Pareto Optimization of Electric City Bus Scheduling

Zvonimir Dabčević*, Branimir Škugor, Joško Deur

University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, Croatia

*Corresponding author; e–mail: zvonimir.dabcevic@fsb.hr

ABSTRACT

This paper presents an approach to electric city bus scheduling optimization, which results in a Pareto frontier in two conflicting criteria being minimized: the number of buses required to serve predetermined routes and the excess of distance travelled (so-called deadhead distance). These criteria reflect the city bus fleet investment and operational costs, respectively. The sequential optimization strategy is executed in two phases: 1) finding the minimal number of buses, and 2) gradually incrementing the number of buses from the minimal one and minimizing the deadhead distance. Two optimization methods are proposed: mixed integer linear programming and genetic algorithm, where the former provides the optimal solution but it is limited to small-scale problems (fleets), while the latter can deal with large fleets but generally results in a nearly optimal solution. The optimization approach is demonstrated on a custom-generated dataset reflecting characteristics of real-world city bus transport systems.

KEYWORDS

City buses, battery electric vehicles, bus scheduling, Pareto optimization, charging constraints

1. INTRODUCTION

In recent years, electric vehicles (EVs) have grown in popularity due to their environmental and social benefits [1]. As the cost of battery electric vehicles drops down and the vehicle range increases, their share continuously grows [2]. The city bus transport system is a natural candidate for electrification due to the strong environmental benefits and predetermined and relatively short routes allowing for exploiting the end station fast charging for reduced battery capacity and cost [3]. Typical tasks associated with establishing a city bus fleet, such as line planning, vehicle scheduling [4], timetabling [5], and crew scheduling [6], become more complex when applied to e-buses when compared to conventional (Diesel) buses. This is because of restricted vehicle range and relatively long charging time.

City bus scheduling is a complex problem that falls under the vehicle scheduling problem (VSP) [7]. The essence of VSP is to assign routes and timetabled trips to vehicles with the aim of minimizing the fleet size and operational costs. There are two main categories of VSP depending on whether a single-depot (SDVSP, [8-10]) or a more complex multi-depot (MDVSP, [11]) transport system configuration is concerned. The optimization methods used include exact ones such as mixed integer linear programming [12] and heuristic approaches such as genetic algorithms [13]. Furthermore, in order to improve the overall cost effectiveness, there is a noticeable trend towards the utilization of mixed fleets, leading to the VSP with multiple vehicle types (MVTVSP, [14]). On the other hand, the focus can be shifted to fleets propelled by different types of fuel [15], where the Diesel bus fleet poses traditional challenges, while e-bus scheduling needs to address the distinct range and charging delay limitations. Static models of transport system have commonly been used [16], with the recent trend of employing dynamic models to handle the unpredictable nature of city traffic [17]. Another layer of complexity is introduced when partial recharging policy is concerned instead of more common, full recharging policy to provide additional operational savings [18, 19].

Building upon existing literature, reference [20] proposes a linear mathematical approach to electric bus scheduling, incorporating specific elements such as partial charging and charging at varying locations. This paper aims to extend upon this approach by adding constraints to ensure buses are fully charged by the end of each day (charge sustaining condition), while locally considering the state of energy for each bus, the power output of individual chargers, and the specific number of buses that can be charged at each station, underscoring the non-uniformity across buses and chargers. Moreover, a sequential optimization is concerned, where in addition to minimizing the total number of buses in operation, one minimizes the distance travelled without passengers between end stations of different routes (so-called deadhead distance), thereby offering a broader range of optimal solutions in terms of Pareto optimality.

2. PROBLEM DEFINITION

2.1. Electric city bus scheduling framework

This study attempts to tackle some major challenges of electric bus scheduling. The buses are allowed to operate on partial charges, and being charged both at main depots and designated route stops (end stations). Different charging stations can have different values of (i) maximum charging power and (ii) capacities to handle buses simultaneously. A general case of uninterrupted, full day operation satisfying the charging sustaining condition is concerned, as opposed to special cases based on, for instance, operation pauses for depot slow charging during night. It is assumed that the bus lines, timetables, location of charging stations and the number of charging spots per charger are predetermined.

Fig. 1 illustrates the e-bus scheduling optimization process, which starts by minimizing the number of electric buses needed to satisfy the predetermined timetables. The minimum fleet size is typically associated with a long deadhead distance, i.e. the total distance travelled by empty buses to switch between different lines (i.e., their end stations) to serve them and/or recharge on their charging stations. In other words, the minimum bus fleet investment cost is compromised by a higher operating cost (e.g., higher energy and maintenance costs). In order to obtain a set of optimal solutions in both criteria, i.e. to generate a Pareto frontier, the number of buses is incremented by one and a deadhead minimization problem is solved. The process continues until the deadhead distance saturates to its lower value.

