Horizon 2020

OLGA

D2.1: Software solution for e-bus transport electrification and related transport system optimization

Grant Agreement number: 101036871

Project acronym: OLGA

Project title: HOListic & Green Airports

Funding scheme: Innovation Action (IA)

Start date of the project: 1st October 2021

Duration: 60 months

Project coordinator: Virginie Pasquier, Project Manager Environment / Energy, ADP

Tel: +33 7 88 35 16 07

E-mail: virginie.pasquier@adp.fr

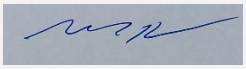
Project website address: www.olga-project.eu

OLGA project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 101036871

DOCUMENT INFORMATION

Document Name	**Software solution for e-bus transport electrification and related transport system optimization**
Version	V1
Version date	08/01/2024
Author	Joško Deur, Zvonimir Dabčević, Branimir Škugor, Daria Matković
Security	Public (PU)

APPROVALS

	Name	Company	Date	Visa
Coordinator	Virginie Pasquier	ADP	19/12/2023	
WP Leader	Marin Tica	MZLZ	15/11/2023	
Task Leader	Marin Bračić	FTTS	15/11/2023	
Quality Manager	Anamul (Andy) Hogue	LUP	18/12/2023	

DOCUMENT HISTORY

Version	Date	Modification	Authors
V01	22/10/2023	Initial version	J. Deur et al.
V02	18/12/2023	Quality reviewed	A. Hoque
V1	08/01/2024	Final version	J. Deur, A. Hoque

Table of Contents

List of tables

List of figures

List of Abbreviations

A/C	Air Conditioning	**GPS**	Global Positioning System
BEV	Battery Electric Vehicle	**GUI**	Graphical User Interface
BIC	Bayesian Information Criterion	**HVAC**	Heat, Ventilation, and Air Conditioning
CO_2	Carbon dioxide	**H2**	Hydrogen
CD/CS	Charge Depleting/Charge Sustaining	**HEV**	Hybrid Electric Vehicle
CE	Central Europe	**ICT**	Information and Communication Technology
COM	Charging Optimization Module	**I/O**	Input-Output
COP	Coefficient of Performance	**LASSO**	Least Absolute Shrinkage and Selection Operator
CONV	Conventional	**LFP**	Lithium Iron Phosphate
DMM	Data Management Module	**MILP**	Mixed Integer Linear Programming
DPPM	Data Post-Processing Module	**MPC**	Model Predictive Control
DHD	Deadhead Distance	**MLP**	Multilayer Perceptron
DOE	Design of Experiments	**PHVAC**	Powers of Auxiliary Devices and Heat, Ventilation, and Air Conditioning
DIM	Diminishing Horizon	**PHEV**	Plug-in Hybrid Electric Vehicle
DC	Distribution Centre	**PID**	Proportional-Integral-Derivative
DP	Dynamic Programming	**REC**	Receding Horizon
EBSM	Electric Bus Simulation Module	**RES**	Renewable Energy Sources
e-bus	Electric Bus	**RMS**	Root Mean Square
e-hub	Electric Hub	**SOLEZ**	Smart Solutions Supporting Low Emission Zones
ECM	Equivalent Circuit Model	**SoC**	State of Charge
ECMS	Equivalent Consumption Minimization Strategy	**SoE**	State of Energy
EREV	Extended Range Electric Vehicles	**TEAM**	Techno-Economic Analysis Module
FME*	Faculty of Mechanical Engineering and Naval Architecture	**TCO**	Total Cost of Ownership
GA	Genetic Algorithm	**WP**	Work Package

1 Executive summary

1.1 Introduction

City-type buses are used to connect airports with cities and also to provide intra-airport landside transport (terminal-terminal, parking-terminal). In order to reduce emissions of pollutants and greenhouse gases, reduce noise emissions, and increase customer satisfaction, the existing conventional airport-city and intra-airport buses should be replaced by electrical ones, where an ultimate solution is fully electric bus (e-bus). Moreover, electric power engaged for intermittent charging of e-buses can be utilized to establish an airport e-hub to provide charging services for customer (passenger) vehicles at the airport parking lots.

To mitigate the disadvantages of e-buses in terms of limited range, high cost, need for charging infrastructure, and relatively long charging time, and speed up the electrification process, there is a need for software solutions (applications) for electrification planning. Such a software solution has been developed by the FME team through the Interreg CE project SOLEZ and demonstrated on the pilots of cities of Dubrovnik and Žilina [1]. It is based on virtual simulation of an e-bus fleet over actual (recorded) driving cycles and setting optimal e-bus and charging infrastructure configurations to minimize the total cost of ownership calculated by the techno-economic analysis module.

The SOLEZ software solution has been significantly modified and extended through this deliverable in terms of computationally more efficient e-bus fleet simulations, automated and more accurate charging configuration optimization, predictive and optimal charging management, and optimization of e-bus scheduling for lower investment cost, having in mind application to the city-airport and intra-airport bus transport systems and the airport e-hub. The extended software solution will be applied in the remaining course of the project for optimal planning of the airport bus system electrification at Paris-Charles de Gaulle airport and Zagreb airport and the e-hub system at the Zagreb airport (to be delivered in D2.2 by M54).

1.2 Brief description of the work performed and results achieved

The work performed and the results achieved are outlined below in relation to the four components of the work.

1) A **trip-based data-driven e-bus model** has been built up to substantially speed up the e-bus fleet virtual simulation and, thus, the whole software solution, while requiring only trip-based driving cycle features that are usually available to transport operators through bus tracking system or planning tools.

2) A multi-objective genetic algorithm (GA)-based **charging configuration optimization solution** has been developed to automate optimization of charging stations and chargers' deployment and minimize the investment cost and the fleet service delay. The optimization results in a Pareto frontier of optimal solutions reflecting the minimization of the conflicting objectives including the number of charging stations, the total number of chargers, and the service delay, from which the operator can select a most favourable solution for the given transport system.
3) Model predictive control strategy for **optimal online charging management** of a fleet of vehicles aggregated through a transport operator or an e-hub provider has been designed. The strategy minimizes the cost of charging electricity based on prediction of transport demand, electricity cost, and production from local renewable energy sources, thus providing a more accurate electrification planning outcomes and more competitive and cleaner charging services.
4) **E-bus scheduling optimization** has been developed based on a mixed integer linear programming (MILP) formulation and two complementary solvers (MILP and GA). This solution allows for scheduling optimization targeted at e-buses to minimize the number of busses, where the buses can change the lines dynamically for the best utilization of their remaining range and charging opportunities. The optimization results in a Pareto frontier in two conflicting objectives including the total number of buses and a deadhead distance (the distance travelled while changing the lines), from which the operator can pick a most beneficial solution for the given transport system.

2 Structure of software solution

2.1 Target application

The airport-to-city (and intra-airport) e-bus transport system should be electrified for clear environmental and social benefits (Section 1). Due to the availability of high-power electricity installation including an excellent potential for deploying renewable energy sources (RES; photovoltaic panels, in particular), the airport would be a natural candidate for installing an e-bus fast-charging station. Since the e-bus dwelling/charging time at the airport would be relatively small, the engaged power remains for the most of time at disposal for supplying other airport landside vehicles and particularly customer electric vehicles (passenger, taxi, rent-a-car, and sharing vehicles) being parked at the airport parking lots (Fig. 1). By designing such an e-hub, the airport can provide better utilization of the local grid and RES, and at the same time offer charging services for better customer satisfaction, environment protection, e-mobility proliferation, and increased income/profit.

