	997.04	5.69	95.74	994.09	3.44	149.08	972.49	0.00	135.55	4.50	2.44	2.13	0.00
L3	1203.86	9.18	138.90	1167.15	7.71	109.19	1162.69	4.32	166.19	1145.40	4.11	151.12	5.10	2.02	1.62	0.00
L4	1372.64	7.83	143.75	1332.11	6.63	113.59	1326.60	3.85	170.83	1311.75	3.74	155.36	4.64	1.86	1.32	0.00
L5	1908.32	7.58	154.35	1837.92	6.17	124.21	1829.11	4.14	184.02	1773.56	3.23	168.07	7.60	3.49	2.94	0.00
L6	2182.44	8.69	171.28	2105.04	7.23	132.12	2093.28	4.47	202.87	2038.12	3.84	185.44	7.05	3.24	2.62	0.00
L7	2815.78	8.26	187.02	2718.95	6.94	147.02	2699.58	3.63	223.05	2651.88	3.11	204.19	6.13	2.44	1.82	0.00
L8	2605.44	8.15	193.77	2516.83	6.86	150.78	2503.97	3.84	230.92	2463.72	3.76	211.60	5.69	2.13	1.85	0.00
L9	2840.85	8.74	210.32	2745.61	7.19	167.96	2721.60	5.11	250.07	2658.29	3.96	228.77	6.88	3.11	1.97	0.00
L10	3127.48	7.91	225.67	3037.18	6.33	178.52	2975.76	4.92	262.14	2903.93	3.87	247.57	7.69	4.59	2.44	0.00



(a) Comparison of solution gaps in small and medium-sized samples: Relative to the GAMS's solutions.



(b) Comparison of solution gaps in large-sized samples: Relative to the best-found solution among all the metaheuristics.

Fig. 13. Comparing the solution gaps of the metaheuristics.

including the average objective function values (OFVs) across all runs and the average gap values, are presented in Table 3. As can be seen, the GAMS's CPU times are notably high (even for small-scale cases), making it inefficient to address large-scale instances effectively. In contrast, the proposed metaheuristics deliver near-optimal solutions in significantly shorter CPU times. For example, our testing demonstrates that the gap between GAMS and GWO-SA is statistically insignificant (averaging just 0.106 % across all instances), while showing remarkable computational efficiency compared to GAMS.

Since GAMS software is unable to address large-scale instances, we employ the proposed metaheuristics to solve those instances (see Table 4). In this case, we employ the relative percentage deviation (RPD) metric, using the results of the best-performing metaheuristic as our benchmark, to assess the effectiveness of the algorithms (Hajiaghaei-Keshteli et al., 2023; Nayeri et al., 2021):

$$RPD = \frac{Heur_{sol} - Min_{sol}}{Min_{sol}} \times 100 \tag{37}$$

where $Heur_{sol}$ is the solution produced by the selected metaheuristic, and Min_{sol} is the best solution found among all the metaheuristics.

To facilitate a comprehensive comparison, we present a boxplot in Fig. 13 (a) showing the solution gaps of the metaheuristics relative to the GAMS's solutions in small and medium-scale samples. Additionally, Fig. 13 (b) displays the RPDs among the metaheuristics in large-scale samples. Our analysis demonstrates the GWO-SA algorithm's ability to provide superior solutions. In small and medium-scale test samples, the GWO-SA algorithm consistently achieves zero or negligible solution gaps compared to the GAMS's solutions. This trend carries over to large-

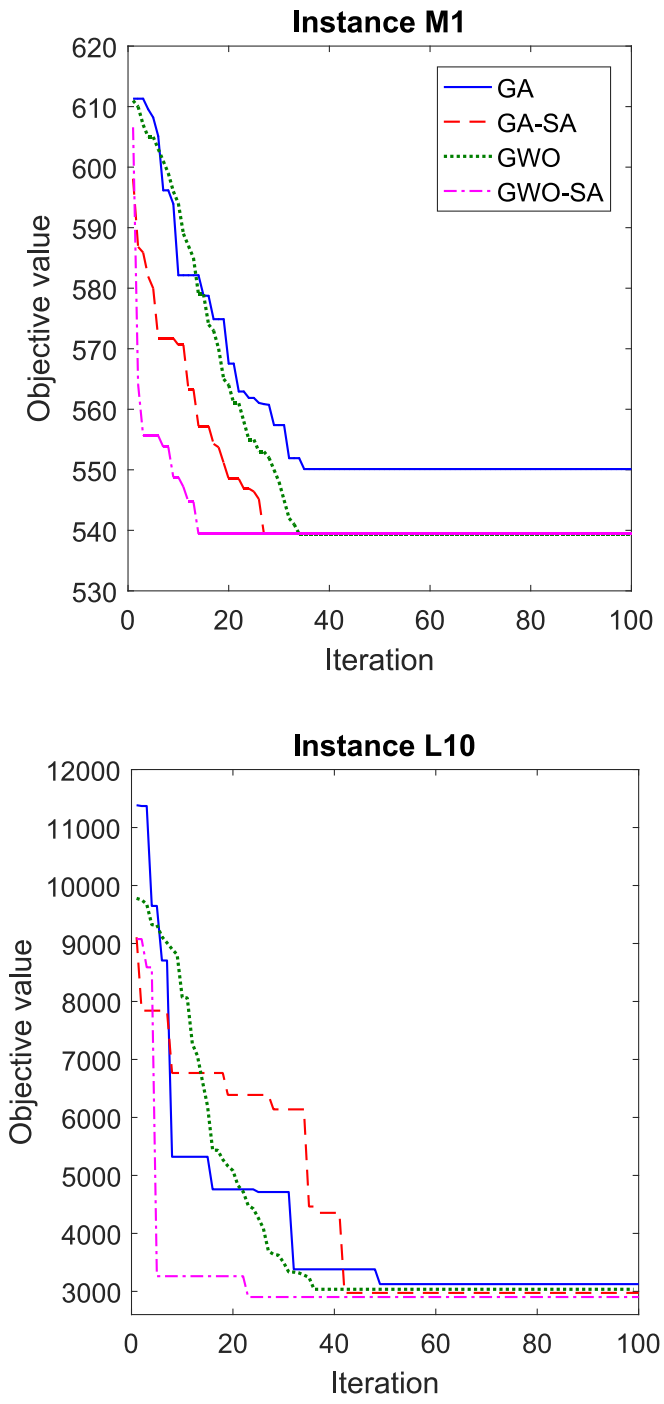


Fig. 14. Convergence curves of the proposed metaheuristics.

Table 5
Results of Wilcoxon test on the proposed metaheuristics.

Algorithms	*p*-values
GWO-SA vs. GA	0.000
GWO-SA vs. GWO	0.000
GWO-SA vs. GA-SA	0.000
GA-SA vs. GA	0.000
GA-SA vs. GWO	0.000
GWO vs. GA	0.000

scale samples, where the GWO-SA algorithm maintains its position as the top-performing algorithm. Significantly, no other algorithm outperforms the GWO-SA algorithm across large-scale instances, leading to the RPD values of zero. The ranking of the other algorithms, in descending order, is as follows: GA-SA, GWO, and GA. We also evaluated the stability of the metaheuristics by examining the standard deviation of their answers across multiple runs (Tables 3 and 4). The GWO-SA algorithm demonstrates superior stability with fewer variations, ensuring a higher level of confidence in the obtained solutions.

In Tables 3 and 4, the CPU times of the algorithms are presented, with the GWO algorithm showing the shortest times. However, the difference in CPU times between GWO-SA and GWO, around 1 min in large-scale problems like *L*10, is not considered significant in the MFBS context, while the GWO-SA algorithm delivers superior solutions. Besides, in Fig. 14, the convergence behavior of the algorithms for different-sized examples (M1 and *L*10) is depicted. The GWO-SA algorithm witnesses a faster convergence compared to the other algorithms, and this trend is consistent across all examples. Overall, our findings affirm the superiority of the GWO-SA algorithm and position it as a favorable choice for solving the MFBS problem.