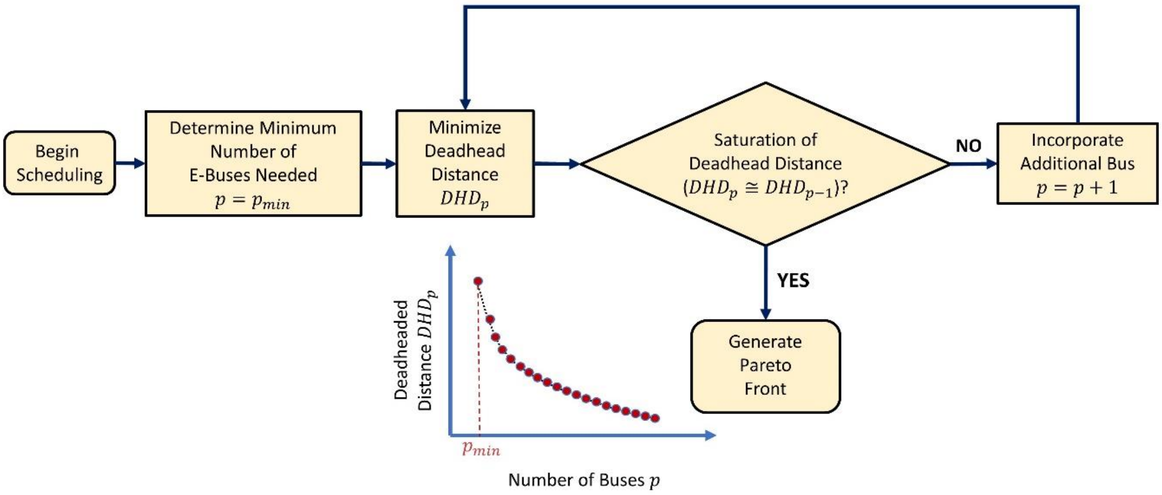

Fig. 1. Flowchart of e-bus scheduling sequential optimization process.

When optimizing the schedule, it is imperative to address both conventional scheduling constraints and those that are unique to electric vehicles. The conventional constraints encompass the following:

1. Every service trip is allocated to only one vehicle.
2. Each vehicle follows a feasible sequence of service trips, meaning the order and arrangement of trips for each vehicle must be logical and achievable within given time frames and operational conditions.

Electric vehicles bring additional constraints related to battery state-of-energy (SoE) limits:

1. The SoE must be high enough to complete the service trip or reach the nearest depot or charging station.
2. Buses can be recharged only at specific, predetermined charging station locations, and the battery cannot exceed its maximum value.
3. Only a limited number of buses can be recharged at a charging station at the same time (depending on the predetermined number of charging spots).
4. Each bus must finish its day with a fully charged battery, i.e. the final SOE must be equal to the initial SOE assumed to be at the maximum level (charge sustaining condition).

2.2. Formal problem formulation

Let N represent the set of service trips awaiting for scheduling, and let K represent the set of available vehicles, where every vehicle $k \in K$ carries a battery defined by its minimal and maximal SoE, SoE^k_{min} and SoE^k_{max}, respectively. For optimization to yield a feasible solution, the initial set of vehicles K should be set at a sufficiently high level. Distinct from the set N there are two specific points: D_0 and D_n. D_0 marks the depot starting position where vehicles initiate their routes, while D_n indicates the concluding point where vehicles conclude their service trips and revert back to the depot. Each service trip, denoted by an index i in the set N, possesses the following distinct attributes:

- starting time: s_i,
- duration: t_i,
- energy required: c_i,
- starting S_i and end location E_i.

Moreover, each trip i has a defined set of feasible succeeding service trips, $F(i)$, where a service trip j is deemed to feasibly succeed a service trip i if the condition $s_i + t_i + t_{ij} \leq s_j$ is satisfied. Here, t_{ij} marks the time needed to transit from the endpoint of trip i to the starting point of trip j, while the energy consumed during this transit period is quantified by c_{ij}. A symmetrical set, $B(i)$, lists trips j that can precede trip i: $s_j + t_j + t_{ij} < s_i$.

Additionally, a set R encompasses all charging stations. Each charging station $r \in R$, is distinguished by:

- Its location: situated either at starting or end stations of trips (S_i, E_i) or at the depot (D_0, D_n),
- Charging power q_r: (in Wh/per unit time) at which an electric bus is recharged,
- Charging spot capacity N_r: maximum number of buses a charger can handle at once, based on available charging spots.

The constants t_{ir} and t_{rj} stand for the time required to move from the end of a service trip *i* to a charger *r* and from the charger *r* to the service trip *j*, respectively. The energy costs associated with these routes are denoted by c_{ir} and c_{rj}.

Furthermore, each charger r possesses a charging event set, T^r, equivalent in count to the number of service trips. These charging events effectively provide a time discretization of the entire transport system by marking potential start or end times for charging. Specifically, the beginning time s_{rt} for charging event t from service trip i is defined as $s_{rt} = s_i + t_i + t_{ir}$, where charging events are organized chronologically by start times for each charger.

Moreover, to enhance optimization efficiency, service trips are aligned with charging events. To capture these relationships, specific sets are defined for each charger $r \in R$, each charging event $t \in T^r$ on charger r, and each service trip $i \in N$:

- $F_c(r, i)$ represents charging events that are initiated after trip i has reached charger r: $s_{rt} \geq s_i + t_i + t_{ir}$,
- $B_c(r, i)$ denotes charging events that occur before trip i reaches charger r,
- $F_i(r, t)$ indicates trips starting after charging event t: $s_i \geq s_{rt} + t_{ir}$,
- $B_i(r, t)$ captures trips ending before charging event t at charger r.