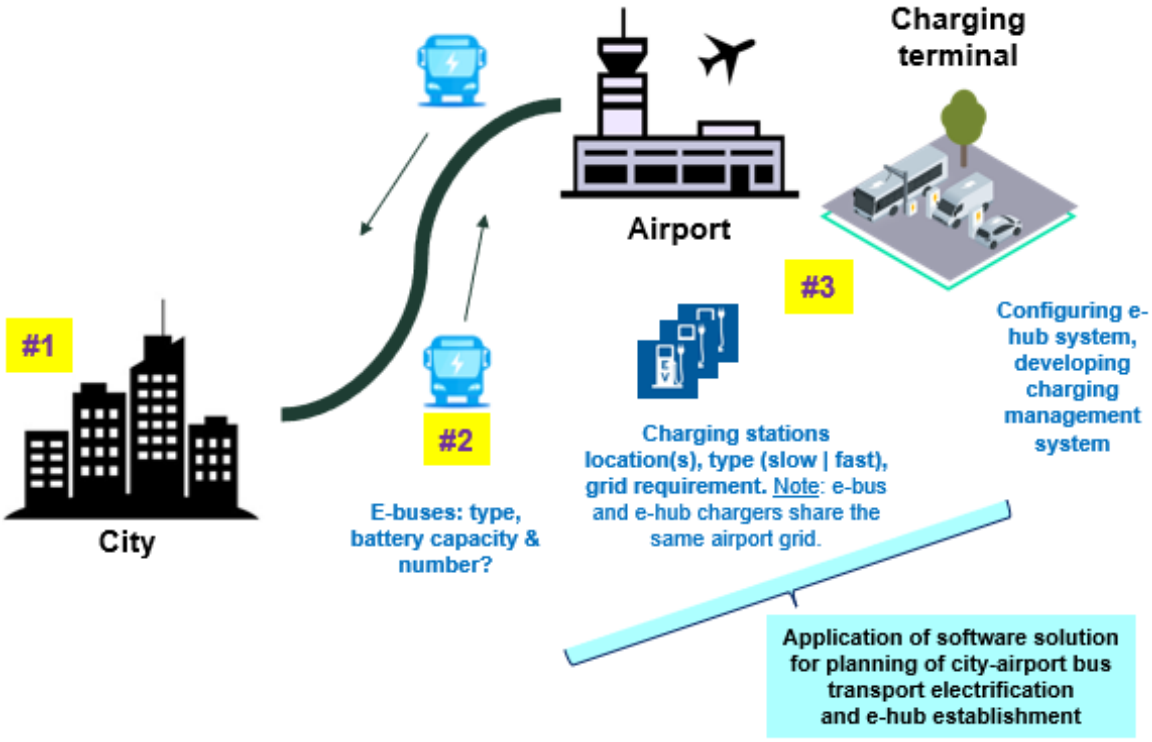


Fig. 1 Illustration of airport-city bus transport electrification and e-hub establishment planning framework

The overall e-bus transport and e-hub system becomes quite complex and requires proper software solution for optimized planning to minimize the investment (e.g., e-buses, charging stations) and operational costs (e.g., electric energy). Typical questions which such an electrification planning study should generally address include:

(i) what type of e-bus (HEV, PHEV, BEV) and with what size of battery are recommended (the emphasis is on BEV, i.e. fully electric vehicle);

(ii) what type of chargers and with what rated power are recommended (e.g., slow/night charging only, fast charging from grid, fast charging from a stationary battery);

(iii) where should the chargers be located (city or airport) and how the airport building power grid (including RES, stationary battery and similar) should be configured for minimum cost and CO_2 emissions;

(iv) how should the e-hub system be configured in support of charging other e-vehicles;

(v) how should the e-bus fleet and e-hub scheduling and charging management system be implemented; and

(vi) what would be proper ICT solutions in support of overall vehicle-grid integration including interoperability?

As far as the bus transport electrification planning is concerned, the software solution/tool developed by the FME team through the Interreg CE project SOLEZ [1] may be used to provide answers to the above questions, based on a virtual simulation of the targeted electrified transport system using recorded driving cycles and faithful physical e-bus models. However, to address the e-hub design needs and provide automated charging configuration and bus scheduling optimization, the SOLEZ tool should be significantly extended. The tool is described in Subsection 2.2, its limitations are discussed in Subsections 2.3, and its modifications and extension made through this project are outlined in Subsection 2.4.

2.2 Description of existing software solution developed through SOLEZ project

Fig. 2 illustrates the structure of the city bus transport planning software tool developed through SOLEZ project [1]. The tool is driven by recorded driving cycle data, and as the main output it delivers the Total Cost of Ownership (TCO) over the projected fleet operational period (e.g., 12 years). The Data Post-Processing Module (DPPM) transforms the recorded driving data into individual driving cycles, and it also calculates various statistical features characterizing the conventional city bus transport behaviours. The Electric Bus Simulation Module (EBSM) provides computer simulations of different types of city buses (CONV, HEV, PHEV, BEV) over the driving cycles extracted by the DPPM. The module outputs include the individual bus energy consumption (fuel and/or electricity) and various features of powertrain response (e.g., engine/e-motor operating points, gear ratio trajectories, etc.). The Charging Optimization Module (COM) utilizes the outputs of DPPM and EBSM to virtually simulate the overall city bus fleet over the recorded driving cycles and optimise the PHEV- and BEV-type bus charging configuration and management. This module provides the number, location, and type of chargers, the bus battery capacity, and the number of reserve buses in the BEV case, which are required to fulfil the driving routes with sufficient battery charge. The COM also outputs the total fuel and/or electricity consumption over the considered period of operation. The Techno-Economic Analysis Module (TEAM) uses the output data from the COM module, as well as the data on bus transport investment and exploitation/maintenance cost, in order to calculate the TCO.

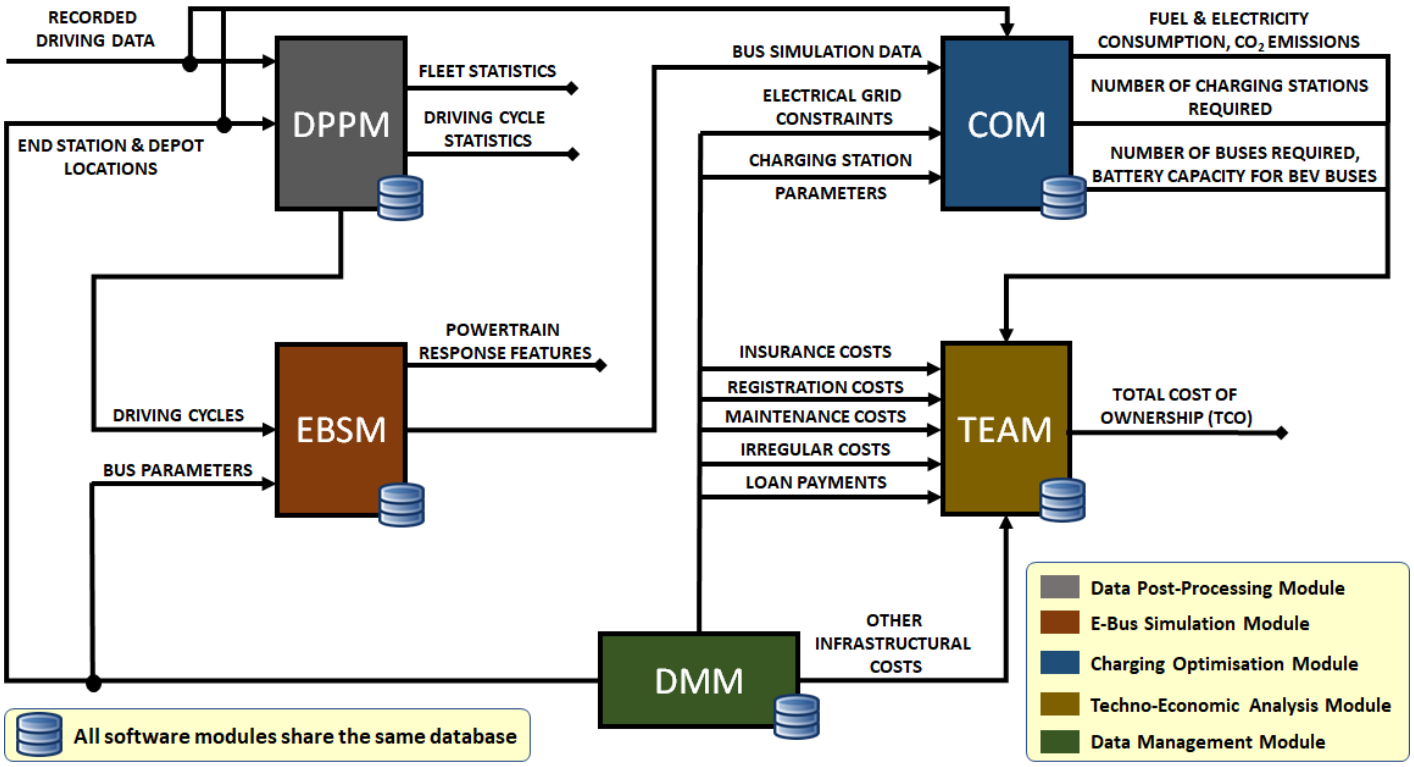


Fig. 2 Organizational structure of e-bus transport electrification tool developed through SOLEZ project

