Search Space Reduction-Supported Multi-objective Optimization of Charging System Configuration for Electrified City Bus Transport System

Daria Matković*, Jakov Topić, Branimir Škugor, Joško Deur
Faculty of Mechanical Engineering and Naval Architecture
University of Zagreb, Zagreb, Croatia
e–mail: daria.matkovic@fsb.hr

ABSTRACT

Significant research efforts have been recently devoted to city bus transport system electrification and its integration into the smart city framework. A key challenge is to determine the optimal charging infrastructure regarding (i) the selection of terminals to be equipped with chargers and (ii) the number of chargers to be installed at each terminal while satisfying the buses' schedule and minimizing the investment cost. To address this issue, the paper proposes a charging system configuration optimization method based on a genetic algorithm, which is aimed at minimizing the number of charging terminals, the number of chargers per terminal and charging-related cumulative time delay of the driving missions. Furthermore, each route needs to be covered with at least one charging terminal to meet the transport system charging sustainability condition. To reduce the wide-range search space of the optimization algorithm and facilitate convergence, a search space reduction is conducted by determining the best charging terminal candidates based on a modified greedy set-cover algorithm.

KEYWORDS

Electric vehicles, charging schedules, charging terminals, e-mobility, optimization, genetic algorithm, greedy algorithm

1. INTRODUCTION

Integrating fully electric buses into the city transport system substantially reduces air and noise pollution, lowers the bus fleet exploitation cost as the energy and maintenance cost sinks compared to conventional fleets, and improves passenger comfort and overall satisfaction. However, there are several drawbacks of fleet electrification such as lack of charging terminals, long charging time and relatively low driving range, and high vehicle cost including the cost of possible battery replacement. Many researchers have studied the charging infrastructure optimization problem for electric vehicles in general, focusing on locating the fast and slow chargers and sizing the charging terminals. For example, reference [1] presents a method of finding the number and locations of fast-charging terminals to maximize long-distance driving missions' completion. The problem of low charger utilization related to inferior charging terminal placement is addressed in [2]. The same reference proposes charging location optimization while considering home charging accessibility, thus minimizing the number of trips that could not be made due to the charging constraints. An approach where the placement and number of distinct charging terminals are optimized and analysed within an active distribution network is considered in [3]. A lightning search algorithm for optimal location and sizing of fast charging terminals is proposed in [4] while

* Corresponding author

considering transportation grid power losses and investment costs. In [5] a model is built to predict the total number of electric vehicles and calculate the size of charging terminals to optimize their distribution and make them more balanced through the years in which the number of electric vehicles might change. The study is focused on future charging system maintainability based on the predicted number of electric vehicles. Reference [6] presents an approach of finding the optimal number of chargers and their locations while considering the demand. The optimization problem is solved by using a genetic algorithm with constraints related to travelled distance per day, charging time, and power limitations.

This paper deals with city bus fleet charging configuration optimization resulting in the optimal selection of charging terminals and the number of chargers installed on those terminals. Charging terminals are selected for the predefined number of buses (equal to that of the conventional fleet) and the given bus type including the battery capacity. The distribution of buses on different routes is also predefined, where the buses are not considered to be shifted from one route to another. GPS data retrieved from the existing (Diesel) bus fleet are used to build up a transport model, which provides velocity and terminal dwell time data for every route and direction at every time interval of the day. Also, the transport model includes energy consumption maps for e-buses, which are determined by simulating a physical e-bus model over realistic high-resolution driving cycle data. The research represented in this paper is the follow-up to [7] where a simulation tool for planning city bus transport electrification is represented and [8] focused on a Markov-chain-based method synthesis of the high-resolution driving cycles. The driving cycle relates to the peak day in view of traffic load and weather conditions, thus concerning the worst-case scenario of powertrain and HVAC (heating, ventilation and air-conditioning) system energy consumption, respectively. Charging configuration optimization is based on the pilOPT multi-objective genetic algorithm provided in the modeFRONTIER optimization environment. To reduce the input space represented by the number of chargers on different terminals, the modified greedy set-cover algorithm is developed and used in pre-optimization. Finally, the optimized charging configurations are compared with near-optimal charging configurations found through expert knowledge.

The main contributions of the paper include: (i) developing a minute-based macro-simulation tool for simulating city bus driving missions and related energy consumption for the defined driving schedule and a specified charging configuration, (ii) applying a modified greedy set-cover algorithm for determining charging terminal candidates used to reduce input search space of charging configuration optimization by eliminating unpromising charging terminals, and (iii) proposing a macro-simulation model-based optimization framework for obtaining optimal charging configuration characterized by the minimum number of charging terminals and chargers.

The remaining part of this paper is structured as follows. Section 2 provides an overview of the optimization framework. Section 3 describes the transport system macro-simulation model used to simulate the city bus fleet. Greedy algorithm-based optimization of charging locations used for search space reduction is described in Section 4. Section 5 presents the overall charging system configuration optimization algorithm, with the results given and discussed in Section 6. Section 7 presents concluding remarks and outlines future work opportunities.

2. OPTIMIZATION FRAMEWORK

This section provides an overview of the optimization framework shown in Figure 1 and aimed to determine the optimal charging configuration for a city bus transport system. The optimization framework consists of (i) a modeFRONTIER optimization tool based on the pilOPT algorithm and (ii) a macro-simulation model of the transport-energy system that is run in Python in every optimization iteration. First, DOE (Design of Experiments) is defined, which sets the initial charging configuration and is generated by pilOPT algorithm. The number of chargers at every terminal is denoted by Ch_i, $i = 1,...,n$, where the subscript i denotes the terminal index and n is the number of terminals. It is requested that the minimum number of chargers per charging terminal is $N_{ch,min} = 2$, while the maximum number of chargers is equal to the number of buses N_b coming to the terminal. This is implemented through the constraint $N_{ch,min} \leq Ch_i \leq N_b$, with the note that if a terminal has no chargers installed, the number of chargers Ch_i is set to 0. The second constraint, $RC_r \geq 1$ is related to the number of charging terminals RC_r placed at route $r \in [1, N_r]$, where N_r is the number of routes. There are three objective functions to be minimized, which are denoted by J_i, $i \in [1, 3]$ (see Fig. 1), and which relate to the total number of chargers (N_c), the total number of charging terminals (N_{ct}) and the total time delay for all buses' departures during the single-day operation (D_{tot}) affected by charging and the lack of buses, where N_c and N_{ct} are calculated directly from charging configuration, and D_{tot} is obtained from macro-simulation. The charging configuration is represented by the set $[Ch_i,...,Ch_n]$, i.e. by the number of installed chargers at every terminal. The macro-simulation model parameters include the driving schedule set, S, a deadzone time, T_{dz}, charging power, P_{ch}, and e-bus battery capacity, E_{batt}. The deadzone time T_{dz} is the minimum time the bus should spent at the charging terminal to start charging, representing the time needed to park to the charger spot and plug in the charger.