Based on the above foundational elements, several decision variables to be optimized have been introduced in the system:

- x_{ij}^k: Binary decision variable indicating whether the service trip $j \in N$ succeeds the service trip $i \in N$ using the vehicle $k \in K$, valid only if $j \in F(i)$.
- y_{irt}^k: Binary decision variable determining if the vehicle $k \in K$ recharges at the event $t \in T^r$ on the charger spot $r \in R$ after completing the service trip $i \in N$.
- z_{rtj}^k: Binary decision variable marking if the vehicle $k \in K$ undertakes the service trip $j \in N$ after charging at the event $t \in T^r$ on the charger $r \in R$.
- w_{rt}^k: Binary decision variable signifying if the vehicle $k \in K$ continues charging at the subsequent event $t + 1 \in T^r$ on the charger $r \in R$ after charging at charging event $t \in T^r$ on the same charger.

Fig. 2 visualizes the role of above decision variables and the overall scheduling mechanism.

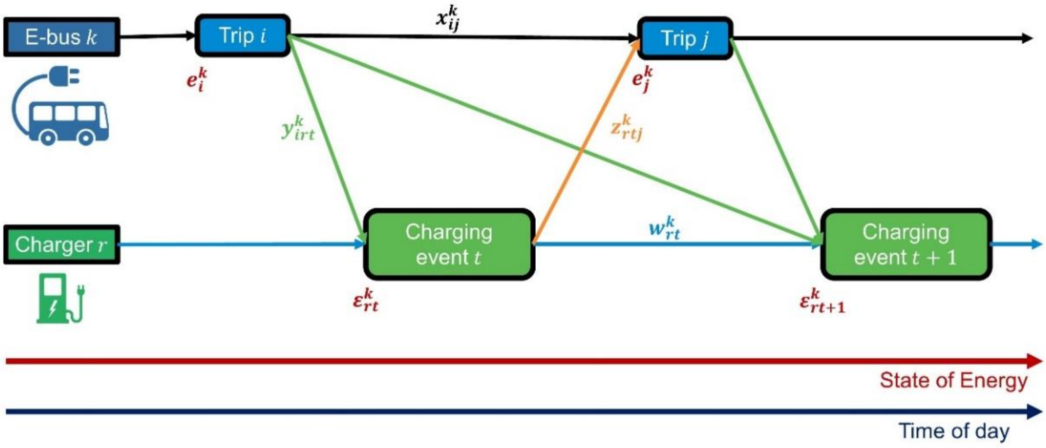

Fig. 2. Visualization of vehicle scheduling problem definition.

The battery SoE of k^{th} bus is defined by variables e_i^k and ε_{rt}^k. The variable e_i^k signifies the battery SoE of the bus just before starting service trip $i \in N$, ensuring the bus has enough charge for the trip. On the other hand, ε_{rt}^k represents the battery SoE before it begins charging at event $t \in T^r$ on charger $r \in R$. This not only indicates the battery depletion level but also, when compared to SoE upper limit SoE_{max}^k, helps determine the necessary charging amount and duration.

3. MIXED-INTEGER LINEAR POGRAMMING FORMULATION

By utilizing mathematical optimization based on the Mixed Integer Linear Programming Algorithm (MILP), a structured approach for solving the bus scheduling problem defined in Section 2 is proposed, which yields Pareto optimal solution in terms of minimization of total number of buses and deadhead distance. MILP solvers inherently possess certain capabilities, which include achieving optimal solution, ensuring solution convergence, and terminating automatically if they cannot satisfy the constraints [21]. In this study, the coin-or branch and cut solver, accessible via the PuLP library in Python is utilized to solve the MILP formulation.

3.1. Objective functions

To optimize the fleet usage while meeting the service demands, it is first aimed to minimize the number of electric buses deployed (see the second block in Fig. 1). The total number of buses in the system is determined by those dispatched from the depot, as buses are introduced exclusively from there (this does not restrict buses from shifting between lines during their journey). Therefore, the objective function achieves this by tracking the initial trip j of each bus k from the depot D_0, and is formulated as:

$$\min \sum_{k \in K} \sum_{j \in N} x_{D_0 j}^k. \tag{1}$$

The second objective aims to minimize the total deadhead distance (see the third block in Fig. 1), which sums the distances the buses travel outside of regular service. They include the distance for line switching between consecutive service trips i and j (d_{ij}), the distance to access a charger r from an i^{th} service trip endpoint (d_{ir}), and the distance from charger r to the next service trip j (d_{rj}) after charging is complete:

$$min \sum_{k \in K} \sum_{i \in N} \sum_{j \in F(i)} d_{ij} x_{ij}^{k} + \sum_{k \in K} \sum_{i \in N} \sum_{j \in F(i)} \sum_{t \in F_c(r,i)} d_{ir} y_{irt}^{k} + \sum_{k \in K} \sum_{r \in R} \sum_{t \in r} \sum_{j \in F_i(r,t)} d_{rj} z_{rtj}^{k}. \tag{2}$$

3.2. Vehicle scheduling constraints

To ensure that each service trip is served only by one bus, the following constraint is set:

$$\sum_{k \in K} \sum_{i \in B(j)} x_{ij}^{k} + \sum_{k \in K} \sum_{r \in R} \sum_{t \in B_c(r,j)} z_{rtj}^{k} = 1;\ \forall j \in N \tag{3}$$