To assess whether there is a significant difference between the solutions generated by the metaheuristics, we conduct a nonparametric Wilcoxon signed rank test at a significance level of 0.05. The corresponding *p*-values, obtained from pairwise comparisons of the metaheuristics, are presented in Table 5, indicating statistically significant differences among the solutions (p-value ≤ 0.05). Overall, based on the Wilcoxon test results and the reported solution values, it is confirmed that the GWO-SA outperforms the other metaheuristics. Furthermore, the superiority of the GA-SA over the GWO and GA, as well as the superiority of the GWO over the GA, are validated.

5. Real-world application

In this section, we examine the practicality of the model through a series of computational tests conducted on a real case study, the Los Pajaritos bus corridor in Santiago, Chile. This bidirectional corridor consists of 20 stations, with 10 stations in each direction, spanning from the south-west of the city towards the center. Our analysis utilizes the demand data sourced from Sadrani et al. (2022b). During the morning peak period, the corridor experiences a high passenger volume, averaging 4500 passengers per hour, resulting in significant crowding. The primary objective for planners is to enhance the quality of service for travelers by reducing waiting times and improving trip comfort, while minimizing operating costs. For the simulation of our numerical programs, we take into account the crucial morning peak period from 7:00 AM to 10:00 AM, with the demand profile displayed in Fig. B.1 (Appendix B). To operate the designated bus corridor, we assume a maximum availability of 25 type A buses (12-m long), 10 type B buses (15-m long), and 16 type C buses (18-m long).

Fig. 15 illustrates the best vehicle dispatching plans for the first hour of the simulations. As can be seen, the algorithms propose different resource allocation solutions. Fig. 16 presents a comparison of the cost and average occupancy levels (measured as the ratio of the number of passengers to the capacity of the vehicles) among the solutions obtained by each metaheuristic. The GWO-SA algorithm suggests the most cost-effective solution, with a fleet configuration of 39 buses, including 19 type-A, 7 type-B, and 13 type-C buses. Besides, the GWO-SA's solution exhibits the lowest occupancy levels, indicating its effectiveness in optimizing resource utilization and minimizing crowding.

Furthermore, while both GWO-SA and GA-SA suggest the same fleet composition, they offer different dispatching plans regarding sequence and timing, as illustrated in Fig. 15. Notably, the dispatching plan generated by the GWO-SA demonstrates superior efficiency compared to the GA-SA, resulting in a 7.4 % decrease in user costs, and consequently, a 2.9 % decrease in overall costs. Additionally, the average occupancy level inside buses decreases by 10.8 % (from 0.74 to 0.66). Overall,

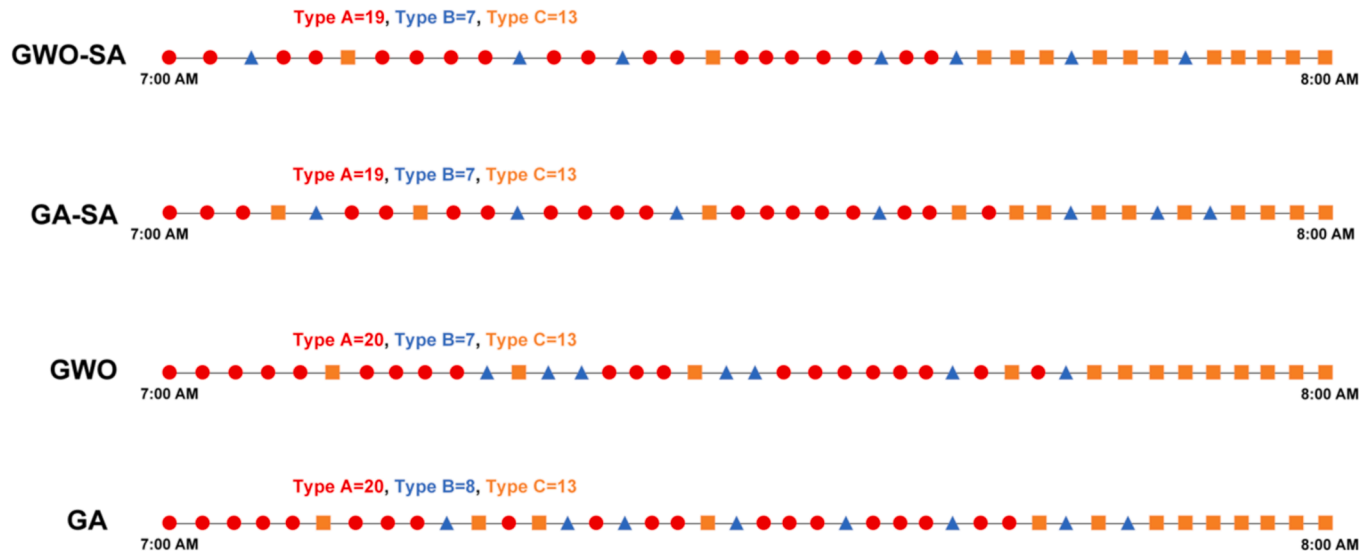


Fig. 15. Best vehicle dispatching solutions suggested by each metaheuristic.

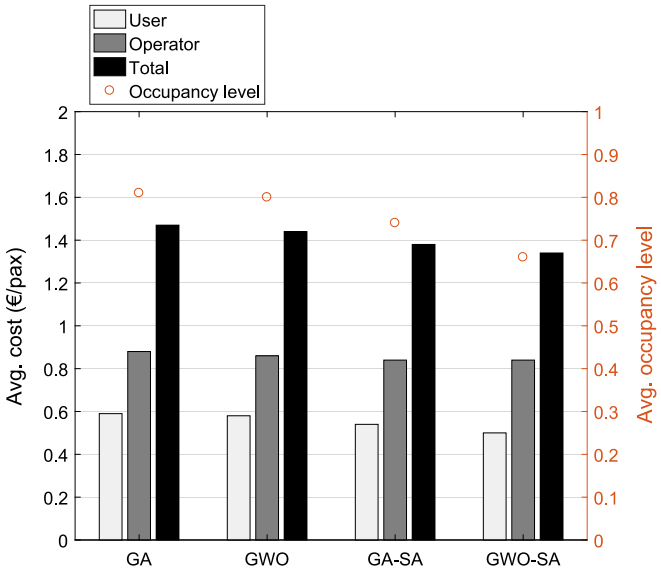


Fig. 16. Comparison of cost and occupancy levels in the best solutions found by each metaheuristic.

during periods of high demand, it is common practice to deploy larger vehicles and reduce dispatching headways compared to periods of lower demand, aligning with the temporal variations in demand illustrated in Fig. B.1. By implementing these optimized programs, users experience improved trip comfort due to reduced vehicle loads, and their waiting times are reduced due to shorter dispatching headways during periods of high demand. Only a framework that explicitly accounts for crowding externalities is able to correctly measure the advantage of using GWO-SA, in terms of increased service quality, in this type of busy transport corridor.

Interestingly, the GA and GWO algorithms propose larger fleet assignments of 41 and 40 buses, respectively, compared to the GA-SA and GWO-SA algorithms recommending a fleet of 39 buses. However, the best dispatching plans derived from the GA-SA and GWO-SA algorithms result in lower user costs and vehicle occupancy levels, despite their smaller fleet assignments. This emphasizes the crucial role of efficient and precise dispatching plans obtained from these advanced algorithms in achieving cost savings and improving the overall performance of the transportation system. We also perform further component-level numerical experiments to better analyze the marginal contribution of specific components to the overall performance of the algorithms (see Appendix C).

5.1. Sensitivity to demand

In this section, we analyze the sensitivity of solutions to variations in the demand level. To accomplish this, we introduce a demand multiplier (DM) to adjust the demand accordingly. Given that the base demand level is already high in this busy corridor (4500 pax/h), our analysis includes the application of various demand multipliers to simulate decreased demand levels, such as DM = 0.8 representing a 20 % reduction from the base case demand.