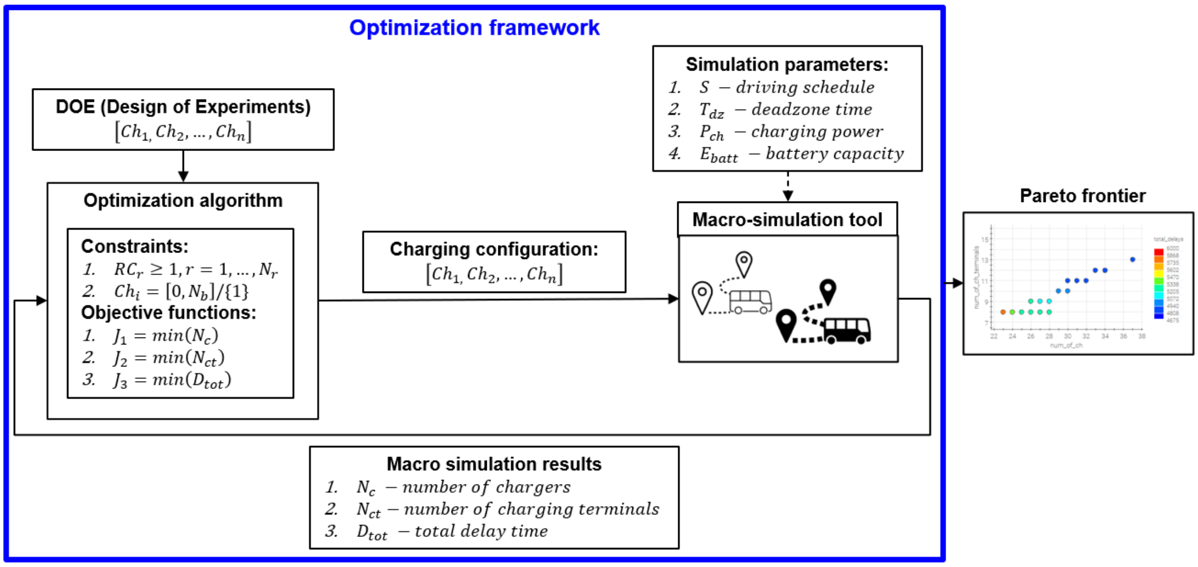

Figure 1 Block diagram of optimization framework used for optimizing the charging configuration

The optimization algorithm iteratively generates a charging configuration used as an input to the macro-simulation model, which simulates the driving missions over the peak-load day based on the specified simulation parameters. The simulation results are used in the optimization algorithm to generate a new charging configuration, according to the constraints and objective functions, determined in a way that minimizes the number of charging terminals, the total number of chargers, and the total bus delay time affected by prolongated bus departure due to the charging dwell time.

3. TRANSPORT SYSTEM MACRO-SIMULATION MODEL

The macro-simulation model describes the city bus transport and energy system on a daily basis and with a time resolution of one minute. It simulates a bus fleet containing 303 buses allocated to 29 routes and 25 terminals in a part of the city of Jerusalem. Each bus is assumed to operate only on one of the routes. Simulation outputs are post-processed to obtain detailed transport analysis data for every route and bus, i.e., the dwell time at each terminal, the delay time of driving missions, and a variety of metrics regarding the battery state of charge (SoC), energy-charged, and bus utilization.

Figure 2 overviews the macro-simulation model in the form of a flowchart. In every sampling instant (with a sampling time of 1 minute) the algorithm checks the scheduled departure and arrival time for driving missions. A driving mission is allocated to the bus with the largest battery SoC (from the set of buses assigned to that route), while considering the constraint that the bus cannot leave the terminal (if equipped with chargers) if $SoC < 20\%$. If there is no bus with $SoC \geq 20\%$ at the charging terminal, the departure is postponed, i.e. a delay occurs. The driving mission travel time and energy consumption are obtained from the corresponding maps, which were determined (i.e., pre-processed) from the driving cycle data and e-bus physical micro-simulation and stored in a database over different routes and on an hourly basis. The bus battery SoC is updated at the end of a driving mission in accordance with the energy consumption of the driving mission (including the HVAC system energy consumption which depends on the external ambient conditions, i.e. time of the day). After the bus arrives at the terminal, a simple heuristic charging management algorithm is executed.

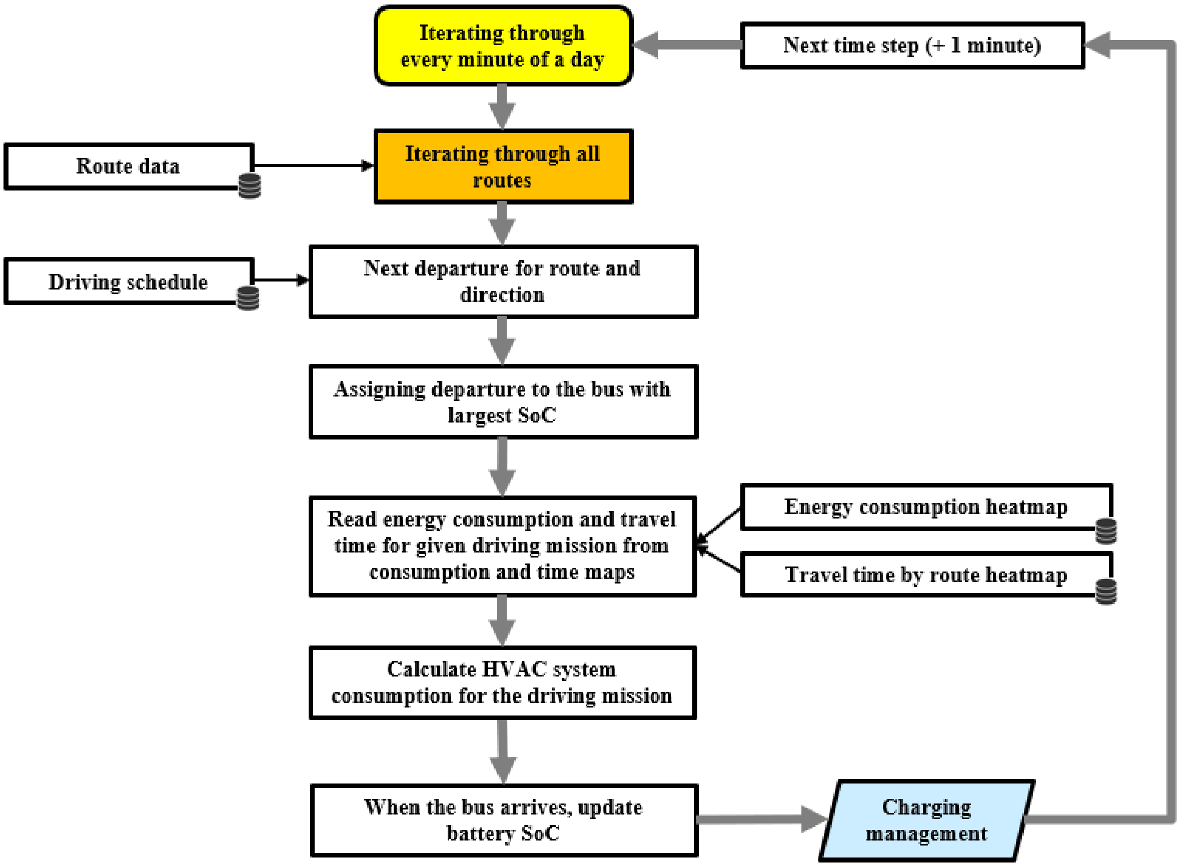

Figure 2 Flowchart of macro-simulation model

The charging management algorithm is described by the flowchart shown in Figure 3. First, when the bus arrives at the terminal equipped with chargers, it gets connected to the unused charger if there is any. If all chargers are occupied, the bus with the largest SoC gets disconnected, but only if its SoC is greater than the SoC of the arrived bus. When the battery is fully charged ($SoC = 95\%$) or the bus with $SoC \geq 20\%$ is scheduled to departure, bus disconnects from the charger, and the bus with the lowest SoC connects.