Moreover, to guarantee a continuous flow of electric bus operations, a flow constraint is imposed for each service trip. This constraint mandates that after a bus completes a service trip or charging event, it needs to proceed to its next activity:

$$\sum_{i \in B(j)} x_{ij}^{k} + \sum_{r \in R} \sum_{t \in B_c(r,j)} z_{rtj}^{k} = \sum_{l \in F(j)} x_{jl}^{k} + \sum_{r \in R} \sum_{t \in F_c(r,j)} y_{jrt}^{k}\ ; \forall j \in N, \forall k \in K \tag{4}$$

For each charging station, there is a need to ensure that the number of vehicles charging simultaneously does not exceed its charging spot capacity N_r:

$$\sum_{k \in K} \sum_{j \in B_i(r,t)} y_{jrt}^{k} + \sum_{k \in K} w_{rt-1}^{k} \leq N_r;\ \forall r \in R, \forall t \in T^r \tag{5}$$

Moreover, when a bus arrives to charging station, it needs to depart from charging station after completing its specified charging event:

$$\sum_{i \in B_i(r,t)} y_{irt}^{k} + w_{rt-1}^{k} = \sum_{j \in F_i(r,t)} z_{rtj}^{k} + w_{rt}^{k};\ \forall r \in R, \forall t \in T^r, \forall k \in K \tag{6}$$

To ensure that the total number of deployed buses matches the predetermined fleet size p in the case of deadhead distance minimization step (Fig. 1), the following constraint is introduced:

$$\sum_{j \in N} x_{D_0 j}^{k} = p;\ \forall k \in K \tag{7}$$

3.3. Energy consumption constraints

First, every vehicle is set to begin the operating day with the battery charged at its upper limit:

$$e_{D_0}^{k} = SoE_{max}^{k};\ \forall k \in K \tag{8}$$

Furthermore, each bus must maintain its energy above the lower limit SoE_{min}^{k}, while considering its service trips, transfers, and routes to chargers whose SoE demands are specified by the constants c_i, c_{ij}, and c_{ir}, respectively:

$$e_i^k \geq SoE_{min}^k + c_i + \sum_{j \in F(i)} x_{ij}^k c_{ij} + \sum_{r \in R} \sum_{t \in F_c(r,i)} y_{irt}^k \, c_{ir} \, ; \forall i \in N, \forall k \in K \tag{9}$$

The following two constraints provides energy conservation between consecutive service trips, where the first one ensures that the bus does not exceed its battery capacity, while the second one guarantees that it retains enough energy for subsequent service trip:

$$e_j^k \leq e_i^k - x_{ij}^k\left(c_i + c_{ij}\right) + SoE_{max}^k\left(1 - x_{ij}^k\right); \forall j \in N, \forall i \in B(j), \forall k \in K \tag{10}$$

$$e_j^k \geq e_i^k - x_{ij}^k\left(c_i + c_{ij}\right) - SoE_{max}^k\left(1 - x_{ij}^k\right); \forall j \in N, \forall i \in B(j), \forall k \in K \tag{11}$$

The energy level of a bus, before embarking on a service trip, should reflect the balance of energy gained during its last charge and the energy consumed traveling from the last charging point to the trip start:

$$e_j^k \leq \varepsilon_{rt}^k + z_{rtj}^k\left(\left(s_j - t_{rj} - s_{rt}\right)q_r - c_{rj}\right) + SoE_{max}^k\left(1 - z_{rtj}^k\right); \forall j \in N, \forall r \in R, \forall t \in B_c(r,j), \forall k \in K \tag{12}$$

The following two constraints manage e-bus energy levels utilizing a large enough constant M for flexibility. The first constraint ensures that energy in a bus after charging remains within its maximum capacity when adjusted for the next trip. The M-term provides flexibility if the trip is not scheduled:

$$SoE_{max}^k \geq e_j^k + c_{rj} - Mq_r\left(1 - z_{rtj}^k\right); \forall r \in R, \forall t \in T^r, \forall k \in K, \forall j \in F_i(r,t) \tag{13}$$

The second constraint oversees energy levels during charging to ensure that the post-charge energy does not exceed the maximum one, while considering the next charging event, with the note that if the bus does not advance to its next charge, the M-term offers flexibility:

$$SoE_{max}^k \geq \varepsilon_{rt+1}^k - Mq_r\left(1 - w_{rt}^k\right); \forall r \in R, \forall t \in T^r, \forall k \in K \tag{14}$$

The following constraint ensures that a charged bus has adequate energy to travel from the charger to the next service trip:

$$e_j^k + c_{rj} + Mq_r\left(1 - z_{rtj}^k\right) \geq SoE_{min}^k + z_{rtj}^k c_{rj}; \forall r \in R, \forall t \in T^r, \forall k \in K, \forall j \in F_i(r,t) \tag{15}$$

The following two equations limit the energy level of a bus when it arrives at a charging station after its service trip. The first equation sets a maximum energy limit, ensuring the bus does not have more energy than expected after its trip:

$$\varepsilon_{rt}^{k} \leq e_{i}^{k} - y_{irt}^{k}(c_i + c_{ir}) + SoE_{max}^{k}\left(1 - y_{irt}^{k}\right); \forall r \in R, \forall t \in T^r, \forall k \in K, \forall i \in B_i(r,t) \quad (16)$$

The second equation establishes a minimum energy threshold to prevent the bus from arriving with an inadequate energy level:

$$\varepsilon_{rt}^{k} \geq e_{i}^{k} - y_{irt}^{k}(c_i + c_{ir}) - SoE_{max}^{k}\left(1 - y_{irt}^{k}\right); \forall r \in R, \forall t \in T^r, \forall k \in K, \forall i \in B_i(r,t) \quad (17)$$