For each demand level, Fig. 17 displays the fleet configurations recommended by each metaheuristic. In addition, Fig. 18 provides a comparative analysis of the cost and occupancy levels among the different metaheuristics. Based on our findings, utilizing more accurate and advanced solution algorithms, such as GWO-SA and GA-SA, proves to be more relevant for the scheduling of mixed fleets in highly crowded situations that require a larger fleet. The diversity of potential solutions for assigning different types of vehicles becomes increasingly intricate in such scenarios. However, as the demand decreases, the performance differences among the algorithms become less significant, and the use of simpler algorithms is enough to produce satisfactory results. For example, when comparing the solutions of the GA and GWO-SA at DM = 0.8 (Fig. 18), the GWO-SA's solution leads to an 8.1 % reduction in total cost (from 1.48 to 1.36 €/pax) and an 18.2 % reduction in occupancy level (from 0.77 to 0.63). However, at lower demand levels, there are no significant differences in the solutions provided by the algorithms. The occupancy levels depicted in Fig. 18 highlight the superiority of the GWO-SA algorithm in highly crowded situations, whereas the performance differences between the algorithms diminish as the demand decreases. Fig. D1 (Appendix D) showcases the dispatching plans for heterogeneous fleet operations at DM = 0.8 and DM = 0.6. Although all three algorithms (GWO, GA-SA, and GWO-SA) propose the same fleet composition of 26 vehicles at DM = 0.6, the dispatching scheme generated by GWO-SA proves to be more effective, resulting in greater reductions in cost and occupancy levels.

As the demand declines in the scheduling of heterogeneous fleets, the overall size of the allocated fleet, particularly the proportion of larger vehicles, decreases. For instance, when the demand is reduced using a

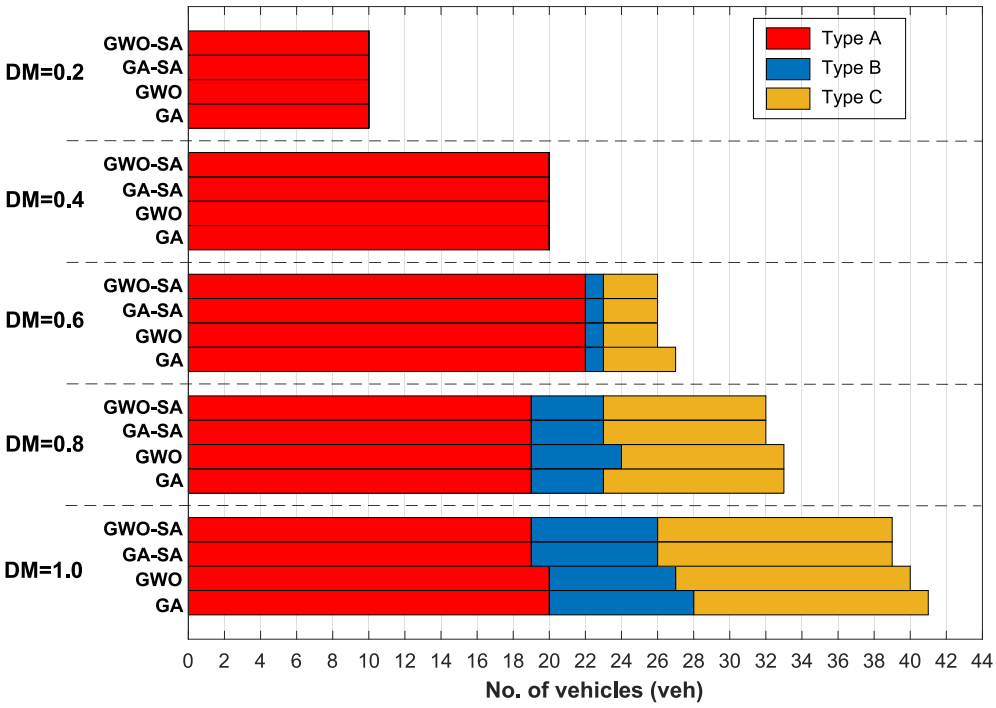


Fig. 17. Fleet composition (vehicle assignment solutions) proposed by each metaheuristic for different demand levels (DM stands for demand multiplier, e.g., DM = 0.8 represents a 20 % reduction in the base case demand).

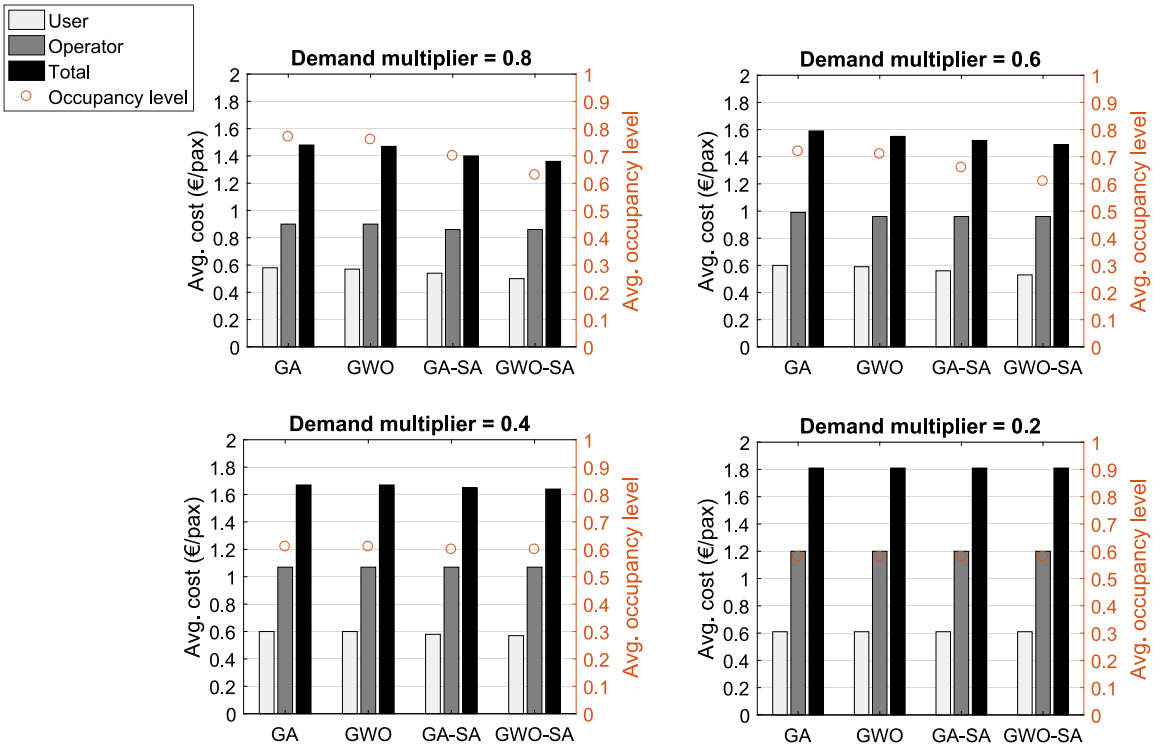


Fig. 18. Comparison of cost and occupancy levels obtained by each metaheuristic at different demand levels.

demand multiplier (DM) of 0.6 (Fig. 17), the allocated fleet of 39 vehicles (including 19 type-A, 7 type-B, and 13 type-C buses) decreases to 26 vehicles (including 22 type-A, 1 type-B, and 3 type-C buses) in the GWO-SA algorithm. This indicates that the number of smaller (type A) vehicles increases while the number of larger vehicles decreases, resulting in a more cost-effective operation.