The simulation tool is written in Python object-oriented programming language, with computationally demanded routines coded in C language. It is designed in a user-friendly way (based on a graphical user interface (GUI) including windows, tabs, I/O data interfaces, etc.) and having in mind transferability to other cities in a way that it uses a common/shared database. The database serves as a main storage for recorded driving cycle data and plays the role of an intermediary between the main tool modules. In addition, the simulation tool includes the Data Management Module (DMM), which provides greater flexibility and adaptability to different cities' transport system configurations. DMM enables the user to define all static data (system parameters) required by the simulation tool, e.g. those related to vehicle model parameters, end-station and depot locations, charging station parameters and techno-economic data.

2.3 Limitations of existing software solution

The following restrictions of the SOLEZ solution have been identified from the perspective of more effective and accurate application to a wide range of electrification studies including the one considered in this project and outlined in Subsection 2.1:

1) Physical models of e-buses are used in the EBSM, which is connected with the following disadvantages: (i) physical models require high-resolution driving cycles (with a typical sampling frequency of 1 Hz) that are not regularly available to e-bus operators and (ii) use of such an approach ends up in a relatively long simulations which prevents the tool from application to large-scale transport systems (hundreds or thousands of e-buses) and/or studies based on automated optimization (where a great number of simulation is conducted).
2) Charging configuration determined by locations and number of chargers is determined "manually" by the COM, i.e. by nominating a certain number of configurations, repeating the virtual simulation, and finding the one which gives the minimum TCO. Based on the knowledge gained from the initial COM execution, a new set of configurations can be nominated, and the process is repeated. The main disadvantage of this process includes the need for significant involvement of expert knowledge and hours. Also, it generally gives suboptimal results due to a limited configuration search.
3) The transport system virtual simulation is run by the COM for a fixed (predetermined) e-bus schedules, typically corresponding to the conventional e-bus fleet schedules. The solution obtained in that way is generally suboptimal, because the range and charging restrictions of e-buses are missed to be overcome to some extent by re-scheduling the buses (e.g., a charged bus can move to another line to take the service of buses being charged on that line, or an empty bus can be reallocated to another line where there is a free charger).
4) The charging management algorithm implemented within the COM is based on a simple heuristic, rule-based logic, which may result in distinctively sub-optimal solution for mid/large-scale transport systems and can hardly be used when planning and managing the e-hub system.

2.4 New features of upgraded software solution

The limitations of the SOLEZ solution have been overcome by developing a set of new software modules. They are outlined below in the same order as given for the corresponding limitations listed in Subsection 2.3, and are elaborated in details in Sections 3-6:

1) **Trip-based data-driven e-bus model** (Section 3). The physical model is replaced by its data-driven approximation, which calculates the e-bus energy consumption directly from the trip-lumped features such as distance travelled, average velocity, average road slope etc. Provided that the approximation errors are small, the advantage of using the trip-based data-driven model is twofold: (i) the model executes swiftly, thus being suitable for large-scale and optimization-based electrification studies and (ii) the model training requires usually available trip-based features of the driving cycle. The e-bus powertrain model development includes the

following characteristic phases: a) preparation of training and validation/testing datasets, b) selection of relevant driving cycle features, c) model parameterization and validation, and d) model assessment. The HVAC system model is developed separately, and the energy consumptions predicted by the two models are summed up.

2) **Charging configuration optimization** (Section 4). A multi-objective genetic algorithm (GA) is used to optimize the charging configuration given by the locations of charging stations and the number of chargers per each charging station. The optimization is conducted for the e-bus transport system represented by a computationally efficient trip-based model. The optimization results in a Pareto frontier in the following three objectives: the number of charging stations, the total number of chargers, and the total (cumulative) e-bus fleet delay caused by charging restrictions. A special attention is given to designing an algorithm that finds a set of configurations with the minimum number of charging stations, whose results are used to reduce the search space of the GA. The designer picks a point from the Pareto frontier which provides a good trade-off of investment cost (number of charging stations and chargers) and operational delay cost, and at the same time provides a reserve (i.e., robustness) against the transport system modelling errors.
3) **E-bus scheduling optimization** (Section 6). For the given transport system (defined by e-bus type, lines, and timetables) and (pre-optimized) charging configuration, e-bus scheduling is optimized in terms of which bus takes which service trip, including the possibility to move buses between lines to fulfil the timetables and/or take the opportunity to recharge. The optimization problem is defined as a MILP problem, considering various charging constraints including the charge sustaining condition (all the buses should have the same initial and final state of charge conditions and allowing for a wide system specification flexibility (e.g. in terms of specifying individual charger maximum power, the number of chargers per stations, the bus battery capacity etc.). The optimization problem is solved by a MILP algorithm and a GA, where the former provides optimal solution but is impractical for large-scale systems due to computational inefficiency, while the latter is computationally efficient but only nearly optimal. The optimization results in a Pareto frontier in two objectives being minimized: the total number of buses and the deadhead distance. The Pareto frontiers have been obtained for both e-bus and conventional bus fleets, and they are comparatively analysed.
4) **Model predictive charging management** (Section 5). A model predictive control (MPC) strategy, run over a receding horizon or a single-day shrinking horizon, is designed to handle online charging management in an optimal and predictive manner. The optimality is formulated in minimizing the total charging electricity cost, while the predictive feature relates to anticipating varying electricity prices, RES production, and transport demand. In order to make

the strategy feasible for large-scale systems, such as the e-hub one, it is formulated to have a hierarchical structure, where (i) the upper layer optimizes the aggregate charging power based on the transport system modelled by an aggregate battery, and (ii) the lower layer distributes the aggregate charging power to individual vehicles based on their priorities in terms of actual battery state of charge and charging station departure time. The upper-layer MPC strategy relies on a dedicated dynamic programming (DP) solver to handle generally nonlinear and discontinuous transport and energy system model and constraints. The efficiency of online MPC strategy is demonstrated by comparing the charging management results with those obtained by using globally optimal (full-horizon) DP optimization on the aggregate level.

The above software modules have been demonstrated on the case studies available through previous projects of the FME team, because the anticipated OLGA pilot studies' data have not been fully acquired during the course of software solution development. The OLGA pilot studies are subject of forthcoming work, and the results will be published in D2.2 by M54.

3 E-bus model

3.1 Introduction

E-bus model is a key element of the overall software solution (Section 2), as it provides virtual simulation of e-bus fleet for realistic driving cycle. The e-bus fleet energy consumption gained through the virtual simulation represents a crucial input to charging configuration and management algorithms, as well as techno-economic analyses (Fig. 1).

The energy consumption predictions are usually based on elementary, physics-based, and data-driven models. The elementary model includes a direct relation between the e-bus energy consumption and a travel feature (typically the distance travelled). It tends to oversimplify real-world scenarios, thus compromising the prediction accuracy. The physical models include first-principle equation describing the vehicle powertrain and longitudinal dynamics behaviours. However, they require a number of physical parameters and maps, as well as high-sampling-rate driving cycle data, which can be challenging and costly to obtain in regular fleet operation applications. On the other hand, the data-driven methods employ machine learning techniques, including neural networks and random forests, and generally provide favourable accuracy. However, to accurately capture real-world patterns, they require large input/output datasets that are not readily available, particularly for e-bus fleets.