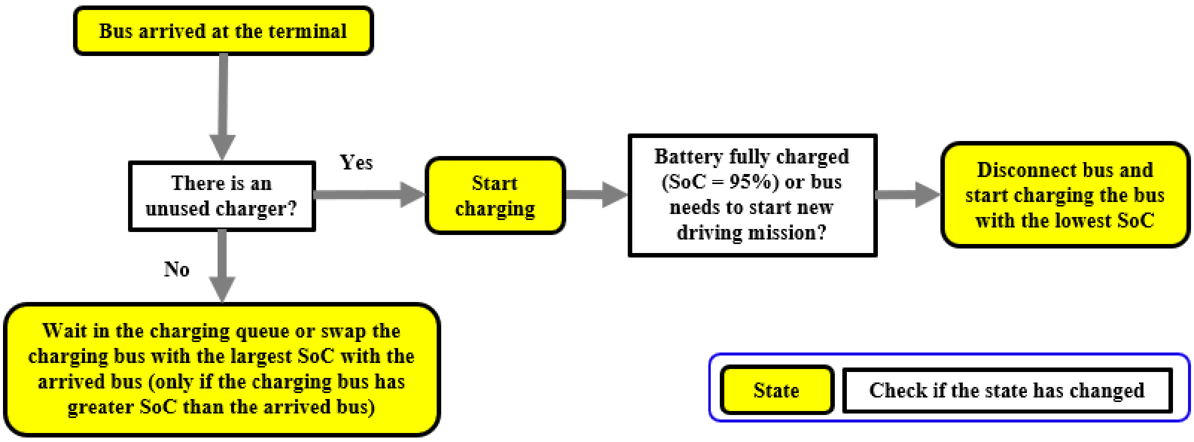

Figure 3 Flowchart of heuristic charging management algorithm

4. “GREEDY ALGORITHM” BASED OPTIMIZATION OF CHARGING LOCATIONS

The bus transport system includes 25 terminals, which may or may not include chargers. Since there are two options for every terminal, there are 2^{25} = 33,554,432 charging configuration combinations. Thus, it would be very time-consuming to manually find viable charging configuration combinations to be used in full optimization in Section 5, while considering route coverage constraint meaning that every route has at least one charging terminal. Therefore, a modified greedy set-cover algorithm for charging terminal candidates’ optimization is proposed in this section. The search space is reduced by decreasing the number of input variables, in this case the charging terminal candidates, otherwise set to all terminals.

4.1. Charging candidate problem

The charging terminal candidate problem is defined as finding the minimum number of charging terminals while considering the route coverage constraint. Since there may be more distinct configurations with the same minimum number of charging terminals the charging candidate problem should cover all those minimum configurations. For this purpose, a modified greedy set-cover algorithm is designed. The final reduced input space or reduced charging candidate set is determined as the union of charging terminals in all configurations obtained from the modified greedy set-cover algorithm.

The charging candidate problem can be reduced to the well-known NP-complete (nondeterministic polynomial) problem called set-cover. Given the set and subsets of the set, the objective is to find the set of subsets containing the minimum number of subsets where the union of those subsets is equal to the set. Reducing the charging candidate problem to set-cover means that subsets in the set-cover problem would be related to terminals containing the routes related to the terminal. As already mentioned, the set-cover problem is NP-complete, which means that it is NP and NP-hard and does not have the exact algorithm that solves it.

The set-cover problem can be approached by many approximation algorithms, e.g. the branch and bound algorithm, layering, and the greedy algorithm. In this work, a greedy set-cover approximation for charging candidate optimization is used.

Since the greedy algorithm for the set-cover problem presented in [11] returns only one solution, while the charging candidate problem should ultimately return all charging configurations with the minimum number of charging terminals satisfying the route coverage constraint, a modified greedy algorithm is proposed. The details are given in the next subsection.

4.2. Modified greedy set-cover algorithm for charging candidate optimization

The modified greedy set-cover algorithm is used for reducing the number of charging terminals and is motivated by the greedy set-cover problem represented in [11]. The mathematical formulation of the set-cover problem is as follows:

$$\text{Given the elements of } U = \{u_1, u_2, \dots, u_n\},$$

$$\text{Subsets } S_1, S_2, \dots, S_k \subseteq U, \tag{1}$$

$$\text{weights } w_1, w_2, \dots, w_k,$$

$$\text{find } I \subseteq \{1, 2, \dots, k\},$$

$$\text{that min } \sum_{i \in I} w_i, \tag{2}$$

$$\text{s.t.} \bigcup_{i \in I} S_i = U.$$

The greedy set-cover algorithm is shown in Algorithm 1 below and described in [11]. It executes in the following steps: (i) initializes empty array of selected subsets $S_1, S_2, \dots, S_k$, (ii) iterates while the array of selected subsets does not contain all elements from set U and, in every iteration, it selects the subset with the smallest cost. The cost function is the ratio between the subset cost and the number of elements contained in a subset, not added in the array of selected subsets. The subset weight is predefined, and it depends on the system, where sometimes it may be the same for all subsets, but it may also be diverse.

```
Algorithm 1: Greedy set cover algorithm
  Data: U, S_i, ..., S_n, w_i, ..., w_n
  Result: C = ∅
  while C ≠ U do
      D ← U\C;
      for i ← 1 to n do
          c_i ← |S_i ∩ D| / w_i;
      end
      S_i* ← max(c_1, ..., c_n);
      C ← C ∪ S_i*;
  end
  return C
```

Algorithm 1 Greedy set-cover algorithm

The proposed, modified greedy set-cover algorithm is a version of the original algorithm, where modifications relate to ultimately returning all possible combinations of configurations with the minimum number of charging terminals and adapting the cost function according to the bus transport system. The mathematical formulation of the modified set-cover problem is the same as for the set-cover problem shown in equations (1) and (2), while the programming implementation has a few modifications, as given by Algorithm 2 below.

Algorithm 2 executes in a dynamically chosen number of iterations, and it runs as follows: (i) it initializes an empty set of generated configurations, (ii) starts iteration and stops when no new combination or configuration is found for at least 20 iterations, and (iii) in every step it generates weights for every subset; in this case, the subset is represented as a set of routes covered with every terminal, (iv) for previously generated weights, the algorithm iterates and builds new configuration based on the cost function that prioritizes the terminals that cover more routes in total and more of the uncovered routes, scaling it with the weights $w_1, w_2, \dots, w_k$ (described later in more detail), (v) when no configuration is generated for at more than 20 iterations, the algorithm returns charging configurations with the minimum number of charging terminals. The cost function is the ratio of the sum of the total number of routes covered by the terminal and uncovered routes divided by the terminal weight value. The weights are generated by using Gaussian distribution with the mean value μ equal to 3 and the standard deviation σ equal to 1. These values are empirically chosen to introduce randomness to the cost function, i.e. to generate distinct weights $w_1, w_2, \dots, w_k$, in every iteration resulting in more charging combinations. In the case of the same weights in every iteration, the algorithm would result in one configuration combination.

Algorithm 2: Adapted greedy set cover algorithm

```
Data: U, S_i, ..., S_n
Result: C = ∅
i ← 0;
cs_min ← n
C_min = ∅;
while i < 20 do
    w_i, ..., w_n ← N(μ, σ^2);
    I = ∅;
    while I ≠ U do
        D ← U\I;
        for i ← 1 to n do
            c_i ← |S_i ∩ D| / w_i;
        end
        S_i* ← max(c_1, ..., c_n);
        I ← I ∪ S_i*;
    end
    if I ∉ C then
        C ← [C, I];
        cs_min = min(cs_min, |I|);
    end
end
for I ∈ C do
    if |I| = cs_min then
        C_min ← [C_min, C];
    end
end
return C_min
```

Algorithm 2 Modified greedy set-cover algorithm

4.3. Charging candidate optimization results

Figure 4 shows the assignment of terminals to routes for the considered city bus system. The goal is to find the minimum charging configuration combinations satisfying the route coverage constraint (each route is covered by at least one charging terminal).