Furthermore, the following constraint delineates the maximum energy that can be charged during an event, accounting for the time gap between consecutive charging events. Specifically, if two successive charging events have the same start time (influenced by the start times and durations of service trips), no energy is charged between them:

$$\varepsilon_{rt+1}^{k} \leq \varepsilon_{rt}^{k} + w_{rt}^{k}(s_{rt+1} - s_{rt})q_r + SoE_{max}^{k}\left(1 - w_{rt}^{k}\right); \forall r \in R, \forall t \in T^r, \forall k \in K \quad (18)$$

The constraint below sets a limit on the energy that can be charged during an event. It does so by considering the maximum energy that can be added before the next charging event starts on the same charger if the bus moves on to the next service trip after charging:

$$e_{j}^{k} + c_{rj} - \varepsilon_{rt}^{k} - SoE_{max}^{k}\left(1 - z_{rtj}^{k}\right) \leq (s_{rt+1} - s_{rt})q_r; \forall r \in R, \forall t \in T^r, \forall k \in K, \forall j \in F_i(r,t) \quad (19)$$

Furthermore, the constraints below ensure that the energy charged during a charging event remains non-negative. This is determined by the energy requirements on the subsequent trip or the next charging event.

$$e_{j}^{k} + c_{rj} - \varepsilon_{rt}^{k} + SoE_{max}^{k}\left(1 - z_{rtj}^{k}\right) \geq 0; \forall r \in R, \forall t \in T^r, \forall k \in K, \forall j \in F_i(r,t) \quad (20)$$

$$\varepsilon_{rt+1}^{k} - \varepsilon_{rt}^{k} + SoE_{max}^{k}\left(1 - w_{rt}^{k}\right) \geq 0; \forall r \in R, \forall t \in T^r, \forall k \in K \quad (21)$$

Finally, it is necessary to ensure that buses are fully charged at the end of the operating day. First, it is stipulated that each bus needs to undergo charging before being parked at the depot for the start of the next operating day:

$$\sum_{r \in R} \sum_{t \in T^r} z_{rtD_n}^{k} = 1;\ \forall k \in K \quad (22)$$

Next, it is ensured that each bus is fully charged when completing the daily operation:

$$\varepsilon_{rt}^{k} + \left(s_{rt+1}^{k} - s_{rt}^{k}\right)q_r = SoE_{max}^{k} - M(1 - z_{rtD_n}^{k}),\ \forall r \in R, \forall t \in T^r, \forall k \in K \quad (23)$$

Finally, the system ensures that the conclusion of the final charging event for each bus should occur early enough to allow the bus adequate time to be prepared for its initial trip on the subsequent day:

$$s^k_{rt+1} \leq s^k_j + t_{rj} + 1440 + M(1 - z^k_{rtD_n}),\ \forall r \in R, \forall t \in T^r, \forall j \in N, \forall k \in K \quad (24)$$

where the constant 1440 represents a full day measured in minutes.

3.4. Domain constraints

The domain constraints specify the permissible values for the decision variables and the energy state variables:

$$x^k_{ij} \in \{0,1\}; \forall k \in K, \forall i \in N \cup D_0 \cup D_n, \forall j \in F(i) \quad (25)$$

$$z^k_{rtj} \in \{0,1\}; \forall k \in K, \forall r \in R, \forall t \in T^r, \forall j \in F_i(r,t) \quad (26)$$

$$y^k_{irt} \in \{0,1\}; \forall k \in K, \forall i \in N, \forall r \in R, \forall t \in F_c(r,i) \quad (27)$$

$$w^k_{rt} \in \{0,1\}; \forall k \in K, \forall r \in R, \forall t \in T^r \quad (28)$$

$$\varepsilon^k_{rt} \geq 0; \forall k \in K, \forall r \in R, \forall t \in T^r \quad (29)$$

$$e^k_i \geq 0; \forall k \in K, \forall i \in N \quad (30)$$

4. GENETIC ALGORITHM APPROACH

Genetic algorithms (GA), inspired by biological evolution, offer a unique and general approach to optimization by simulating natural selection [22]. As such, they can handle complex and constraint-heavy MILP formulations. When compared to MILP algorithms, the advantage of GAs is that they can handle large-scale problems (e.g. a large number of trips, charging stations, constraints), while the disadvantage is they typically do not provide optimal solution (but rather converge in a nearly optimal solution, which is closer to the optimal solution if the number of iterations is set to be higher). Thus, in the context of e-bus scheduling optimization, the GA approach is employed as an alternative method for large-scale problems.

Formation of initial population of the GA entailed solving a relaxed MILP problem for 14 sub-formulations. All 14 sub-formulations utilize either objective function (1) or (2), depending on the optimization phase (see Fig. 1), while adhering to the vehicle scheduling constraints (3) to (7) and the domain constraints (25)-(30). Between the remaining constraints (8)-(24), two randomly selected constraints are added to each sub-formulation. Furthermore, each of these constraints is present in at least one sub-formulation. This approach is aimed at fostering a swifter convergence of the GA.