To mitigate the limitations of the existing methods, this section presents a novel macroscopic data-driven regression model for e-bus energy consumption prediction. A rich dataset for model parameterization and validation is generated by using an experimentally validated physical e-bus model. The data-driven model relies on trip-centric input data such as distance travelled, mean

velocity, and road gradient, which are usually available from city bus transport planning or GPS tracking datasets. Given its rapid execution speed, the model is well-suited for comprehensive large-scale city-bus electrification planning studies. The model is presented for a fully-electric city bus of 12 m size. It can readily be extended to other bus types such as conventional, hybrid, and plug-in hybrid buses and other bus sizes such as 18 m bus. A special attention is devoted to modelling of heating, ventilation and air-conditioning (HVAC) system, since it represents the second largest energy consumer (after the powertrain).

The remaining part of this section is organized as follows. Subsection 3.2 presents the experimentally validated physical e-bus model, which is used in Subsection 3.3 as a basis for sensitivity analysis of e-bus energy consumption and in Subsection 3.4 for data collection used in data driven e-bus modelling. Feature selection in support of data-driven modelling is described in Subsection 3.5. Subsection 3.6 presents, validates, and assesses the final data-driven e-bus model consisting of separate powertrain and HVAC system submodels. Subsection 3.7. gives concluding remarks.

Note: The work presented in this section has been disseminated through the following conference papers (the first one on physical model, and the second one on data-driven model), which also include a methodology state-of-the-art review and elaborates on the contributions of the approaches proposed:

J. Deur, I. Cvok, I. Ratković, J. Topić, J. Soldo, F. Maletić, "Backward-looking Modelling of a Fully Electric City Bus with Emphasis on Cabin Heating and Cooling Subsystem", 18th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Dubrovnik, Croatia, 2023.

Z. Dabčević, B. Škugor, J. Deur, "A Trip-Based Data-Driven Model for Predicting Battery Energy Consumption of Electric City Buses", 18th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Dubrovnik, Croatia, 2023.

3.2 Physical e-bus model

3.2.1 Recorded driving cycle and energy consumption data

The driving cycle and energy consumption data have been recorded on a single, 12 m e-bus operating on Route 15 in the city of Jerusalem [7]. The route is bidirectional and stretches between the end stations Binyenei HaUma and Talpiot. The data were collected on August 13, 2020 (peak summer season) in the period from 7 am to 9 pm. The recording was continuous with the sampling time of 1 second, and it concerned the following data: timestamp, geographical coordinates (longitude, latitude and altitude), velocity, distance travelled, cumulative battery energy consumption and battery SoC.

The considered dataset contains 14 trips in total (7 for each travel direction), where each individual trip is defined as driving between the two end stations in either direction. The velocity profile along the day is shown in Fig. 3a. The total distance travelled is approximately 122.5 km for the net operating time of 11.5 h. The corresponding reconstructed ridership profile is shown in Fig. 3b. Finally, the actual ambient temperature (T_a) and solar irradiance ($\dot{Q}_{sol}$) data profiles are shown in Fig. 3c.

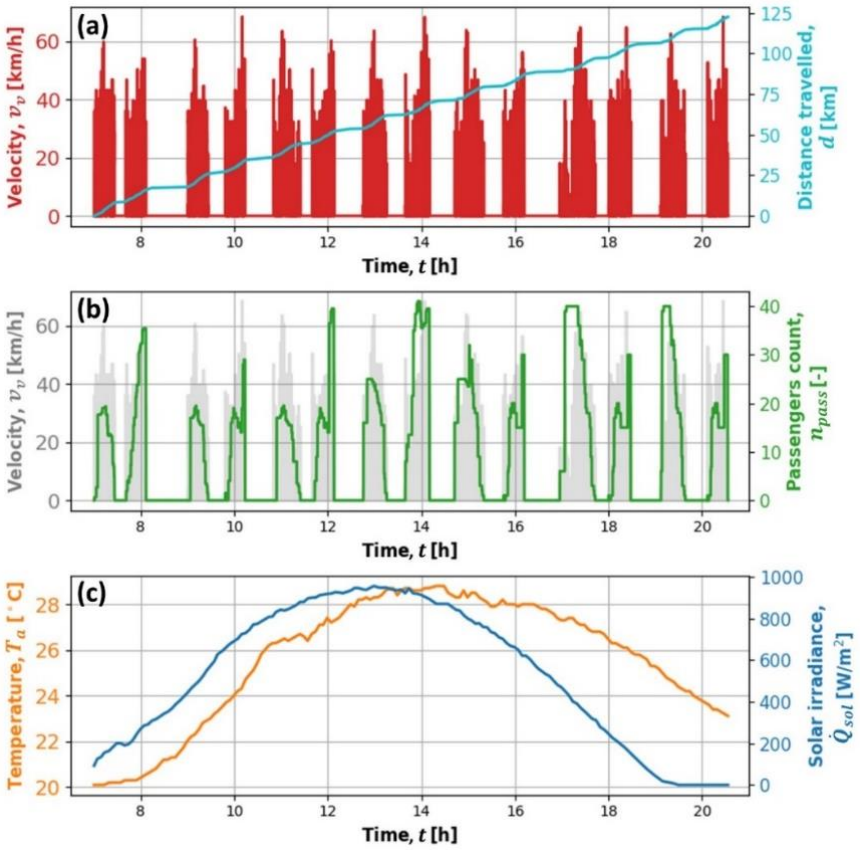


Fig. 3 Recorded city bus driving cycle time profile data: vehicle velocity and distance travelled (a), ridership (b), and ambient temperature and solar irradiance (c)

Fig. 4a shows the scatter plots of recorded altitude data in relation to distance travelled for direction Binyenei HaUma–Talpiot (further abbreviated as A–B) and multiple trips. The reconstructed road slope profile is shown in Fig. 4b. The driving direction A-B is characterised by mostly downhill driving with the road slope peaks up to 5 deg. In order to reduce noise in the reconstructed road slope profile, before being differentiated the recorded altitude profiles were averaged and filtered by a low-pass double-sided Butterworth filter of third order (see solid line in Fig. 4a). The results for the opposite direction (B-A) are not shown for the sake of brevity.

Fig. 5 shows the recorded battery SoC and cumulative energy consumption time profiles corresponding to the driving cycle in Fig. 3a. These profiles are used as a reference for e-bus model parameterization. By linearly extrapolating the energy consumption profile over the whole SoC range [0,1] and subtracting the observed end values, one obtains the total battery capacity of 292.5 kWh, which equals 91% of the declared, new bus battery capacity of 324 kWh. The difference between the two battery capacity values can be attributed partly to nonlinear battery behaviour, and partly to battery aging (the bus was produced in 2017).

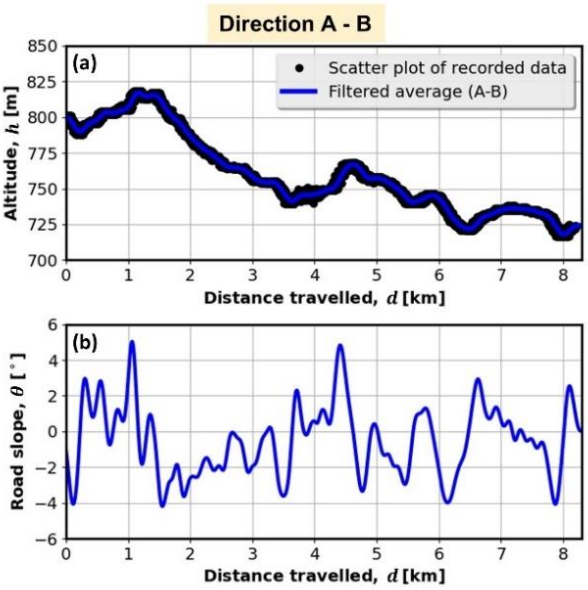


Fig. 4 Reconstructed road altitude (a) and road slope profiles (b) with respect to distance travelled