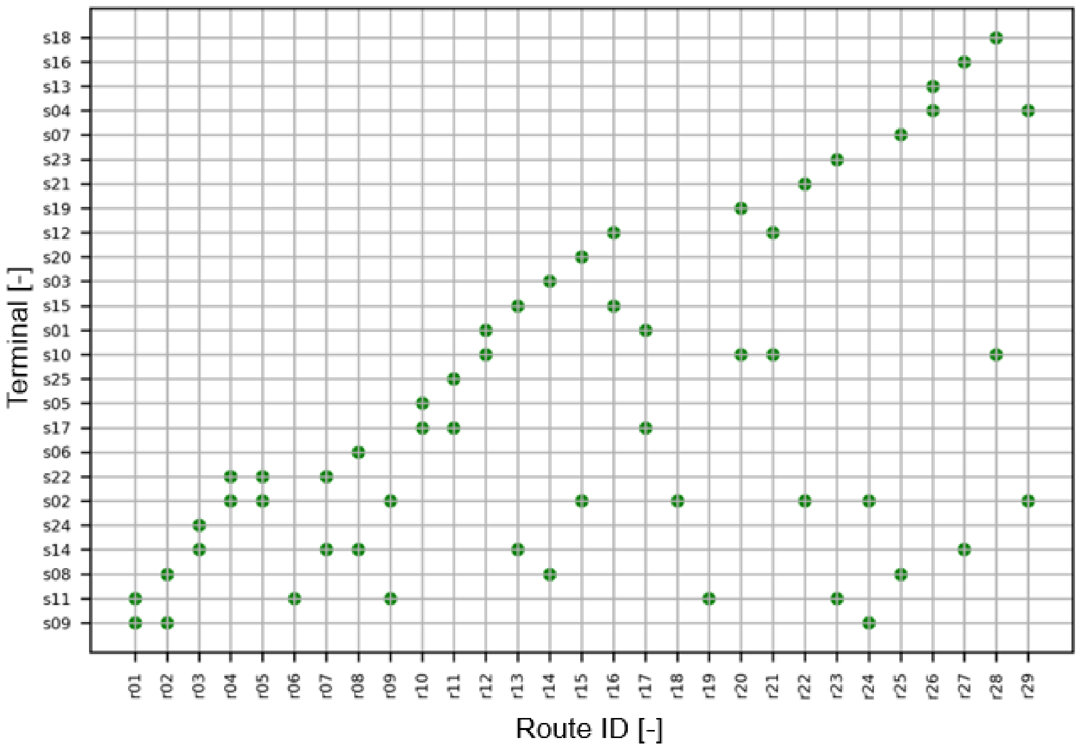

Figure 4 Route and belonging terminals of the considered city bus system

Table 1 shows combinations of configurations found (i) “manually” based on expert knowledge and (ii) using the proposed modified greedy set-cover algorithm. Only one combination of charging configurations with the minimum number of charging candidates is found manually, while the modified greedy set-cover algorithm manages to find four distinct combinations, including the manually found one. The reduced input space-based optimizations (Section 5) consider the union of charging terminals determined by the greedy set-cover algorithm (marked green in Table 1). That said, the number of input variables (set by default to $[Ch_1,...,Ch_n]$ in Figure 1), i.e. terminals decreases from n = 25 to 10 which is a significant improvement in terms of search space reduction.

Table 1 Charging candidate optimization results obtained by expert knowledge (i.e, 'manually') and application of modified greedy set-cover algorithm

Terminal / Configuration	s01	s02	s03	s04	s05	s06	s07	s08	s09	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21	s22	s23	s24	s25	Number of charging terminals
Manually found combination	0	x	0	x	0	0	0	x	0	x	x	x	0	x	0	0	x	0	0	0	0	0	0	0	0	8
Greedy combination 1	0	x	0	0	0	0	0	x	0	x	x	0	x	x	x	0	x	0	0	0	0	0	0	0	0	8
Greedy combination 2	0	x	0	x	0	0	0	x	0	x	x	0	0	x	x	0	x	0	0	0	0	0	0	0	0	8
Greedy combination 3	0	x	0	0	0	0	0	x	0	x	x	x	x	x	0	0	x	0	0	0	0	0	0	0	0	8
Greedy combination 4	0	x	0	x	0	0	0	x	0	x	x	x	0	x	0	0	x	0	0	0	0	0	0	0	0	8

The modified set-cover greedy algorithm has proven to be a computationally efficient space reduction approach, as its execution time for the given, relatively large transport system takes only 7 ms on the processor Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz and installed RAM with 8.00 GB.

5. OPTIMIZATION OF OVERALL CHARGING SYSTEM CONFIGURATION

This section presents details of the overall, multi-objective optimization framework built around modeFRONTIER genetic algorithm pilOPT. Figure 5 shows the modeFRONTIER optimization scheme, which includes inputs that represent charging configuration (marked green), and outputs that are used in constraints and objective functions (marked red). The next subsections explain in detail each component of the optimization scheme.

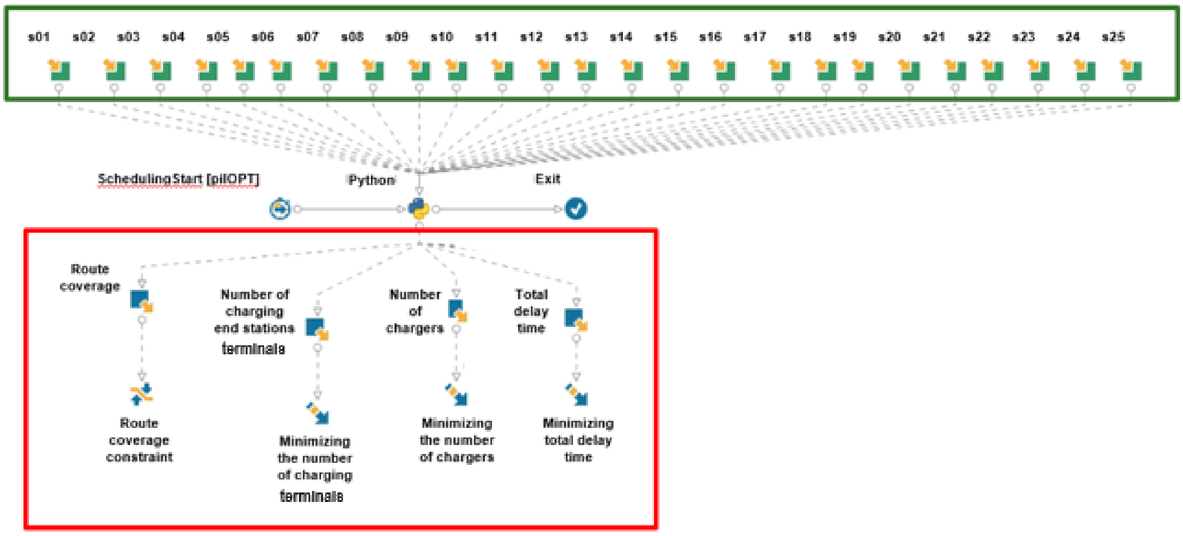

Figure 5 modeFRONTIER scheme of overall charging configuration optimization

5.1. Objective functions

As outlined in Section 2, the considered objective functions to be minimized include the total number of terminals equipped with chargers (N_{cs}), the total number of chargers (N_{ch}) and the total city bus transport system delay time (D_{tot}):

$$\min N_{cs} \tag{3}$$

$$\min N_{ch} \tag{4}$$

$$\min D_{tot} \tag{5}$$

The objectives N_{cs} and N_{ch} are simply determined from the charging configuration candidate generated in each iteration of genetic algorithm, while D_{tot} is calculated by the macro-simulation model (Section 3).