The GA employed the same solution representation as in the case of MILP formulation, where the binary decision variables x^k_{ij}, z^k_{rtj}, y^k_{irt}, and w^k_{rt} are optimized to obtain the final solution (see Fig 2). To ensure feasibility and optimality, the GA fitness function was carefully designed (see Algorithm 1 below). It assigns a lower value to solutions that use more vehicles and correspond to more constraint violations.

Algorithm 1: Fitness function of genetic algorithm

```
FUNCTION fitness_function(solution: Array) -> float:
    # Decompose the solution into individual decision variables
        x, y, z, w = reconstruct(solution)
    # Translate the values into corresponding bus events
        bus_events = generate_bus_events(x, y, z, w)
    # Initialize violations counter
        number_of_violations = 0
    # Check for scheduling constraints
        number_of_violations = CHECK_VEHICLE_SCHEDULING_CONSTRAINTS(bus_events)
    # Initialize SoE penalty
        soe_penalty = 0
    # Compute energy details for each bus
        FOR each_bus in bus_events:
            # Initialize the State of Energy
                soe = MAXIMUM_SOE
            FOR event in each_bus:
                # Update SoE based on the event
                    soe = UPDATE_SOE(soe, event)
                # Adjust soe_penalty if soe is negative
                    IF soe < 0:
                        soe_penalty += ABS(soe)
            # Adjust soe_penalty if the final soe is not 100
                IF soe != 100:
                    soe_penalty += (100 - soe)
    # Compute the fitness value
        penalty = 1 / num_of_buses
        P = penalty * (soe_penalty + number_of_violations)
        fitness = 1 - (num_of_vehicles / num_of_buses) - P

    RETURN fitness
```

The GA algorithm is set to run for 5,000 generations, and four mating parents were designated for each generation. The steady-state selection method is chosen for parent selection, promoting a gradual and consistent replacement of individuals in the population. A two-point crossover technique is employed, where two random crossover points are determined and genes between these points are swapped between two parent individuals. The mutation approach is of inversion type, where a selected gene segment is reversed to introduce diversity and 10% of genes are subjected to mutation. To maintain continuity, four parents from the current generation were retained for the subsequent one. The GA was implemented using Python PyGad library.

5. OPTIMIZATION RESULTS

5.1. Scenario generation and data description

A detailed system scenario has been developed to replicate the complexity of a city bus transport system [23]. For the purpose of verifying the MILP optimization algorithm (Subsection 5.3), a scenario involving 50 trips distributed across six distinct bus lines has been set up. Each line is delineated by two endpoints (start and final) selected from a pool of six possible end stations, resulting in some lines sharing the same end stations. Within this setup, three chargers are randomly placed among these six end stations. The electric buses are set to have a battery with the capacity of 100 kWh, while the chargers provide power of 1.74 kWh/min, serving one bus at a time. The trips are scheduled to begin randomly throughout the day, with intervals of 10 to 30 minutes between consecutive trips. The trip duration ranges from 10 to 50 minutes, and the buses energy consumption rate randomly varies in the range from 0.8 to 1.2 kWh/min. The deadhead distance is set to randomly vary in the range from 10 to 50 km.

The GA optimization algorithm has been verified and compared with the MILP algorithm for a set of scenarios having the number of trips setting in the range from 5 to 500 and maintaining the remaining foundational input parameters.

5.2. MILP optimization

The MILP methodology depicted in Figure 1 and elaborated in Section 3 has initially been applied to the case of conventional bus fleet. In this case, the problem formulation was reduced by removing the charging elements and constraints. More specifically, the scheduling of conventional buses was carried out by using the objective functions (1) and (2), and the constraints (3), (4), and (7), while solely the decision variable x_{ij}^{k} was involved. Subsequently, the MILP optimization has been conducted for the target case of e-bus fleet, where the full problem formulation of Section 3 is used.

Fig. 3 shows the comparative Pareto frontiers obtained for the cases of conventional and electric city bus fleets, where the MILP algorithm is used along the basic scenario including 50 trips. Evidently, this system of relatively small size can be handled by only 5 conventional buses, in which case the deadhead distance equals to almost 550 km (Fig. 3a) or around 20% of the total distance made when the number of buses is large enough to eliminate the deadhead distance (at least 32 buses). Due to the range and charging constraints, the e-bus fleet requires higher minimum number of buses compared to the conventional fleet (6 vs. 5, Fig. 3a) with the deadhead distance being reduced to some extent (from 20% to 17.5%, Fig. 3b), and the Pareto frontier generally shifts to higher values of the two objectives. However, as the number of electric buses increases (to 22), Pareto front approaches that of the conventional fleet. This is because for the large enough fleet, the charging system is efficient enough not to disturb the bus scheduling.

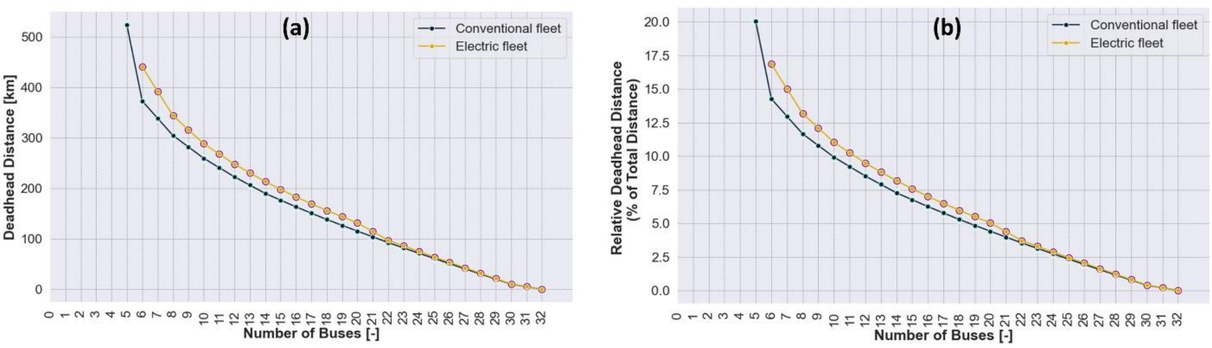

Fig. 3. Comparative Pareto frontiers obtained by MILP approach in the case of conventional and electric bus scheduling optimization.