5.2. *Sensitivity to crowding discomfort valuations*

We conduct a sensitivity analysis on the solutions generated by the GWO-SA algorithm to assess their sensitivity to variations in user cost savings from crowding. We change the crowding multiplier values by ± 10 %, ±20 %, and ± 30 % to evaluate the algorithm's response. The

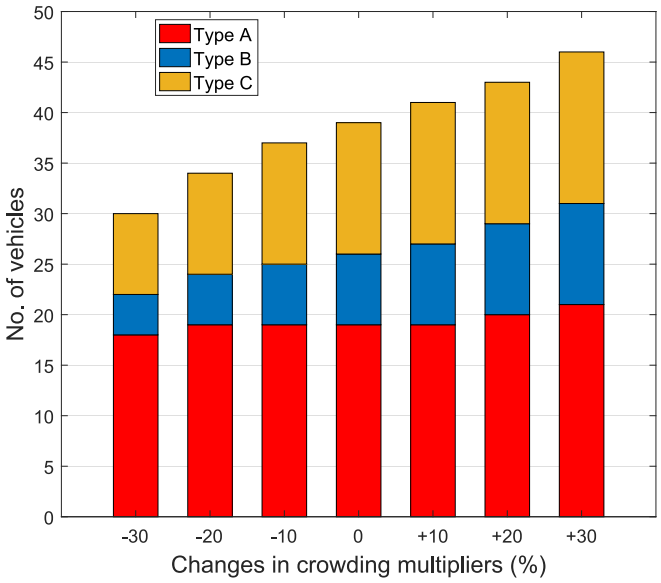


Fig. 19. Sensitivity to crowding discomfort.

Table 6
Sensitivity to driving time uncertainty.

Travel time scenario	No. of vehicles (veh)		
	Type A	Type B	Type C
Deterministic	19	7	13
Uncertain	22	7	13

higher crowding multipliers represent scenarios comparable to the COVID-19 situation, where the fear or possibility of infection can increase passenger discomfort inside vehicles, as demonstrated by studies such as Basnak et al. (2022) estimating crowding multipliers during the COVID-19 pandemic. The sensitivity analysis results provide valuable managerial insights for fleet planning in pandemic-related circumstances, as shown in Fig. 19. For example, when user dissatisfaction from crowding increases by 30 %, we observe a corresponding 18 % increase in the allocated fleet. Notably, the number of larger vehicles (type B and type C) shows a higher rate of increase by 42.8 % and 15.4 %, respectively. In contrast, the number of type-A vehicles shows a more modest increase of 10.5 %.

5.3. Sensitivity to uncertain driving times

We investigate the impact of uncertain travel times on vehicle assignment strategies in the MFBS problem. External factors such as road congestion, driver performance, and weather can introduce variability in the driving times of vehicles (Sadrani et al., 2022b). In this part, instead of using deterministic driving times, we incorporate random driving times derived from a log-normal distribution[1], i.e., $M^{n}_{r,s}$ lognormal $(V_s\ ,\ D_s)$, where V_s and D_s stand for the average and standard deviation of driving times, respectively, for the segment between stations $s-1$ and s.

[1] Asymmetrical distributions such as the lognormal, loglogistic and gamma are usually found to provide good fits to the observed distributions of travel times by cars and buses, given that travel times are skewed with long right tails due to congestion (Durán-Hormazábal and Tirachini, 2016), while in some cases normal distributions have also been proposed (for a full review of travel time distributions in public transport see Büchel and Corman 2020).

In the stochastic setting, each service may encounter a different driving time for the same segment, introducing uncertainty in travel times. To handle this uncertainty, we deploy an independent Monte Carlo Simulation (MCS) program during the evaluation phase of the metaheuristics, enabling us to conduct enough evaluation runs (1000) to determine the OFV of each solution (Zhang et al., 2021, Sadrani et al., 2022a), as opposed to a single run that is suitable for deterministic cases. Essentially, the MCS program functions as a subroutine and is triggered specifically during the evaluation phase. More details can be found in Fig. E.1 in Appendix E.

In Table 6, we report the sensitivity of the GWO-SA algorithm's solutions when uncertain travel times are incorporated. The results show an increase in the number of operational services, specifically a 13.6 % rise in the number of type-A vehicles (from 19 to 22). This adjustment compensates for unfavorable waiting times experienced by users due to the unpredictability of driving times.

6. Synthesis and concluding remarks

In this research, we proposed a mixed-fleet bus scheduling (MFBS) problem to optimize the number and type of vehicles assigned for mixed-fleet operations, as well as their dispatching order and times. It accounted for user costs (waiting and crowding-sensitive in-vehicle time costs) and operator costs (driver and vehicle running costs).

To solve the MFBS problem, we employed Genetic Algorithm (GA) and Grey Wolf Optimizer (GWO). We also developed two hybrid metaheuristics, GA-SA [a combination of GA and Simulated Annealing (SA)] and GWO-SA (a combination of GWO and SA). Extensive numerical experiments were conducted to assess the performance of the proposed metaheuristics. Overall, the results showed that the GWO-SA outperformed the other algorithms. It also exhibited superior stability with fewer variations across multiple runs. Furthermore, the GA-SA demonstrated superior solution quality compared to GA and GWO.

The applicability of the model was tested on a real bus corridor in Santiago, Chile. The GWO-SA algorithm proposed the most cost-effective solution, with a fleet configuration of 39 buses (19 type-A, 7 type-B, and 13 type-C), resulting in the lowest occupancy levels and optimized resource utilization. While both GWO-SA and GA-SA algorithms suggested the same fleet composition, the dispatching plan generated by GWO-SA outperformed GA-SA, leading to a 7.4 % decrease in user costs and a 2.9 % decrease in overall costs, with a 10.8 % reduction in average bus occupancy levels, enhancing trip comfort.

Interestingly, although GA and GWO algorithms proposed larger fleet assignments (41 and 40 buses), GA-SA and GWO-SA algorithms with 39 buses achieved lower user costs and vehicle occupancy levels. This result underscores the importance of generating precise dispatching plans through advanced algorithms for cost savings and improved performance.

Our sensitivity analysis of demand variations revealed that more advanced solution algorithms, such as GWO-SA and GA-SA, enhance the scheduling of mixed fleets in crowded scenarios, where the complexity of assigning different vehicle types becomes more intricate. On the contrary, as demand decreases, the performance differences among the algorithms become less significant, and simpler algorithms can be chosen to optimize the bus service without a deterioration of service quality.

We conducted a sensitivity analysis on the solutions generated by the GWO-SA algorithm. The results showed that as user dissatisfaction from crowding increased, the allocated fleet increased accordingly, with a higher rate of increase in the number of larger vehicles. This analysis provides valuable insights for fleet planning, especially in pandemic-related circumstances. Incorporating uncertainty in travel times using a Monte Carlo Simulation (MCS) setting, we observed that the GWO-SA algorithm adjusted vehicle assignment programs to compensate for unfavorable waiting times due to unpredictable driving times, resulting in more operational services, especially for 12-m long buses.

Future research can extend the model to include other vehicle

technologies, such as automated buses, and explore the combined case of human-driven and automated vehicles for public transport services. Investigating further metaheuristics for solving the MFBS problem would also be interesting. Moreover, future studies can focus on designing efficient transformation methods to convert our nonlinear model into a linear one, which can improve computational efficiency when solving the problem using off-the-shelf commercial solvers.

CRediT authorship contribution statement

Mohammad Sadrani: Conceptualization, Methodology, Formal analysis, Software, Visualization, Writing – original draft. **Alejandro Tirachini:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Constantinos Antoniou:** Conceptualization, Methodology, Supervision, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the financial support from the German MCube cluster (Förderkennzeichen: 03ZU1105KA) and the TUM-IGSSE (project 12.04-MO3, Germany). We sincerely appreciate the insightful comments from the Editor and two anonymous reviewers, which have improved the content and presentation of this paper.

Appendix A. Taguchi orthogonal arrays

Table A1
Taguchi orthogonal array L9 (3^4) (4 factors (A, B, C, and D) at 3 levels).

Trial number	Factor			
	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Appendix B. Input parameter values utilized in test and real-life programs

Table B1 lists the characteristics of fleet composition (resource availability), demand level, and bus route for the simulation of test samples, solved to validate the performance of the metaheuristics.

Table B1
Characteristics of test instances.