5.2. Optimization problem constraints

As discussed in Section 2, the optimization constraints are formulated as:

$$Ch_i = [0, N_b]/\{1\},\ i = 1,\ldots,n \tag{6}$$

$$RC_r \geq 1,\ r = 1,\ldots,N_r \tag{7}$$

The constraint (6) specifies that the number of chargers at every terminal needs to be in the range from 0 to N_b, except 1. Namely, it is deemed to be cost-ineffective to build the whole terminal charging infrastructure for only one charger. The constraint (7) represents a route coverage constraint meaning that every route needs to include at least one charging terminal. Note that since every charging terminal has at least 2 chargers, the minimum number of chargers available on any route is 2.

5.3. Optimization scenarios

Table 2 overviews the scenarios for which the optimization and related analyses will be carried on in the following subsections. There are four scenarios, each with its own properties related to pilOPT algorithm modes, number of iterations in the case self-initialized mode and the input space size. The pilOPT algorithm has two modes: autonomous and self-initialized mode, where the former stops when the Pareto frontier cannot improve any further, while the latter halts when a predefined number of algorithm iterations is exceeded.

The first scenario is the basic one, where all terminals can be charging terminal candidates, and the optimization algorithm is running in autonomous mode. The second scenario has reduced input space, where the number of charging terminal candidates is reduced from 25 to 10, as discussed in Section 4.3. The charging candidates can have the number of chargers in the range [2, N_b], while the number of chargers for no-charging candidates is set to 0. The third scenario involves the self-initialized mode and the reduced input space, where the pilOPT algorithm is initialized to the number of iterations that was automatically generated in the first scenario. Finally, the fourth scenario is the same as the third one, but the number of iterations is set to the maximum value of 20000.

Table 2 Scenario overview

Scenario	pilOPT mode	Number of iterations (only for self-initialized mode)	Input space
1) Autonomous complete space	Autonomous	-	Whole input space
2) Autonomous reduced space	Autonomous	-	Reduced input space
3) Self-initialized reduced space	Self-initialized	11723	Reduced input space
4) Self-initialized reduced space II	Self-initialized	20000	Reduced input space

5.4. Comparative analysis

Table 3 shows which charging configuration combinations from Table 1 are found in which optimization scenario from Table 2. The labels *Feasible* and *Pareto optimal* designate whether the solution is feasible (in terms of satisfying the constraints) or Pareto optimal (the best at least in one objective), respectively. “Greedy combination 4” results in feasible and Pareto optimal solutions for all optimization scenarios, while other configuration combinations yield only feasible solutions and only in some optimization scenarios.

Table 3 Overview of the charging configuration combinations found in each model, both according to all feasible and Pareto solutions

Scenario \ Charging configuration combination	Greedy combination 1 (Feasible / Pareto optimal)	Greedy combination 2 (Feasible / Pareto optimal)	Greedy combination 3 (Feasible / Pareto optimal)	Greedy combination 4 (Feasible / Pareto optimal)
1) Autonomous complete space	- / -	- / -	- / -	+ / +
2) Autonomous reduced space	- / -	+ / -	+ / -	+ / +
3) Self-initialized reduced space	+ / -	+ / -	+ / -	+ / +
4) Self-initialized reduced space II	+ / -	+ / -	+ / -	+ / +

The reason for the success of “Greedy combination 4” has been found to lie in the effect that charging terminals selected in that configuration have bigger terminal dwell time (the time between arrival and departure) than other charging configuration combinations. According to Table 1, “Greedy combination 1” relies on charging terminals “s13” and “s15”, while “Greedy combination 4” uses terminals “s04” and “s12” for charging. Also, “Greedy combination 3” involves the terminal “s13”, as opposed to “s04” in the case of “Greedy combination 4”. The dwell time graph shown in Figure 6 indicates that the terminals “s04” and “s12” have significantly higher dwell time than the terminals “s13” and “s15” (approx. 18 min vs. 10 min in average), which makes them more suitable charging candidates (higher charging availability). Similarly, “Greedy combination 2” involves the charging terminal “s15”, which has lower dwell time as opposed to “s12” of “Greedy combination 4”.

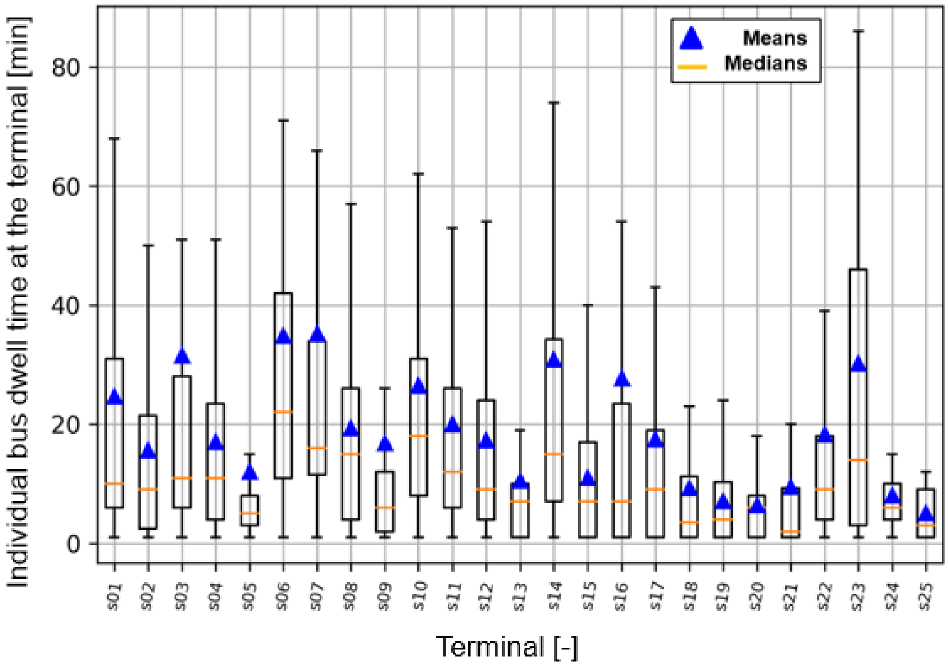

Figure 6 Terminal dwell time statistics

6. RESULTS AND DISCUSSION

In this section, the results for optimization scenarios defined in Table 2 are presented and discussed. First, optimization results are given, which are then supplemented by detailed macro-simulation results.

6.1. Optimization results

The first scenario from Table 2 is the “Autonomous complete space” scenario, which concerns the complete (unreduced) input space and autonomous mode of pilOPT algorithm. The optimization resulted 11723 iterations and it managed to find charging configurations with minimum 8 charging terminals and the total number of chargers in the range from [23, 45], as shown by the 3D Pareto frontier in Figure 7a. This solution is in agreement with the results presented in Section 4 and Table 1, i.e. the full optimization finds the same minimum number of charging terminals as greedy algorithm did. The Pareto frontier in Figure 7a suggests that the total transport system delay time, as the third objective, can be reduced (blue tones) if the number of charging terminals and the number of chargers is increased.

When reducing the input space (Figure 7b), the optimizer again finds configurations with minimum 8 charging terminals, but the number of chargers increase to lie in the range [30, 35], which is suboptimal in comparison to the previous optimization scenario. Since the number of iterations is also significantly lower (4475 vs. 11723), this result can be explained by the solver getting stuck in local optima.