5.3. Comparative analysis of MILP and GA optimization results

A comparative analysis of the MILP and GA optimization results is presented in Fig. 4 for the case of minimizing only the total number of buses criterion (1). Both conventional and electric fleets are considered in the MILP case, while only the electric fleet is concerned in the GA case. For the sake of clear comparison of the two approaches, the computation time of MILP algorithm has been restricted to match that of the GA for the considered size of the transport system (i.e. the number of trips, the x-axis in Fig. 4).

Fig. 4 indicates that the computational inefficiency of the MILP algorithm progressively grows with the rise of system size, i.e. number of trips (not that the execution time axis is as given as logarithmic). Moreover, as the system size expands, the MILP algorithm for electric fleet often fails to produce any feasible solution within the allotted time, as evidenced by the missing solutions for 100, 200, and 500 trips in Fig. 4. In contrast, the GA consistently yields feasible solutions for these larger trip numbers where MILP falls short. While the GA tends to provide sub-optimal results (e.g. for 50 trips, Fig. 4), it aligns with the MILP optimal solution for smaller-scale systems (same solution found for 10 and 25 trips) and consistently follows the solutions yielded by the MILP algorithm for large-size conventional fleet. Hence, the GA emerges as is deemed to be a more suitable choice for large-scale e-bus transport systems than the MILP algorithm.

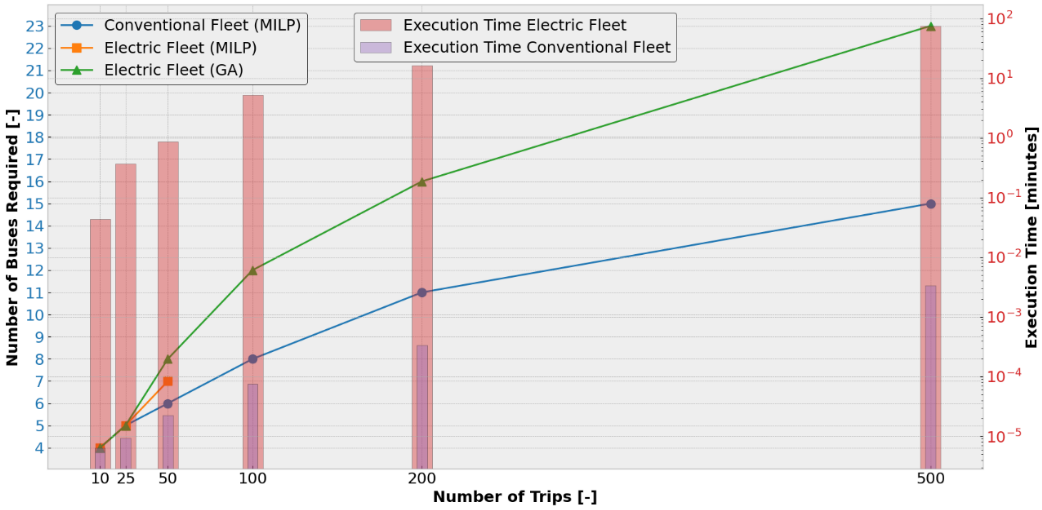

Fig. 4. Comparison of minimal numbers of buses obtained by using MILP and GA approaches for various sizes of city bus transport system.

6. CONCLUSION

A multi-objective electric city bus scheduling optimization approach has been proposed. The optimization problem was formulated as a Mixed Integer Linear Programming (MILP) problem. It was enriched compared to the available literature with several features and constraints such charge sustaining condition and inherent variability of buses and charger parameters. Also, a sequential multi-objective optimization was employed, which in addition to minimizing the total number of buses seeks to minimize the deadhead distance (the distance travelled outside of regular service). This combined approach offers a broader spectrum of Pareto optimal solutions as a trade-off between investment and operating costs.

The optimization problem has first been solved by using a MILP solver. This approach provides convergence to optimal solution, but it is limited to small-scale transport systems due to its computational complexity (particularly for the electrified ones due to the more complex constraint formulation). The optimization results reveal the Pareto front in the number of buses and deadhead distance objectives. The Pareto front is shifted to higher values of objectives (i.e., lower performance) in the case of e-bus fleet when compared to the conventional fleet, which is due to the e-bus range and charging restrictions. Yet, as the number of buses rises, the performance disparity between the two systems diminishes, which is because the range and charging restrictions become less relevant for the expanded fleet size.

The MILP optimization problem has then been solved by using a Genetic Algorithms (GA). The GA has demonstrated adaptability across different transport system scales. That is, the GA aligns with MILP outcomes in small-scale settings and remains reliable, although nearly optimal, in mid/large-scale settings, thus making it a more suitable choice for solving large-size e-bus scheduling problems.