Class	Instance ID	Max No. of available vehicles for operations	U_A	U_B	U_C	No. of stations	Demand level (pax/h)
Small	S1	4	2	1	1	6	300
	S2	4	1	1	2	6	310
	S3	5	2	2	1	6	400
	S4	5	2	1	2	6	420
	S5	6	2	2	2	8	450
	S6	6	1	3	2	8	480
	S7	6	1	2	3	8	500
	S8	7	3	3	1	8	520
	S9	7	2	3	2	8	550
	S10	7	1	3	3	8	590
Medium	M1	8	4	2	2	10	600
	M2	8	3	3	2	10	635
	M3	8	2	3	3	10	670
	M4	9	4	3	2	12	700
	M5	9	3	3	3	12	700
	M6	9	2	3	4	12	750
	M7	10	4	4	2	14	850
	M8	10	4	3	3	14	850
	M9	10	2	4	4	14	870
	M10	11	3	4	4	14	950
Large	L1	14	6	4	4	16	1000
	L2	15	7	4	4	16	1100
	L3	18	9	5	4	16	1300

(*continued on next page*)

Table B1 (*continued*)

Class	Instance ID	Max No. of available vehicles for operations	U_A	U_B	U_C	No. of stations	Demand level (pax/h)
	L4	22	12	5	5	16	1500
	L5	26	13	7	6	18	2000
	L6	30	15	8	7	18	2300
	L7	33	16	9	8	18	2800
	L8	33	12	11	10	18	2800
	L9	38	22	8	8	20	3000
	*L*10	40	20	10	10	20	3300

U_A, U_B, and U_C stand for the maximum resource availability on type A (12-m long), type B (15-m long), and type C (18-m long) vehicles, respectively.

Other time- and cost-related parameters, utilized in our test and real-world programs (in the Santiago case study), are listed in Table B2. It should be noted that size-sensitive parameters (such as vehicle loading capacity and running costs), whose values are dependent on the vehicle size, are reported in Table B3. Moreover, for the simulation of scenarios with uncertain driving times, the log-normal distribution uses the mean and standard deviation of driving times assumed as 2.5 (min) and 0.6 (min), respectively.

Table B2
List of parameter values.

Description	Unit	Value
Minimum dispatching headway	min	1
Maximum dispatching headway	min	10
Riding time between stops	min	2.5
Time for opening and closing doors	s	6
Average time for alighting action	s/pax	1.5
Average time for boarding action	s/pax	2.5
Value of in-vehicle time savings	€/h	2.9
Value of waiting time savings	€/h	5.8
Human driving cost	€/veh-h	6.2

Table B3
Size-sensitive parameter values (Source: Sadrani et al., 2022b).

Description	Unit	Standard bus	Rigid bus	Articulated bus
Vehicle size	m	12	15	18
Vehicle total capacity	pax/veh	70	90	120
Vehicle seating capacity	–	40	50	60
Vehicle running cost	€/veh-h	11.2	14.4	17.6

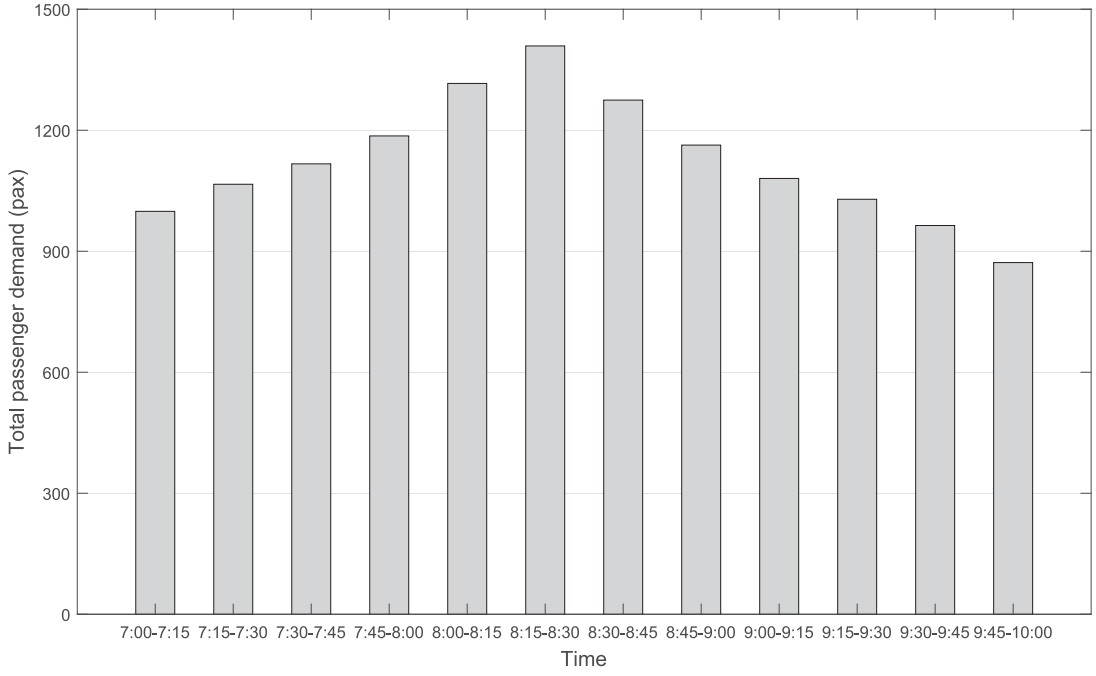


Fig. B1. Passenger arrival volumes on the route at various periods.

Appendix C. Component-level numerical experiments

In this section, we conduct additional component-level numerical experiments to isolate and analyze the contribution of key components to the overall performance of each algorithm. We perform controlled experiments by comparing the algorithms with and without specific components. The components examined include the crossover operator in the GA, the leadership hierarchy in the GWO, and the integration of the SA rule in the hybrid GA-SA and GWO-SA algorithms. The specific tests conducted are as follows:

a) GA – Impact of Crossover.

- GA-V1 (Baseline GA): The standard GA configuration with a crossover rate of 80 % (as fine-tuned in Table 2).
- GA-V2 (Without Crossover): The GA does not apply any crossover, relying solely on mutation.

b) GWO – Impact of Leadership Hierarchy.

- GWO-V1 (Baseline GWO): The standard GWO configuration with its original hierarchy, where the alpha, beta, and delta wolves (the three top solutions) lead the search.
- GWO-V2 (Without Beta and Delta Wolves): Only the alpha wolf guides the search.
- GWO-V3 (Without Delta Wolf): The alpha and beta wolves guide the search (two leading solutions in the archive).
- GWO-V4 (With Four Leading Solutions): The four top solutions guide the search.
- GWO-V5 (With Five Leading Solutions): The five top solutions guide the search.

c) Impact of SA Integration

- GA-SA and GWO-SA: The SA rule is incorporated into the baseline GA and GWO.

The component-level results are presented in Table C1. For each experiment, we report the average objective function value (OFV), the standard deviation, and the convergence speed (the average number of iterations required to converge). Comparing GA-V1 (baseline GA) to GA-V2 (without crossover), the average OFV improves by about 18 % (from 4007 to 3300) when crossover is included. The standard deviation decreases, indicating more consistent performance, and the average number of iterations until convergence reduces from 94 to 60. GWO-V1 (baseline GWO) outperforms GWO-V2 (without beta and delta wolves), with the average OFV improving by 19 % (from 4012 to 3232) and faster convergence (42 vs. 83 iterations). This highlights the importance of the leadership hierarchy in guiding the search more effectively based on multiple leading solutions. GWO-V3 (without delta wolf) shows intermediate performance, indicating that having both alpha and beta wolves is beneficial but not as effective as the original hierarchy. The average OFV is 3424, and convergence is slower than the baseline GWO.

GWO-V4 and GWO-V5 (with four and five leading solutions, respectively) do not improve performance compared to the baseline GWO. In fact, performance slightly degrades, suggesting that increasing the number of leading wolves beyond three may introduce extra convergence pressure, reducing diversity and overall solution quality. Therefore, reducing or excessively enlarging the hierarchy adversely affects the GWO's ability to balance exploration and exploitation. Integrating the SA rule into both GA and GWO further improves their performance. For instance, GA-SA reduces the average OFV by 4 % compared to the baseline GA (from 3300 to 3178). GWO-SA reduces the average OFV by 4 % compared to the baseline GWO (from 3232 to 3107).