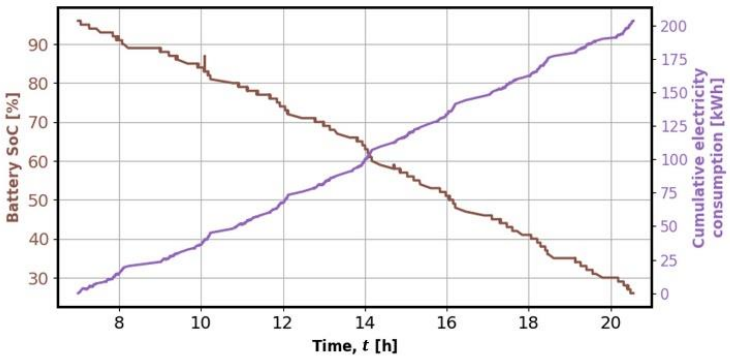


Fig. 5 Time profiles of battery SoC and cumulative battery energy consumption

3.2.2 Powertrain model

The powertrain of the considered fully electric city bus is modelled in the backward-looking manner, i.e. in the direction from the wheels towards the electric machine. The driving cycle-defined vehicle velocity (v_v), road slope (θ) and ridership inputs (Subsection 3.2.1) are fed into the vehicle longitudinal dynamics equations to calculate the total wheel torque and the wheel speed [2]:

$$\tau_w = r_w M_v \dot{v}_v + r_w R_0 M_v g \cos(\theta) + r_w M_v g \sin(\theta) + 0.5 r_w \rho_{air} A_f C_d v_v^2, \quad (3.1)$$

$$\omega_w = \frac{v_v}{r_w}, \quad (3.2)$$

where r_w is the tire effective radius, $M_v = M_{v0} + M_{pass}$ is the sum of the empty vehicle mass (M_{v0}) and the total passengers' mass (M_{pass}), R_0 is the rolling resistance coefficient, ρ_{air} is the air density, A_f is the bus frontal area, C_d is the aerodynamical drag coefficient and g is the gravity acceleration. The individual passenger mass is estimated to 68.125 kg to make a full bus with the passengers' capacity of 80 match the declared maximum vehicle payload. Therefore, the passenger mass M_{pass} is calculated as $68.125 \cdot n_{pass}$, where the ridership n_{pass} is given in Fig. 3b.

The e-machine torque (τ_{MG}) and the speed ω_{MG} are calculated as:

$$\tau_{MG} = \frac{\tau_w \eta_{tr}^{k_t}(\tau_w) + \frac{P_0(\omega_w)}{\omega_w}}{i_0}, \tag{3.3}$$

$$\omega_{MG} = i_0 \omega_w, \tag{3.4}$$

where i_0 is the final drive ratio, while $\eta_{tr}(\tau_w)$ and $P_0(\omega_w)$ are the drivetrain efficiency and the idle power loss maps, respectively, with k_t being defined as -1 for $\tau_w > 0$ (motoring) and 1 for $\tau_w \leq 0$ (regenerative braking). These maps have been reconstructed by properly scaling the maps available in literature with respect to maximum speed and power ratios of the particular e-bus and the reference vehicle from literature (see [3] for more details).

The e-machine efficiency η_{MG} is modelled by a map dependent on the e-machine speed and torque (see Fig. 6), from which the e-machine power load to the battery is calculated as:

$$P_{MG} = \eta_{MG}^{k}(\tau_{MG}, \omega_{MG})\tau_{MG}\omega_{MG}, \tag{3.5}$$

where the exponent k depends on the e-machine operating mode: $k = -1$ for motoring ($P_{MG} > 0$), and $k = 1$ for regenerative braking ($P_{MG} < 0$). The map in Fig. 6 is adopted from the map published in [4] for a similar M/G machine and scaled based on the Willans line method with respect to maximum speed and power ratios of the particular and reference M/G machine.

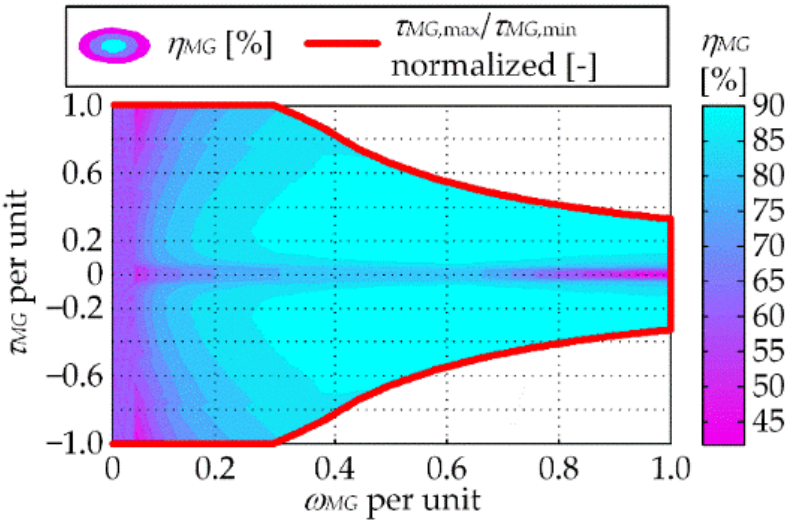


Fig. 6 Normalised efficiency map and maximum torque characteristics of e-machine

3.2.3 Battery model

The battery model is based on a single cell model scaled up to the appropriate number of serially connected cells contained in the battery pack. The single cell equivalent circuit model (ECM) has been developed based on the available data from the SAFT VL30PFe cell datasheet and reference [5]. The battery ECM is shown in in Fig. 7a, and it consists of the open-circuit voltage source (U_{oc}) and the internal resistance (R_{int}). Both parameters are made dependent on the battery SoC, as shown in Fig. 7b. Temperature dependencies of both parameters are neglected, since it is assumed that the e-bus includes an effective battery thermal management system.

The battery SoC dynamics are described by state equation:

$$\dot{SoC} = -\frac{I_{batt}}{Q_{\max}} = \frac{\sqrt{U_{oc}^2(SoC) - 4R_{int}(SoC)P_{batt}} - U_{oc}(SoC)}{2Q_{\max}R_{int}(SoC)}, \tag{3.6}$$

where I_{batt} is the battery current, $Q_{\max}$ is the battery charge capacity, $SoC = Q / Q_{\max}$ is the state of charge, and P_{batt} is the total battery power including the e-machine power P_{MG} given by Eq. (3.5), and the powers of auxiliary devices (P_{aux}) and HVAC system (PHVAC) determined by the models described in the next two subsections:

$$P_{batt} = P_{MG} + P_{aux} + P_{HVAC}. \tag{3.7}$$

Note that the slowly changing SoC variable is the only state variable of the overall e-bus backward-looking model (a quasi-static model). The battery charge capacity is obtained from the energy capacity $E_{\max}$ = 292.5 kWh as $Q_{\max} = E_{\max} / U_{oc}(SoC = 50\%)$ = 459 Ah.