ACKNOWLEDGMENT

It is gratefully acknowledged that this work has been supported by the European Commission through Horizon 2020 Innovation action project OLGA ("hOListic Green Airport") under the Grant Agreement No. 101036871.

REFERENCES

[1] A. Braun, W. Rid, "Energy consumption of an electric and an internal combustion passenger car: A comparative case study from real-world data on the Erfurt circuit in Germany", *Transp. Res. Procedia*, Vol. 27, pp. 468–475, 2017.

[2] G. Kalghatgi, "Is it really the end of internal combustion engines and petroleum in transport?", *Appl. Energy*, Vol. 225, pp. 965–974, 2018.

[3] J. Topić, J. Soldo, F. Maletić, B. Škugor, J. Deur, "Virtual Simulation of Electric Bus Fleets for City Bus Transport Electrification Planning", *Energies*, Vol. 13, No. 13, pp. 23, 2020.

[4] S. S. G. Perumal, R. M. Lusby, J. Larsen, "Electric bus planning & scheduling: A review of related problems and methodologies", *European Journal of Operational Research*, Vol. 301, No. 2, pp. 395-413, 2022.

[5] J. Teng, T. Chen, W. Fan, "Integrated Approach to Vehicle Scheduling and Bus Timetabling for an Electric Bus Line", *Journal of Transportation Engineering, Part A: Systems*, Vol. 146, No. 2, 2019.

[6] S. S. G. Perumal, T. Dollevoet, D. Huisman, R. M. Lusby, J. Larsen, M. Riis, "Solution approaches for integrated vehicle and crew scheduling with electric buses", *Computers & Operations Research*, Vol. 132, 2021.

[7] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research*, Vol. 35, No. 2, pp. 254–265, 1987.

[8] C. S. Orloff, "On general routing problems: Comments", *Networks*, Vol. 6, pp. 281-284, 1976.

[9] S. Raff, "Routing and scheduling of vehicles and crews: The state of the art", *Computers & Operations Research*, Vol. 10, pp. 63-211, 1983.

[10] B. Gavish, E. Shlifer, "An approach for solving a class of transportation scheduling problems", *European Journal of Operational Research*, Vol. 3, No. 2, pp. 122-134, 1979.

[11] N. Kliewer, T. Mellouli, L. Suhl, “A time-space network based exact optimization model for multi-depot bus scheduling”, *European Journal of Operational Research*, Vol. 175, No. 3, pp. 1616-1627, 2006.

[12] K. Gkiotsalitis, C. Iliopoulou, K. Kepaptsoglou, “An exact approach for the multi-depot electric bus scheduling problem with time windows”, *European Journal of Operational Research*, Vol. 306, No. 1, pp. 189-206, 2023.

[13] B. Ombuki-Berman, F.T. Hanshar, “Using Genetic Algorithms for Multi-depot Vehicle Routing”, *Bio-inspired Algorithms for the Vehicle Routing Problem*, Vol. 161, pp. 77-99, 2009.

[14] S. Hassold, A. Ceder, "Public transport vehicle scheduling featuring multiple vehicle types", *Transportation Research Part B: Methodological*, Vol. 67, pp. 129-143, 2014.

[15] N. Reuer, N. Kliewer, L. Wolbeck, "The electric vehicle scheduling problem: A study on time-space network based and heuristic solution approaches", *Proceedings of the 13th conference on advanced systems in public transport (CASPT)*, Rotterdam, 2015.

[16] C. Zhu, X. Chen, “Optimizing Battery Electric Bus Transit Vehicle Scheduling with Battery Exchanging: Model and Case Study”, *Procedia - Social and Behavioral Sciences*, Vol. 96, pp. 2725-2736, 2013.

[17] B. A. Kumar, G. H. Prasath, L. Vanajakshi, “Dynamic Bus Scheduling Based on Real-Time Demand and Travel Time”, *International Journal of Civil Engineering*, Vol. 17, No. 9, pp. 1481-1489, 2019.

[18] M. Keskin, B. Çatay, “Partial recharge strategies for the electric vehicle routing problem with time windows”, *Transportation Research Part C: Emerging Technologies*, Vol. 65, pp. 111-127, 2016.

[19] M. Jiang, Y. Zhang, Y. Zhang, "Multi-Depot Electric Bus Scheduling Considering Operational Constraint and Partial Charging: A Case Study in Shenzhen, China", *Sustainability*, Vol. 14, 255, 2022.

[20] M. Janovec, M. Koháni, "Exact approach to the electric bus fleet scheduling", *Transportation Research Procedia*, Vol. 40, pp. 1380-1387, 2019.

[21] M. Conforti, G. Cornuéjols, G. Zambelli, “Integer Programming (Graduate Texts in Mathematics, 271)”, *Springer*, 2014.

[22] W. Banzhaf, E. Goodman, L. Sheneman, L. Trujillo, B. Worzel (Eds.), “Genetic Programming Theory and Practice XVII (Genetic and Evolutionary Computation)”, *Springer*, 2020.

[23] M. Rogge, E. van der Hurk, A. Larsen, D. U. Sauer, “Electric bus fleet size and mix problem with optimization of charging infrastructure”, *Applied Energy*, Vol. 211, pp. 282-295, 2018.