Table C1
Component-level numerical experiments.

Algorithm	Component Modified	Avg. Objective Value (Over 10 Runs)	Standard Deviation	Ranking of Solution Fitness	Avg. Number of Iterations Until Convergence
GA-V1 (Baseline GA)	–	3300	6.4	**5**	60
GA-V2 (Without Crossover)	No crossover	4007	11.5	**8**	94
GA-SA	Baseline GA integrated with SA rule	3178	5.7	**2**	51
GWO-V1 (Baseline GWO)	–	3232	6.1	**3**	42
GWO-V2 (Without Beta and Delta)	Leadership hierarchy reduced	4012	8.2	**9**	83
GWO-V3 (Without Delta)	Leadership hierarchy reduced	3424	7.1	**6**	69
GWO-V4 (With Four Leading Solutions)	Leadership hierarchy enlarged	3271	6.2	**4**	74
GWO-V5 (With Five Leading Solutions)	Leadership hierarchy enlarged	3485	7.8	**7**	81
GWO-SA	Baseline GWO integrated with SA rule	3107	5.4	**1**	44

Appendix D. Dispatching schemes

Fig. D1. Dispatching schemes offered by each metaheuristic for different demand levels.

Appendix E. Monte Carlo Simulation program

Step (1): Define the parameters of MCS: Define the simulation counter u and adjust its starting value as 1. Let $\overline{OFV}^{(u)}$ represent the estimated objective function value (OFV) in Eq. (1). Set the maximum number of simulations as $U_{max} = 1000$.

Step (2): Conduct driving time sampling: The driving time of vehicles at each trip segment is a random variable whose mean and standard deviation are predefined. For each service r, sample the driving time between stations $s-1$ and s (i.e., $M_{r,s}^{n}$) by means of a log-normal distribution function, i.e., $M_{r,s}^{n} \sim \text{lognormal}\,(V_s\,,\,D_s)$, where V_s and D_s refer to the mean and standard deviation of driving time between stations $s-1$ and s, respectively.

Step (3): Update the variables: In view of the sampled driving time value, update other variables associated with the solution in Eqs. (8)–(22), including vehicle motion and traveler flow computations.

Step (4): Calculate the OFV: Compute and update the $OFV^{(u)}$ based on Eq. (1), and the ultimate output of the OFV is derived from the average value of the simulated samples:

$$\overline{OFV}(u) = \frac{OFV^{(u)} + (u-1)\,.\,\overline{OFV}^{(u-1)}}{u}$$

Step (5): Check the termination condition: Add the number of simulations by one, $u = u+1$. If $u < U_{max}$, return to Step (2); otherwise, stop and report the estimated OFV, $\overline{OFV} = \overline{OFV}^{(u)}$.

Fig. E1. Description of the Monte Carlo Simulation program.

Data availability

Data will be made available on request.

References

Agrawal, K., Suman, H. K., & Bolia, N. B. (2020). Frequency optimization models for reducing overcrowding discomfort. *Transp. Res. Rec., 2674*, 160–171. https://doi.org/10.1177/0361198120912230

Akbari, M., Rashidi, H., & Alizadeh, S. H. (2017). An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. *Eng. Appl. Artif. Intell., 61*, 35–46. https://doi.org/10.1016/j.engappai.2017.02.013

Ala, A., Simic, V., Pamucar, D., & Tirkolaee, E. B. (2022). Appointment scheduling problem under fairness policy in healthcare services: Fuzzy ant lion optimizer. *Expert Syst. Appl., 207*, Article 117949. https://doi.org/10.1016/j.eswa.2022.117949

Aranha, C., Camacho Villalón, C. L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., & Stützle, T. (2022). Metaphor-based metaheuristics, a call for action: The elephant in the room. *Swarm Intell., 16*(1), 1–6. https://doi.org/10.1007/s11721-021-00202-9

Assadi, M. T., & Bagheri, M. (2016). Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. *Comput. Ind. Eng., 96*, 149–161. https://doi.org/10.1016/j.cie.2016.03.021

Basnak, P., Giesen, R., & Muñoz, J. C. (2022). Estimation of crowding factors for public transport during the COVID-19 pandemic in Santiago, Chile. *Transp. Res. Part A Policy Pract., 159*, 140–156. https://doi.org/10.1016/j.tra.2022.03.011

Beheshti, Z. (2021). A novel x-shaped binary particle swarm optimization. *Soft Comput., 25*, 3013–3042. https://doi.org/10.1007/s00500-020-05360-2

Berrebi, S. J., Watkins, K. E., & Laval, J. A. (2015). A real-time bus dispatching policy to minimize passenger wait on a high-frequency route. *Transp. Res. Part B Methodol., 81*, 377–389. https://doi.org/10.1016/j.trb.2015.05.012

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Inf. Sci., 237*, 82–117. https://doi.org/10.1016/j.ins.2013.02.041

Büchel, B., & Corman, F. (2020). Review on statistical modeling of travel time variability for road-based public transport. *Front. Built Environ., 6*, 70. https://doi.org/10.3389/fbuil.2020.00070

Ceder, A. A., Hassold, S., & Dano, B. (2013). Approaching even-load and even-headway transit timetables using different bus sizes. *Public Transp., 5*, 193–217. https://doi.org/10.1007/s12469-013-0062-z

Chen, J., Liu, Z., Zhu, S., & Wang, W. (2015). Design of limited-stop bus service with capacity constraint and stochastic travel time. *Transp. Res. Part E Logist. Transp. Rev., 83*, 1–15. https://doi.org/10.1016/j.tre.2015.08.007

Dai, Z., Liu, X. C., Chen, X., & Ma, X. (2020). Joint optimization of scheduling and capacity for mixed traffic with autonomous and human-driven buses: A dynamic programming approach. *Transp. Res. Part C Emerg. Technol., 114*, 598–619. https://doi.org/10.1016/j.trc.2020.03.001

Dakic, I., Yang, K., Menendez, M., & Chow, J. Y. J. (2021). On the design of an optimal flexible bus dispatching system with modular bus units: Using the three-dimensional macroscopic fundamental diagram. *Transp. Res. Part B Methodol., 148*, 38–59. https://doi.org/10.1016/j.trb.2021.04.005

de Armas, J., Lalla-Ruiz, E., Tilahun, S. L., & Voß, S. (2022). Similarity in metaheuristics: A gentle step towards a comparison methodology. *Nat. Comput., 21*(2), 265–287. https://doi.org/10.1007/s11047-020-09837-9

Desaulniers, G., and Hickman, M.D., 2007. Chapter 2 Public Transit. Handb. Oper. Res. Manag. Sci. 14, 69–127. DOI: 10.1016/S0927-0507(06)14002-5.