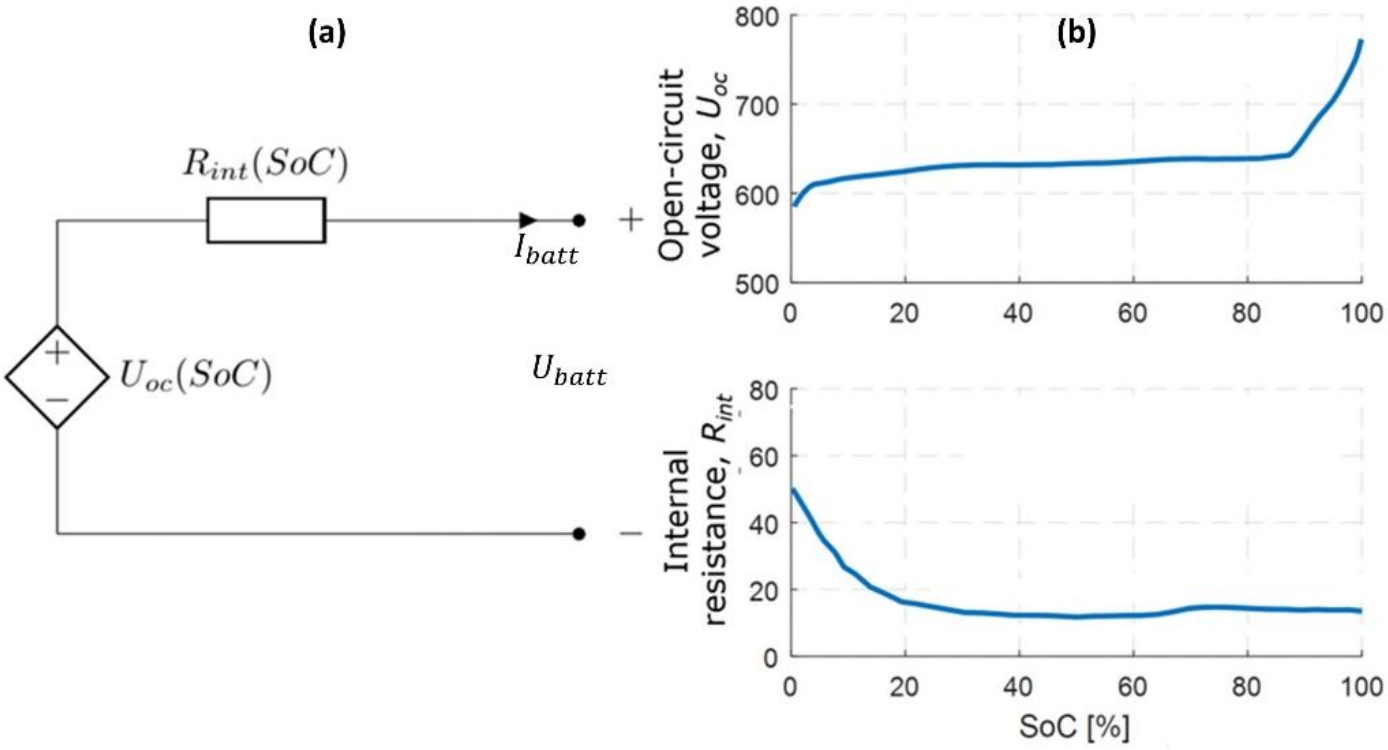


Fig. 7 Battery equivalent circuit (a) and SoC dependencies of open-circuit voltage and internal battery resistance for considered LFP battery (b)

3.2.4 Model of auxiliary devices

The main auxiliary devices considered include servo steering, air compressor supplying brakes and air suspension, and DC/DC converter supplying low voltage auxiliary devices (e.g. wiper, electronic devices, light beams and similar). The power consumption of these devices is modelled based on their nominal power and a power-modulating binary signal (see [1] and references therein). The nominal power values, and the values of binary signal duty cycle and period are given in Table 1 for the three auxiliary devices. Note that the DC/DC converter power load is set to be constant. The total auxiliary device load P_{aux} is obtained by summing up the contributions of each device load (simulated according to the parameters in Table 1) and fed to the total battery load expression (3.7).

Table 1 Values of nominal power ($P_{aux,N}$), duty cycle (d_c) and duty cycle period (t_p) of modelled auxiliary devices

Auxiliary device	$P_{aux,N}$ [W]	d_c [-]	t_p [s]
Servo steering	2500	0.09	400
Air compressor	2000	0.15	100
DC/DC converter with low voltage devices	184	1	N/A

3.2.5 HVAC system model

The presented HVAC modelling method can be applied for both A/C and heat pump modes. Since the considered driving cycle (Fig. 3) corresponds to a hot summer day, the model parameterization is presented for the A/C mode. The main assumption is that the HVAC device response is much faster than the cabin thermal transients, so that the overall system is represented by the bus cabin model depicted in Fig. 8. A proportional-integral-derivative (PID) controller commands the cooling power $\dot{Q}_{HVAC}$ to maintain the cabin temperature T_{cab} at it reference value $T_{cab,R}$. The cooling power $\dot{Q}_{HVAC}$ is limited in accordance with the datasheet of assumed HVAC device (Eberspächer AC136 AE HP HVAC system, [6]). The reference variable $T_{cab,R}$ is generated in dependence of the ambient temperature T_a (see cyan line in Fig. 8), which is set to fall between the bounds defined by VDV 236:2015 guidelines for public transport (red and green lines). Based on the assumption of fast HVAC system response and the assumption of constant coefficient of performance (COP = 1.8), the HVAC power consumption P_{HVAC} from Eq. (3.7) is determined as $\dot{Q}_{HVAC}/COP$.

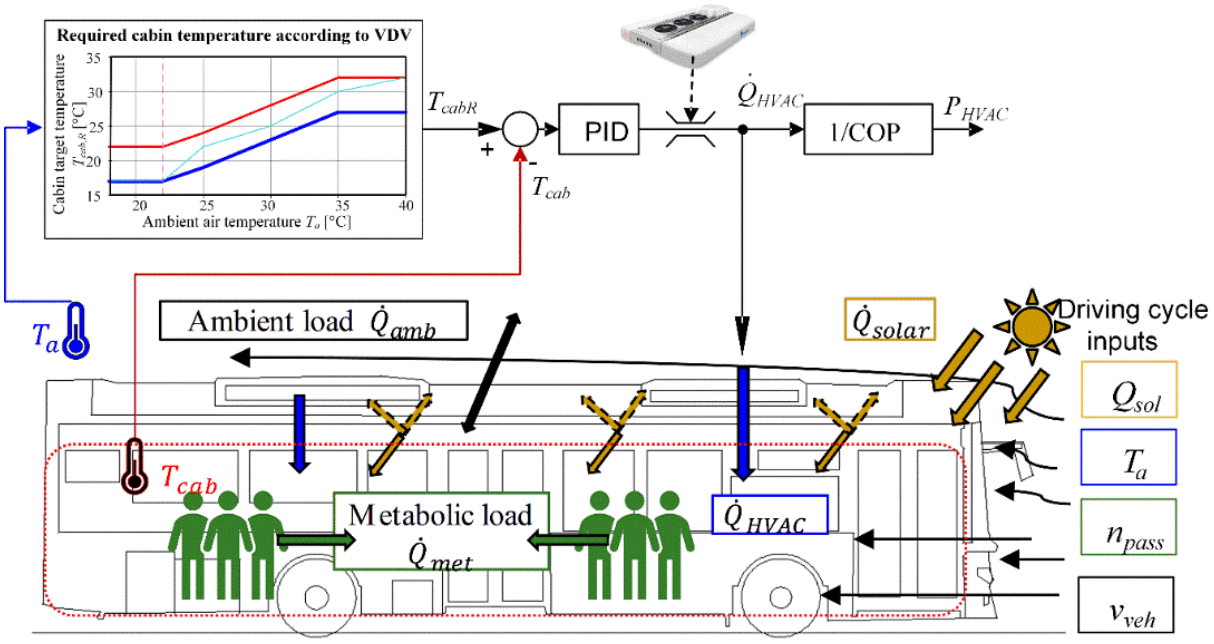


Fig. 8 Illustration of HVAC system energy consumption model

The thermal dynamics model includes four thermal masses (Fig. 8): cabin air, interior body, and inner and outer chassis shells; and different thermal loads: conduction, convection and radiation between the thermal masses, solar irradiance, and passenger metabolic load. The model is implemented in Dymola as illustrated in Fig. 9. The model inputs include the ambient temperature (T_a), the solar irradiance ($\dot{Q}_{sol}$), the vehicle velocity (v_v) and the ridership (n_{pass}).

Certain parameters of the cabin thermal model were difficult to determine or estimate due to either lack of available data (e.g. bus body paint colour or glass tinting) or complex parameter dependencies (e.g. heat exchange between interior elements). There were five such parameters: the heat transfer coefficients combined with lumped interior elements surface for convective heat transfer between interior and cabin air, and between interior chassis shell and cabin air, the conduction coefficient between interior and exterior chassis shell, the glass transmissivity coefficient for interior solar irradiance load, and the combined transmissivity and absorptivity factor for solar irradiance on the body exterior surface.