When using the self-initialized mode with the pre-specified number of iterations (equal to that of the first scenario, i.e. 12723), the Pareto frontier shown in Figure 7c is obtained. Again, the configurations with minimum 8 charging terminals are found, but the total number of chargers is reduced to the range [18, 27]. This is a significant improvement in the comparison with the first and second optimization scenarios, which is due to the reduced input space.

When using the maximum number of iterations, which is 20000, the optimization results in the Pareto frontier shown in Figure 7d. Here, the optimal configurations with the minimum number of charging terminals equal to 8 are extended to the number of chargers in the range [16, 26], i.e. the number of chargers can be reduced to 16 and 17 when compared to the third optimization scenario. However, the maximum time delay for those two configurations is very large (more than 5 hours vs. half an hour for the case of 18 chargers). Thus, those configurations are rejected, and it may be concluded that the previous scenario “Self-initialized reduced space” could not be further improved. Its characteristic charging configurations marked by black circles in Figure 7c will be analysed in detail in Subsection 6.3.

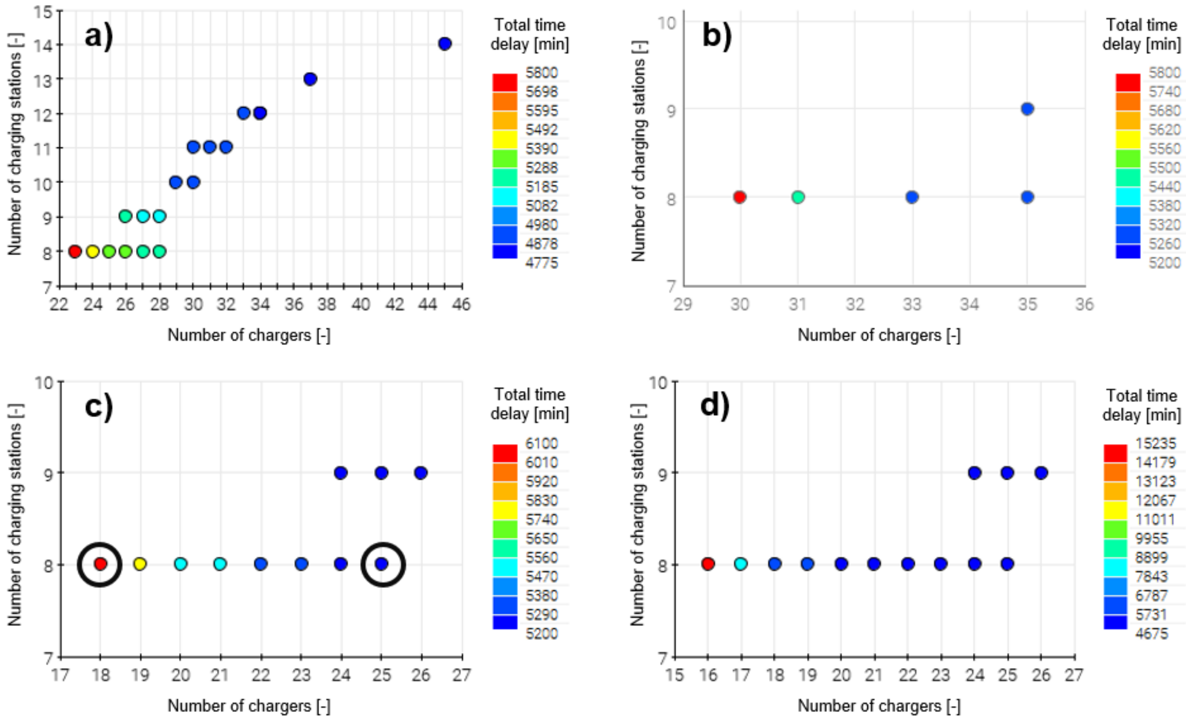

Figure 7 Pareto frontiers obtained for different optimization scenarios: a) Autonomous complete space, b) Autonomous reduced space, c) Self-initialized reduced space, d) Self-initialized reduced space II

6.2. Optimization procedure

Based on the results from the previous subsection, this subsection formalises the optimization steps, as shown in Figure 8 and elaborated as follows: **(i)** Autonomous complete space scenario is run first in order to give the number of iterations for step (iii), **(ii)** Set of charging terminal candidates is generated by using the modified greedy set-cover algorithm, as explained Section 4, **(iii)** Self-initialized reduced space scenario is run with the number of iterations taken from step (i) and charging terminal candidates from step (ii), **(iv)** Pareto frontier obtained in step (iii) is used to obtain configurations with the minimum number of chargers and charging terminals

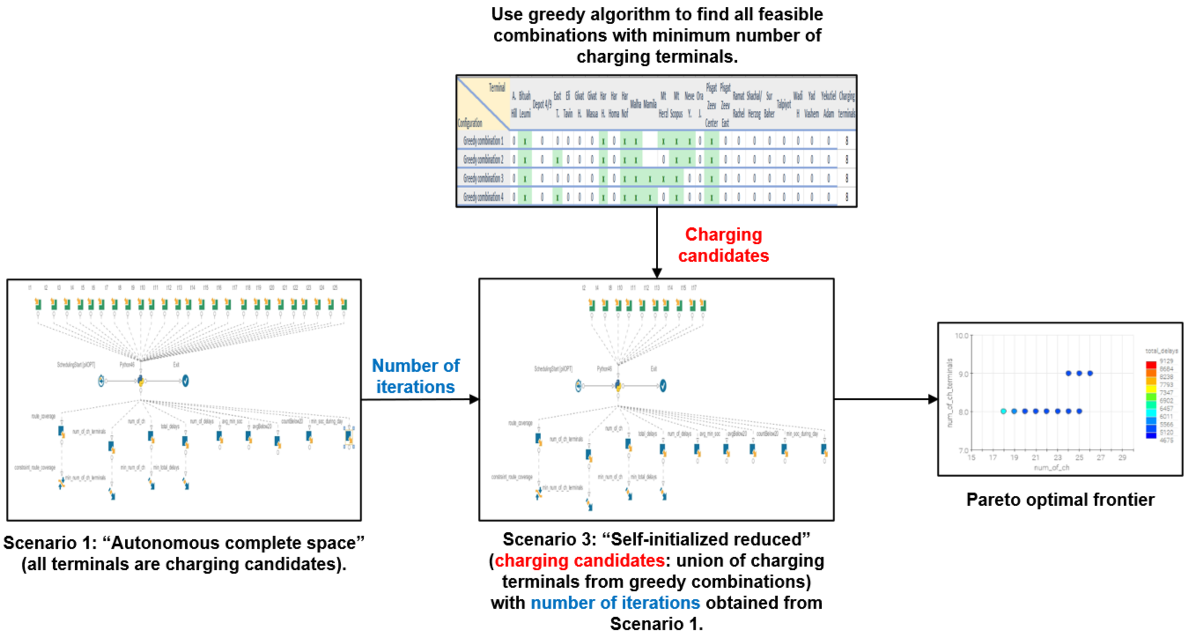

Figure 8 Optimal charging configuration model setup

6.3. Simulation outcomes for Pareto optimal solutions

The optimal configurations obtained in Subsection 6.1 (see Figure 7c) based on the procedure summarized in Subsection 6.2 (i.e., Figure 8) are analysed in this subsection based on the macro-simulation results. The results are compared with those corresponding to the charging configurations found manually, i.e. through expert knowledge (see Table 1). The purpose of the detailed analysis is to assess charging configurations and get a detailed overview of macro-simulation results (i.e. final-SoC distribution, energy-charged, number of utilized buses, dwell time etc.).