Drabicki, A., Cats, O., Kucharski, R., Fonzone, A., & Szarata, A. (2023). Should I stay or should I board? Willingness to wait with real-time crowding information in urban public transport. *Res. Transp. Bus. Manag., 47*, Article 100963. https://doi.org/10.1016/j.rtbm.2023.100963

Dulebenets, M. A., Golias, M. M., & Mishra, S. (2018). A collaborative agreement for berth allocation under excessive demand. *Eng. Appl. Artif. Intell., 69*, 76–92. https://doi.org/10.1016/j.engappai.2017.11.009

Durán-Hormazábal, E., & Tirachini, A. (2016). Estimation of travel time variability for cars, buses, metro and door-to-door public transport trips in Santiago. *Chile. Res. Transp. Econ., 59*, 26–39. https://doi.org/10.1016/j.retrec.2016.06.002

Durán-Micco, J., & Vansteenwegen, P. (2022). A survey on the transit network design and frequency setting problem. *Public Transp., 14*, 155–190. https://doi.org/10.1007/s12469-021-00284-y

Duran-Micco, J., Vermeir, E., & Vansteenwegen, P. (2020). Considering emissions in the transit network design and frequency setting problem with a heterogeneous fleet. *Eur. J. Oper. Res., 282*, 580–592. https://doi.org/10.1016/j.ejor.2019.09.050

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl., 30*, 413–435. https://doi.org/10.1007/s00521-017-3272-5

Fathollahi-Fard, A. M., Ahmadi, A., Goodarzian, F., & Cheikhrouhou, N. (2020a). A bi-objective home healthcare routing and scheduling problem considering patients' satisfaction in a fuzzy environment. *Appl. Soft Comput., 93*, Article 106385. https://doi.org/10.1016/j.asoc.2020.106385

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2020b). Red deer algorithm (RDA): A new nature-inspired meta-heuristic. *Soft Comput., 24*, 14637–14665. https://doi.org/10.1007/s00500-020-04812-z

Fathollahi-Fard, A. M., Dulebenets, M. A., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R., Safaeian, M., & Mirzahosseinian, H. (2021). Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design

problem in the tire industry under uncertainty. *Adv. Eng. Inform., 50*, Article 101418. https://doi.org/10.1016/j.aei.2021.101418
Frieß, N. M., & Pferschy, U. (2024). Planning a zero-emission mixed-fleet public bus system with minimal life cycle cost. *Public Transp., 16*, 39–79. https://doi.org/10.1007/s12469-023-00345-4
Ge, L., Voß, S., & Xie, L. (2022). Robustness and disturbances in public transport. *Public Transp., 14*, 191–261. https://doi.org/10.1007/s12469-022-00301-8
Gkiotsalitis, K. (2020). A model for the periodic optimization of bus dispatching times. *Appl. Math. Model., 82*, 785–801. https://doi.org/10.1016/j.apm.2020.02.003
Gkiotsalitis, K., & Alesiani, F. (2019). Robust timetable optimization for bus lines subject to resource and regulatory constraints. *Transp. Res. Part E Logist. Transp. Rev., 128*, 30–51. https://doi.org/10.1016/j.tre.2019.05.016
Gkiotsalitis, K., & Cats, O. (2021). At-stop control measures in public transport: Literature review and research agenda. *Transp. Res. Part E Logist. Transp. Rev., 145*, Article 102176. https://doi.org/10.1016/j.tre.2020.102176
Gkiotsalitis, K., & Cats, O. (2018). Reliable frequency determination: Incorporating information on service uncertainty when setting dispatching headways. *Transp. Res. Part C Emerg. Technol., 88*, 187–207. https://doi.org/10.1016/j.trc.2018.01.026
Gkiotsalitis, K., Cats, O., & Liu, T. (2022). A review of public transport transfer synchronisation at the real-time control phase. *Transp. Rev., 43*(1), 88–107. https://doi.org/10.1080/01441647.2022.2035014
Goli, A., Ala, A., & Hajiaghaei-Keshteli, M. (2023). Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem. *Expert Syst. Appl., 213*, Article 119077. https://doi.org/10.1016/j.eswa.2022.119077
Goli, A., Tirkolaee, E. B., & Aydın, N. S. (2021). Fuzzy integrated cell formation and production scheduling considering automated guided vehicles and human factors. *IEEE Trans. Fuzzy Syst., 29*, 3686–3695. https://doi.org/10.1109/TFUZZ.2021.3053838
Goodarzian, F., Taleizadeh, A. A., Ghasemi, P., & Abraham, A. (2021). An integrated sustainable medical supply chain network during COVID-19. *Eng. Appl. Artif. Intell., 100*, Article 104188. https://doi.org/10.1016/j.engappai.2021.104188
Hajiaghaei-Keshteli, M., Rahmanifar, G., Mohammadi, M., Gholian-Jouybari, F., Klemeš, J. J., Zahmatkesh, S., Bokhari, A., Fusco, G., & Colombaroni, C. (2023). Designing a multi-period dynamic electric vehicle production-routing problem in a supply chain considering energy consumption. *J. Clean. Prod., 421*, Article 138471. https://doi.org/10.1016/j.jclepro.2023.138471
Hashemi-Amiri, O., Mohammadi, M., Rahmanifar, G., Hajiaghaei-Keshteli, M., Fusco, G., & Colombaroni, C. (2023). An allocation-routing optimization model for integrated solid waste management. *Expert Syst. Appl., 227*, Article 120364. https://doi.org/10.1016/j.eswa.2023.120364
Hörcher, D., Graham, D. J., & Anderson, R. J. (2017). Crowding cost estimation with large scale smart card and vehicle location data. *Transp. Res. Part B Methodol., 95*, 105–125. https://doi.org/10.1016/j.trb.2016.10.015
Jara-Díaz, S., & Gschwender, A. (2003). Towards a general microeconomic model for the operation of public transport. *Transp. Rev., 23*, 453–469. https://doi.org/10.1080/0144164032000048922
Jenelius, E. (2020). Personalized predictive public transport crowding information with automated data sources. *Transp. Res. Part C Emerg. Technol., 117*, Article 102647. https://doi.org/10.1016/j.trc.2020.102647
Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl., 80*, 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
Leon-Blanco, J. M., Gonzalez-R, P., Andrade-Pineda, J. L., Canca, D., & Calle, M. (2022). A multi-agent approach to the truck multi-drone routing problem. *Expert Syst. Appl., 195*, Article 116604. https://doi.org/10.1016/j.eswa.2022.116604
Li, B., Yang, X., & Xuan, H. (2018). A Hybrid Simulated Annealing Heuristic for Multistage Heterogeneous Fleet Scheduling with Fleet Sizing Decisions. *J. Adv. Transp., 2019*(1), 5364201. https://doi.org/10.1155/2019/5364201
Liu, T., Cats, O., & Gkiotsalitis, K. (2021). A review of public transport transfer coordination at the tactical planning phase. *Transp. Res. Part C Emerg. Technol., 133*, Article 103450. https://doi.org/10.1016/j.trc.2021.103450
Liu, Y., As'arry, A., Hassan, M.K., Hairuddin, A.A., Mohamad, H., 2024. Review of the grey wolf optimization algorithm: variants and applications. Neural Comput. Appl. 36, 2713–2735. DOI: 10.1007/s00521-023-09202-8.
Londe, M. A., Pessoa, L. S., Andrade, C. E., & Resende, M. G. (2024). Biased random-key genetic algorithms: A review. *Eur. J. Oper. Res.*. https://doi.org/10.1016/j.ejor.2024.03.030
López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., & Stützle, T. (2015). The irace package: Iterated racing for automatic algorithm configuration. *Oper. Res. Perspect., 3*, 43–58. https://doi.org/10.1016/j.orp.2016.09.002
Luo, X., Liu, Y., Yu, Y., Tang, J., & Li, W. (2019). Dynamic bus dispatching using multiple types of real-time information. *Transportmetrica B Transp. Dyn., 7*, 519–545. https://doi.org/10.1080/21680566.2018.1447408
Mirjalili, S., Dong, J. S., & Lewis, A. (2020). Nature-inspired optimizers. *Cham, Switzerland: Springer, 69–85*. https://doi.org/10.1007/978-3-030-12127-3
Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput., 9*, 1–14. https://doi.org/10.1016/j.swevo.2012.09.002
Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Softw., 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
Mohring, H. (1972). Optimization and scale economies in urban bus transportation. *Am. Econ. Rev., 62*(4), 591–604.
Mou, Z., Zhang, H., & Liang, S. (2020). Reliability optimization model of stop-skipping bus operation with capacity constraints. *J. Adv. Transp., 2020*, 1–11. https://doi.org/10.1155/2020/4317402
Nayeri, S., Tavakkoli-Moghaddam, R., Sazvar, Z., & Heydari, J. (2022). A heuristic-based simulated annealing algorithm for the scheduling of relief teams in natural disasters. *Soft Comput., 26*, 1825–1843. https://doi.org/10.1007/s00500-021-06425-6
Nayeri, S., Tavakoli, M., Tanhaeean, M., & Jolai, F. (2021). A robust fuzzy stochastic model for the responsive-resilient inventory-location problem: Comparison of metaheuristic algorithms. *Ann. Oper. Res., 315*, 1895–1935. https://doi.org/10.1007/s10479-021-03977-6
Nourmohammadzadeh, A., & Voß, S. (2022). A robust multiobjective model for the integrated berth and quay crane scheduling problem at seaside container terminals. *Ann. Math. Artif. Intell., 90*, 831–853. https://doi.org/10.1007/s10472-021-09743-5
Peres, F., & Castelli, M. (2021). Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development. *Appl. Sci., 11*(14), 6449. https://doi.org/10.3390/app11146449
Perumal, S. S., Lusby, R. M., & Larsen, J. (2022). Electric bus planning & scheduling: A review of related problems and methodologies. *Eur. J. Oper. Res., 301*(2), 395–413. https://doi.org/10.1016/j.ejor.2021.10.058
Rahmanifar, G., Mohammadi, M., Sherafat, A., Hajiaghaei-Keshteli, M., Fusco, G., & Colombaroni, C. (2023). Heuristic approaches to address vehicle routing problem in the IoT-based waste management system. *Expert Syst. Appl., 220*, Article 119708. https://doi.org/10.1016/j.eswa.2023.119708
Rinaldi, M., Picarelli, E., D'Ariano, A., & Viti, F. (2020). Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications. *Omega, 96*, Article 102070. https://doi.org/10.1016/j.omega.2019.05.006
Sadrani, M., Tirachini, A., & Antoniou, C. (2022a). Vehicle dispatching plan for minimizing passenger waiting time in a corridor with buses of different sizes: Model formulation and solution approaches. *Eur. J. Oper. Res., 299*, 263–282. https://doi.org/10.1016/j.ejor.2021.07.054
Sadrani, M., Tirachini, A., & Antoniou, C. (2022b). Optimization of service frequency and vehicle size for automated bus systems with crowding externalities and travel time stochasticity. *Transp. Res. Part C Emerg. Technol., 143*, Article 103793. https://doi.org/10.1016/j.trc.2022.103793
Sadrani, M., Jafarian-Moghaddam, A. R., Esfahani, M. A., & Rahimi, A. M. (2023). Designing limited-stop bus services for minimizing operator and user costs under crowding conditions. *Public Transp., 15*, 97–128. https://doi.org/10.1007/s12469-022-00307-2
Shang, H., Huang, H.-J., & Wu, W.-X. (2019). Bus timetabling considering passenger satisfaction: An empirical study in Beijing. *Comput. Ind. Eng., 135*, 1155–1166. https://doi.org/10.1016/j.cie.2019.01.057
Shang, H., Liu, Y., Wu, W., & Zhao, F. (2023). Multi-depot vehicle scheduling with multiple vehicle types on overlapped bus routes. *Expert Syst. Appl., 228*, Article 120352. https://doi.org/10.1016/j.eswa.2023.120352
Sharma, I., Kumar, V., and Sharma, S., 2022. A comprehensive survey on grey wolf optimization. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science) 15, 323–333. DOI: 10.2174/2666255813999201007165454.
Suman, H. K., & Bolia, N. B. (2019). Mitigation of overcrowding in buses through bus planning. *Public Transp., 11*, 159–187. https://doi.org/10.1007/s12469-019-00197-x
Tang, R., de Donato, L., Besinović, N., Flammini, F., Goverde, R. M. P., Lin, Z., Liu, R., Tang, T., Vittorini, V., & Wang, Z. (2022). A literature review of Artificial Intelligence applications in railway systems. *Transp. Res. Part C Emerg. Technol., 140*, Article 103679. https://doi.org/10.1016/j.trc.2022.103679
Tirachini, A., & Antoniou, C. (2020). The economics of automated public transport: Effects on operator cost, travel time, fare and subsidy. *Economics of Transportation, 21*, Article 100151. https://doi.org/10.1016/j.ecotra.2019.100151
Tirachini, A., Hensher, D. A., & Rose, J. M. (2014). Multimodal pricing and optimal design of urban public transport: The interplay between traffic congestion and bus crowding. *Transp. Res. Part B Methodol., 61*, 33–54. https://doi.org/10.1016/j.trb.2014.01.003
Tirachini, A., Hensher, D. A., & Rose, J. M. (2013). Crowding in public transport systems: Effects on users, operation and implications for the estimation of demand. *Transp. Res. Part A Policy Pract., 53*, 36–52. https://doi.org/10.1016/j.tra.2013.06.005
Tirachini, A., Hurtubia, R., Dekker, T., & Daziano, R. A. (2017). Estimation of crowding discomfort in public transport: Results from Santiago de Chile. *Transp. Res. Part A Policy Pract., 103*, 311–326. https://doi.org/10.1016/j.tra.2017.06.008
Tirachini, A., Sun, L., Erath, A., & Chakirov, A. (2016). Valuation of sitting and standing in metro trains using revealed preferences. *Transp. Policy (Oxf), 47*, 94–104. https://doi.org/10.1016/j.tranpol.2015.12.004
Tirkolaee, E. B., Goli, A., Ghasemi, P., & Goodarzian, F. (2022). Designing a sustainable closed-loop supply chain network of face masks during the COVID-19 pandemic: Pareto-based algorithms. *J. Clean. Prod., 333*, Article 130056. https://doi.org/10.1016/j.jclepro.2021.130056
Tirkolaee, E. B., Torkayesh, A. E., Tavana, M., Goli, A., Simic, V., & Ding, W. (2023). An integrated decision support framework for resilient vaccine supply chain network design. *Eng. Appl. Artif. Intell., 126*, Article 106945. https://doi.org/10.1016/j.engappai.2023.106945
Vuchic, V. R. (2017). *Urban transit: Operations, planning, and economics*. John Wiley & Sons.
Wang, J., Lin, D., Zhang, Y., & Huang, S. (2022). An adaptively balanced grey wolf optimization algorithm for feature selection on high-dimensional classification. *Eng. Appl. Artif. Intell., 114*, Article 105088. https://doi.org/10.1016/j.engappai.2022.105088
Wang, M., Wang, L., Xu, X., Qin, Y., & Qin, L. (2019). Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: A case study in China. *J. Adv. Transp., 2019*, 1–12. https://doi.org/10.1155/2019/6090742