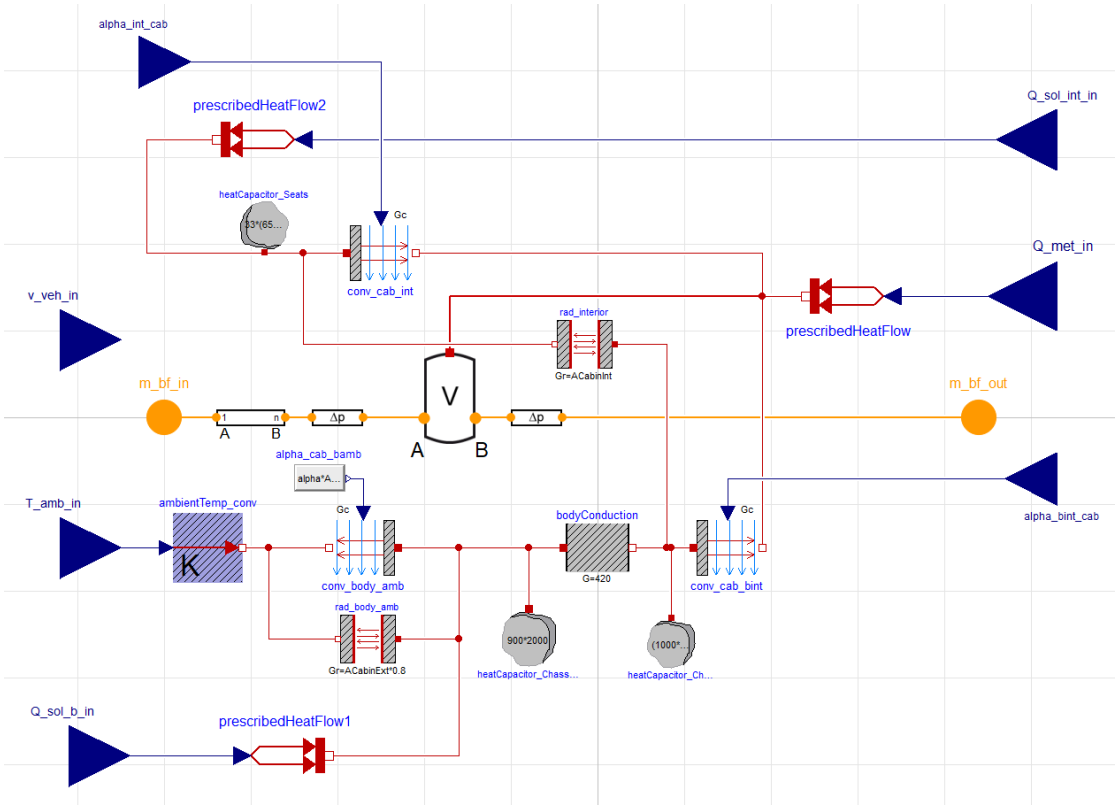


Fig. 9 E-bus cabin thermal model implemented in Dymola

The unknown cabin thermal model parameters have been determined through optimization by using modeFrontier software. The optimization setup is illustrated by the block diagram shown in Fig. 10. The overall model used in the optimization setup includes not only the Dymola thermal model but also the powertrain Python model. This is to obtain simulation responses of the battery SoC and the overall energy consumption $E_{sim} = \int P_{batt} dt$, which are compared with the recorded SoC and energy consumption responses to generate the corresponding RMS errors fed to the optimization genetic algorithm MOGA-II to minimise those errors. The two-objective optimization has resulted in a Pareto

frontier of optimal solutions. The selected solution corresponds to a low energy consumption RMS error, and it results in a favourable overall fit accuracy (partly because of a better resolution of the recorded energy consumption signal than the recorded SoC signal).

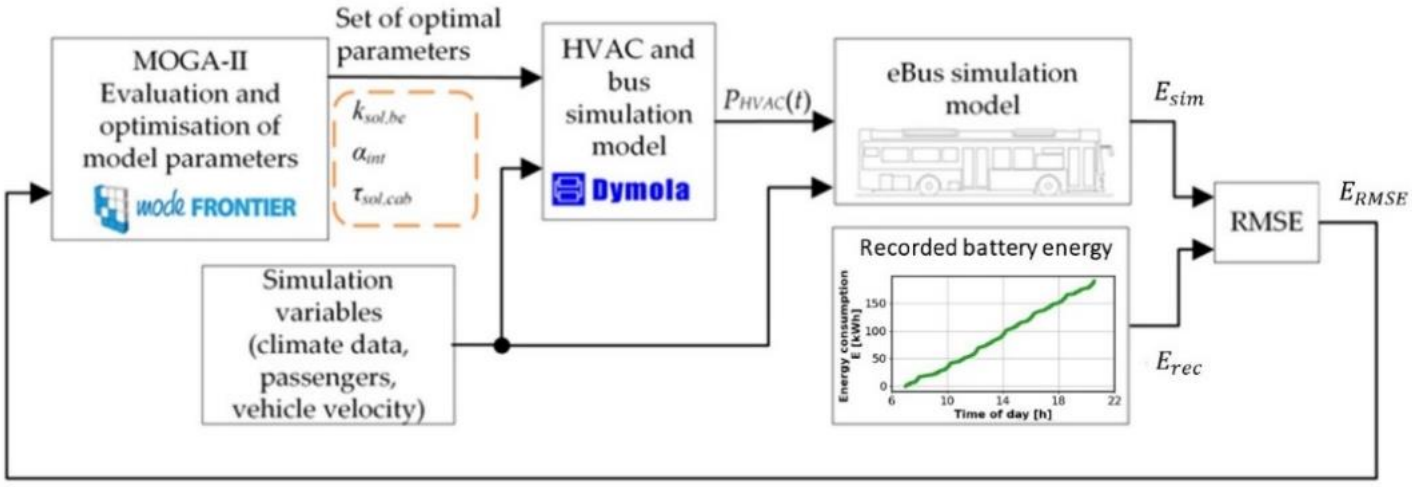


Fig. 10 Block diagram of optimization setup used to determine unknown parameters of bus cabin thermal model

The simulation profiles of e-bus model variables, obtained through the cabin thermal model parameter optimization and shown in Fig. 11, have further been used to optimise parameters of a HVAC regression model to be used within the e-bus backward-looking model. The regression model is linear in parameters and its inputs correspond to the inputs of the cabin thermal model (T_a, $\dot{Q}_{sol}$, v_v and n_{pass}). Matlab function *stepwiselm* available within Statistics and Machine Learning Toolobox has been used to select the model features and optimise its parameters. The selected model is given by:

$$P_{HVAC} = \beta_0 + \beta_1 T_a + \beta_2 \dot{Q}_{sol} + \beta_3 n_{pass} + \beta_4 v_{veh} + \beta_{14} T_a v_{veh} + \beta_{22} \dot{Q}_{sol}^{\ 2}. \quad (3.8)$$

Comparative responses of the actual and simulation responses of SoC, energy consumption and HVAC power, shown in Fig. 11, indicate very good modelling accuracy on the dataset used in model parameterisation (training).