6.4. Comparative results

The macro-simulation results presented in Figure 9 contain 6 plots. The first plot shows the SoC at the end of the day (i.e., the final SoC) for every route. The second graph is a bar chart of the final SoC categories related to unacceptable ($SoC_f < 0$), risky ($0 \leq SoC_f < 20\%$) and safe ($SoC_f \geq 20\%$) final SoC. The third plot shows the total energy charged to all buses at each route. The fourth plot is a boxplot representing the available charging time statistics. The fifth graph shows the number of utilized/unutilized buses on every route. The last (sixth) plot gives the boxplot statistics of the individual bus delay time for every route.

Pareto frontier gathered from the "Self-initialized reduced space" scenario, shown in Figure 7c, resulted in configurations that have [18, 25] chargers distributed on 8 charging terminals, which is the minimum number of charging terminals when satisfying the route coverage constraint. For the sake of simplicity, only configurations with upper and lower bounds of the number of chargers (designated by circles in Fig. 7c) will be simulated and discussed. Figure 9 presents the macro-simulation results for lower-band configuration with 18 charging terminals.

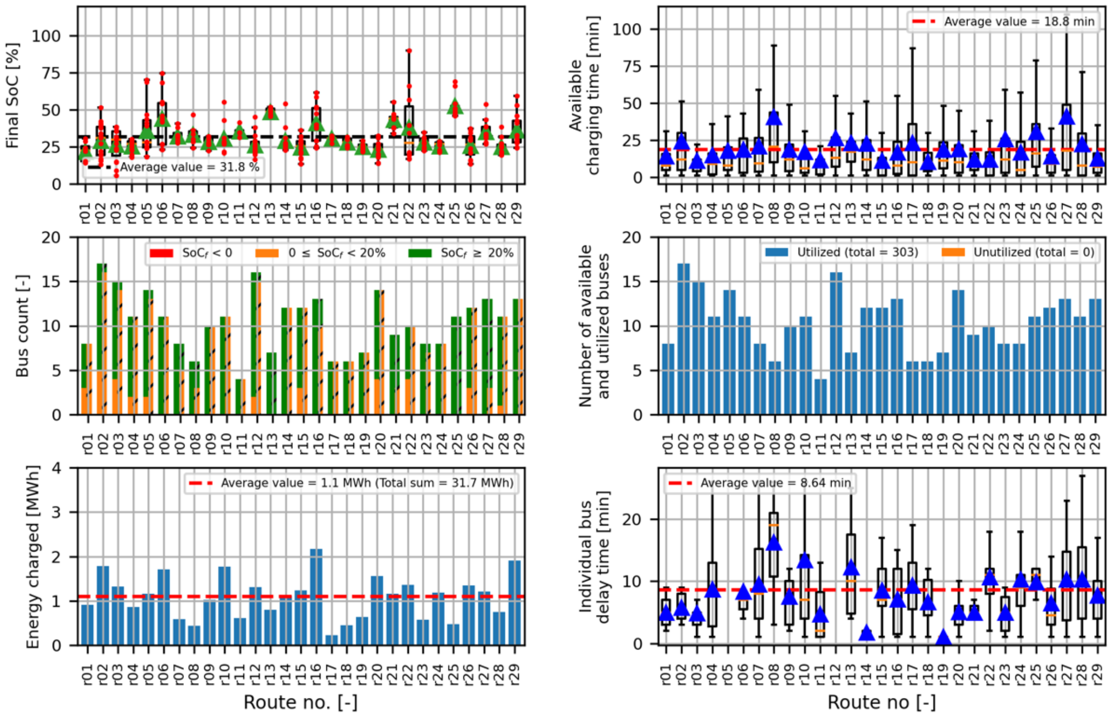

Figure 9 Macro-simulation results for optimal charging configuration related to 8 charging terminals and 18 chargers (see left-hand side circle in Fig. 7c)

The optimal configuration with 18 chargers is sustaining, i.e. all buses have $SoC_f > 0$. The total/cumulative delay time per bus is reasonable, with minimum values of 1 min, and a peak lower than 30 minutes. Note that some routes (i.e. “r05” and “r12”) have no delayed missions. Thus, this configuration may be deemed as overly satisfactory. However, some routes are characterized by low final SoC; i.e. route “r03” has a bus with a final SoC value of around 5%, which can be regarded as risky and can be improved by adding more chargers to a terminal of that route.

Figure 10 shows macro-simulation results related to optimal charging configuration with 25 chargers (see right-hand side circle in Figure 7c). Since the number of chargers is increased by 7 compared to the previous configuration, the final SoC values are higher, and accordingly the total delay time is somewhat reduced (Figure 10).

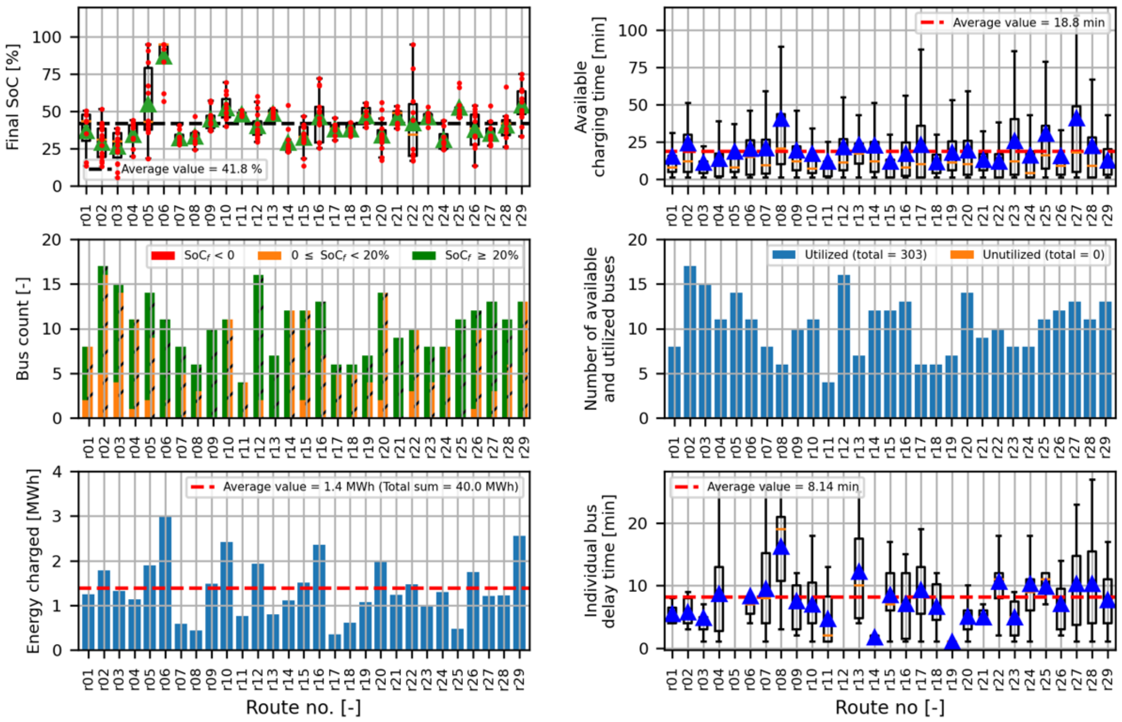

Figure 10 Macro-simulation results for optimal charging configuration related to 8 charging terminals and 25 chargers (see right-hand side circle in Fig. 7c)

The configurations found through expert knowledge (i.e., "manually") are listed in Table 4 based on Table 1 and variation of total number of chargers. Table 4 also shows the above-considered, optimal configurations. All the configurations have 8 charging terminals and a number of chargers in the range [18, 46], where the charging configuration with the minimum number of chargers is the one obtained by using the pilOPT optimization and analysed with Figure 9.