Wardman, M., & Whelan, G. (2011). Twenty years of rail crowding valuation studies: Evidence and lessons from British experience. *Transp. Rev., 31*, 379–398. https://doi.org/10.1080/01441647.2010.519127

Xu, W., Zhao, P., & Ning, L. (2018). Last train delay management in urban rail transit network: Bi-objective MIP model and genetic algorithm. *KSCE J. Civ. Eng., 22*, 1436–1445. https://doi.org/10.1007/s12205-017-1786-0

Yu, H., Gao, Y., Wang, L., & Meng, J. (2020). A hybrid particle swarm optimization algorithm enhanced with nonlinear inertial weight and Gaussian mutation for job shop scheduling problems. *Mathematics, 8*, 1355. https://doi.org/10.3390/math8081355

Zhang, F., & Liu, W. (2019). Responsive bus dispatching strategy in a multi-modal and multi-directional transportation system: A doubly dynamical approach. *Transp. Res. Procedia, 38*, 119–138. https://doi.org/10.1016/j.trc.2019.04.005

Zhang, L., Huang, J., Liu, Z., & Vu, H. L. (2021). An agent-based model for real-time bus stop-skipping and holding schemes. *Transportmetrica A Transp. Sci., 17*, 615–647. https://doi.org/10.1080/23249935.2020.1802363

Zhang, T., Li, D., & Qiao, Y. (2018). Comprehensive optimization of urban rail transit timetable by minimizing total travel times under time-dependent passenger demand and congested conditions. *Appl. Math. Model., 58*, 421–446. https://doi.org/10.1016/j.apm.2018.02.013