Table 4 Manually-found and optimization-obtained charging configurations

Terminal / Configuration	s01	s02	s03	s04	s05	s06	s07	s08	s09	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21	s22	s23	s24	s25	Total number of chargers	Number of charging terminals
Manually found configuration 1	0	11	0	2	0	0	0	3	0	10	6	4	0	8	0	0	2	0	0	0	0	0	0	0	0	46	8
Manually found configuration 3	0	12	0	4	0	0	0	3	0	8	8	3	0	5	0	0	3	0	0	0	0	0	0	0	0	46	8
Manually found configuration 4	0	10	0	2	0	0	0	2	0	8	6	3	0	5	0	0	2	0	0	0	0	0	0	0	0	38	8
Manually found configuration 5	0	7	0	2	0	0	0	2	0	7	6	3	0	4	0	0	2	0	0	0	0	0	0	0	0	33	8
Manually found configuration 6	0	6	0	2	0	0	0	2	0	6	5	2	0	4	0	0	2	0	0	0	0	0	0	0	0	29	8
Manually found configuration 7	0	5	0	2	0	0	0	2	0	4	3	2		4	0	0	2	0	0	0	0	0	0	0	0	24	8
Reduced self-ini. pilOPT 18	0	4	0	2	0	0	0	2	0	2	2	2	0	2	0	0	2	0	0	0	0	0	0	0	0	18	8
Reduced self-ini. pilOPT 25	0	6	0	3	0	0	0	2	0	3	4	2	0	2	0	0	3	0	0	0	0	0	0	0	0	25	8

Table 5 shows comparative performance metrics based on the macro-simulation output data. The pilOPT charging configuration with 18 chargers is optimal in terms of investment cost, but it has a considerably lower final SoC value than other configurations having more chargers. Accordingly, there is also a significant increase in the number of arrivals with $SoC_f < 20\%$ than in other configurations. To this extent, the configuration pilOPT 25 should be preferred over pilOPT 18, and it is distinctively better than the manually found configuration with comparable (or even somewhat higher) number of chargers in terms of final SoC, delay, and energy charged statistics.

Table 5 Overview of macro-simulation-based performance metrics for manually selected and optimal charging configurations

Configuration	Avg. final SoC [%]	Total energy charged [MWh]	Avg. dwell time [min]	Avg. delay time [min]	Number of charging terminals	Total number of chargers	Min. SoC during day [%]	Number of delayed departures [-]	Total delay time [min]	Count of arrival SoC < 20% [-]	Avg. SoC for events where SoC < 20%
Manually found configuration 1	59.1	54.5	18.8	8.67	8	46	5.45	607	5263	9090	16.55
Manually found configuration 2	61.3	56.3	19.1	8.14	8	46	5.45	585	4760	8212	16.46
Manually found configuration 3	55.6	51.6	18.8	8.67	8	38	5.45	607	5263	9411	16.5
Manually found configuration 4	52.5	49	18.9	8.67	8	33	5.45	607	5263	9422	16.5
Manually found configuration 5	49.3	46.3	18.9	8.67	8	29	5.45	607	5263	9460	16.51
Manually found configuration 6	42.7	40.8	18.5	8.66	8	24	5.45	608	5266	9454	16.52
Self-ini. reduced with 18 chargers	31.8	31.7	18.8	8.64	8	18	5.45	612	5286	11375	16.68
Self-ini. reduced with 25 chargers	41.8	40.0	18.8	8.14	8	25	5.45	585	4760	8926	16.5

7. CONCLUSION

A search space reduction-supported multi-objective approach of optimizing the city bus charging configuration system has been proposed and implemented by using the pilOPT algorithm of modeFRONTIER environment. The approach is summarized in Figure 8, and includes **(i)** obtaining the number of iterations from the “Autonomous complete space” scenario, **(ii)** utilizing the modified greedy set-cover algorithm to reduce the input space, i.e. obtain the optimal charging terminals candidates, and **(iii)** creating the self-initialized optimization model with the number of iterations set as obtained in step (i) and with a reduced number of charging terminals according to step (ii), **(iv)** analysing Pareto frontier solutions and choosing the one with a minimum number of chargers and charging terminals while satisfying other practical/operational metrics such as those related to battery state of charge (SoC) final value and cumulative bus departure delay.

The selected Pareto optimal charging configurations have been compared with the ones found based on expert knowledge. It has been proven that the proposed optimization approach results in a lower number of chargers keeping the total delay time low and ensuring bus transport system maintainability in the view of battery state of charge.

The future work can include optimizing the charging power rather than using a constant/maximum one, in order to minimize the exploitation cost in systems with varying electricity price through the day or local renewable, intermittent energy sources. Another relevant topic of future work relates to combined optimization of bus schedules and charging infrastructure.

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. The first author also acknowledges the support from a 2022 small-scale grant program of the University of Zagreb. The authors are grateful to ROM Transportation Engineering Ltd., Tel Aviv, Israel for technical and data support.

REFERENCES

1. He, Y., Kockelman, K. M., & Perrine, K. A. (2019). Optimal locations of U.S. fast charging terminals for long-distance trip completion by battery electric vehicles. *Journal of Cleaner Production*, *214*, 452–461.
2. Pan, L., Yao, E., Yang, Y., & Zhang, R. (2020). A location model for electric vehicle (EV) public charging terminals based on drivers' existing activities. *Sustainable Cities and Society*, *59*.
3. Zeb, M. Z., Imran, K., Khattak, A., Janjua, A. K., Pal, A., Nadeem, M., … Khan, S. (2020). Optimal Placement of Electric Vehicle Charging Terminals in the Active Distribution Network. *IEEE Access*, *8*, 68124–68134.
4. Mainul Islam, M., Shareef, H., & Mohamed, A. (2018). Optimal location and sizing of fast charging terminals for electric vehicles by incorporating traffic and power networks. *IET Intelligent Transport Systems*, *12*(8), 947–957.
5. Ma, J., & Zhang, L. (2018). A deploying method for predicting the size and optimizing the location of an electric vehicle charging terminals. *Information (Switzerland)*, *9*(7).
6. Akbari, M., Brenna, M., & Longo, M. (2018). Optimal locating of electric vehicle charging terminals by application of Genetic Algorithm. *Sustainability (Switzerland)*, *10*(4).
7. Topić, J., Soldo, J., Maletić, F., Škugor, B. & Deur, J. (2020) Virtual Simulation of Electric Bus Fleets for City Bus Transport Electrification Planning. Energies, 13 (13), 3410, 23 doi:10.3390/en13133410.
8. Dabčević, Z.; Škugor, B.; Topić, J.; Deur, J. (2022) Synthesis of Driving Cycles Based on Low-Sampling-Rate Vehicle-Tracking Data and Markov Chain Methodology. Energies 2022, 15, 4108. https://doi.org/10.3390/ en15114108
9. Sundström, O., & Binding, C. (2012). Flexible charging optimization for electric vehicles considering distribution grid constraints. IEEE Transactions on Smart Grid, 3(1), 26–37.
10. Sundström, O., & Binding, C. (2010). Optimization Methods to Plan the Charging of Electric Vehicle Fleets. Proceedings of the International Conference on Control Communication and Power Engineering, 1(2), 28–29.
11. Vazirani, V. V. (2001). *Approximation algorithms.* Springer. ISBN: 978-3-540-65367-7