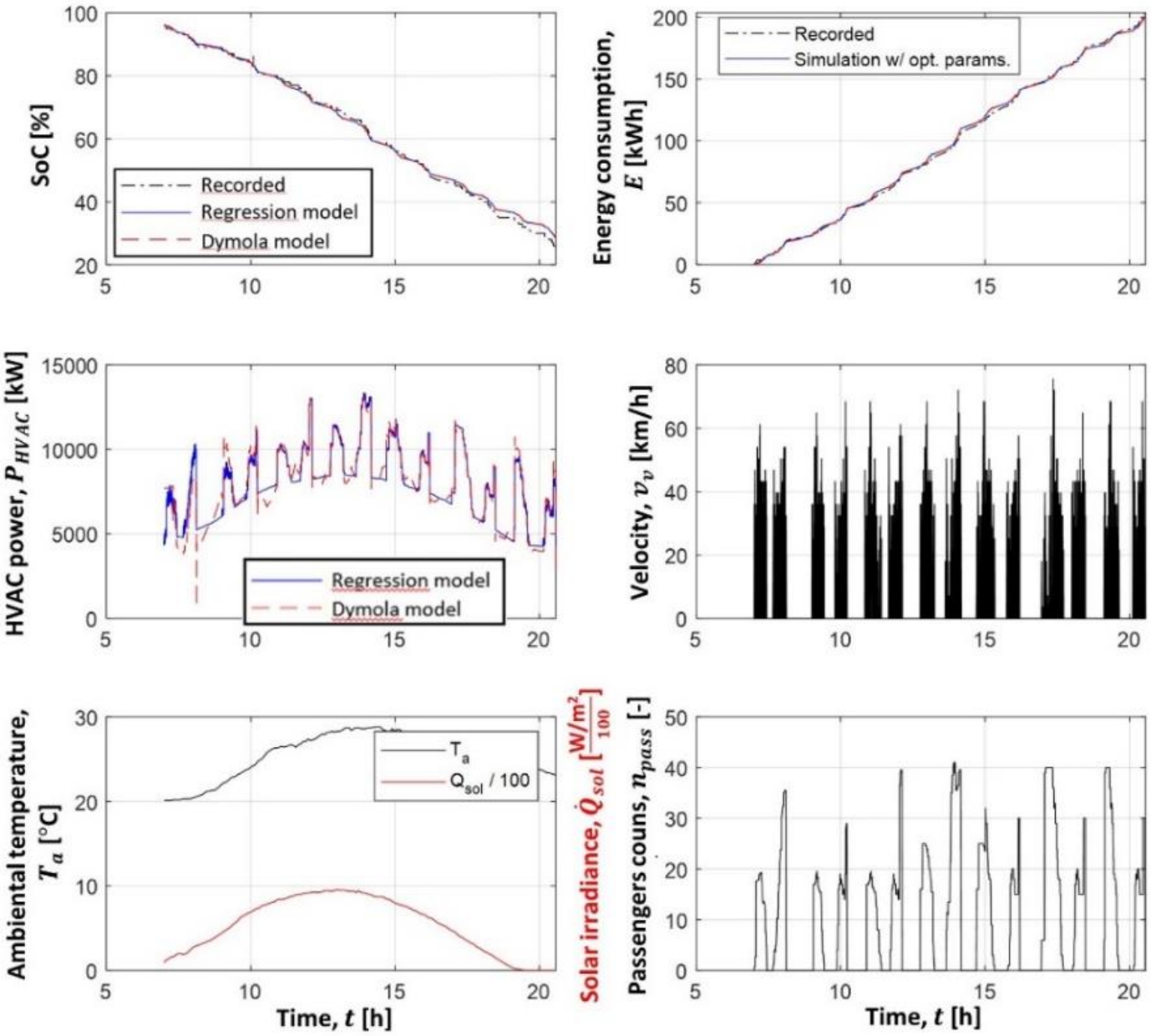


Fig. 11 Response of recorded e-bus model variables for dataset used in model training and corresponding simulation responses of SoC, energy consumption and HVAC power

3.2.6 E-bus model validation

For an unbiased assessment of modelling accuracy, the overall e-bus model has also been validated against a couple of other datasets (corresponding to different days of operation of the same bus on the same route during the same summer month). The results of the first validation, shown in Fig. 12a, confirm the very good modelling accuracy.

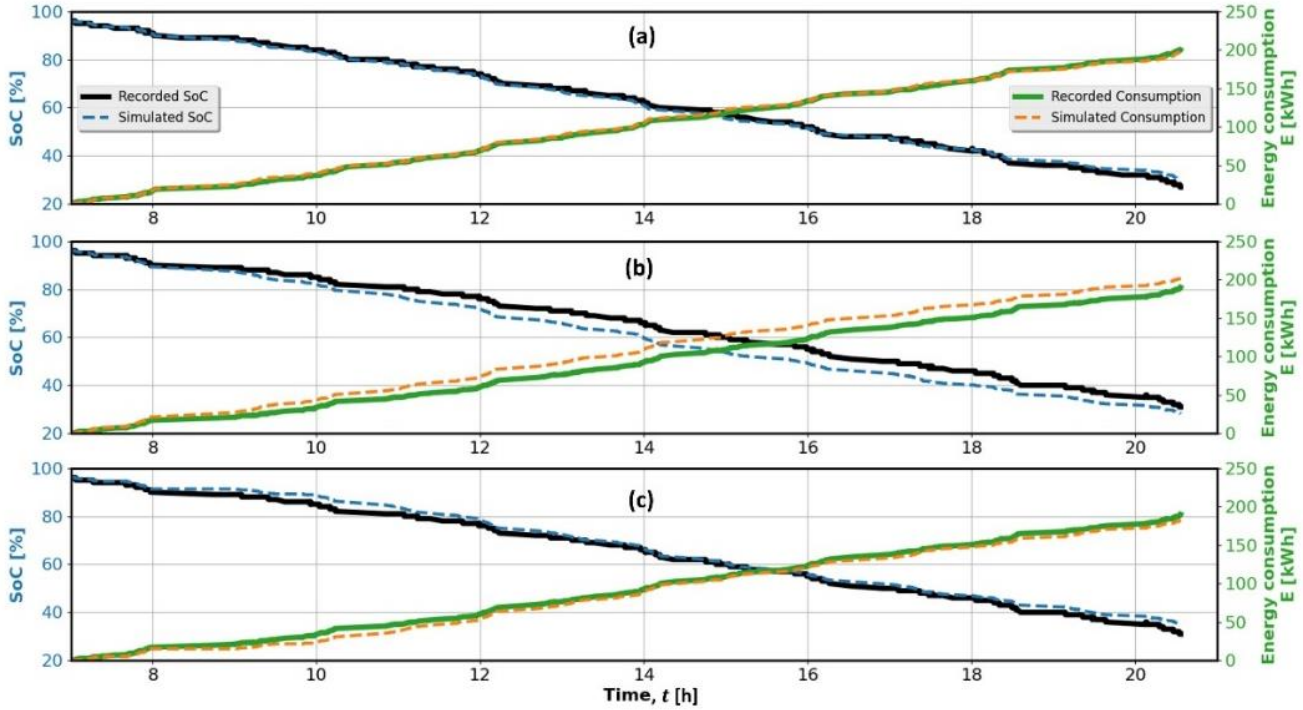


Fig. 12 E-bus model validation for first (a) and second validation dataset (b), as well as for second validation dataset but with simulated A/C system switched of from 7 am and to 10 am

However, the model performance degrades for the second validation (Fig. 12b) in terms of occurrence of SoC and energy consumption offsets during a relatively long bus pause (dwell time) at the end station after the second driving mission (i.e., after 8 am; see also the velocity profile in Fig. 11). It has been hypothesised that, unlike in the previous two datasets, the HVAC system was shut down during the morning hours since the ambient temperature was around the room temperature. Because the model presumed that the HVAC was active during the whole operation period, its SoC and energy consumption predictions persistently changed, thus accumulating the offset during the morning pause. In order to check the above hypothesis, the HVAC submodel is shut down in the period from 7 am to 10 am. The corresponding results shown in Fig. 12c indicate that the modelling accuracy is significantly improved when compared to the original response in Fig. 12b. A small offset is, though, still present in the SoC and energy consumption results around 10 am.

3.3 Physical model-based sensitivity analysis of e-bus energy consumption

Once the e-bus model is successfully validated, it can be used as a basis for energy consumption sensitivity analysis for a wide range of scenarios and operating conditions. Only the SoC trajectory results are presented below since the energy consumption responses directly correlate with the SoC ones (see Fig. 12).

3.3.1 Sensitivity analysis with respect to A/C state and ridership

The sensitivity analysis of battery SoC trajectory is first conducted with respect to different A/C states (on | off | full) and bus ridership (zero | medium (40) | full (80) | varying), in order to reveal the impact of these operating parameters on the e-bus range. The results shown in Fig. 13 indicate that the ambient conditions (i.e., A/C load) and ridership (i.e., the bus load) significantly affect the energy consumption, as the final battery SoC can be anywhere between 20% and 70% after approximately 9.5 h of operation. Accordingly, the e-bus range reduces from the extrapolated maximum value of 255 km to 87.5 km, which is the reduction of around 65%. When expressed per kilometre of ride, the energy consumption reduces from 2.41 kWh/km to 0.87 kWh/km (Table 2).

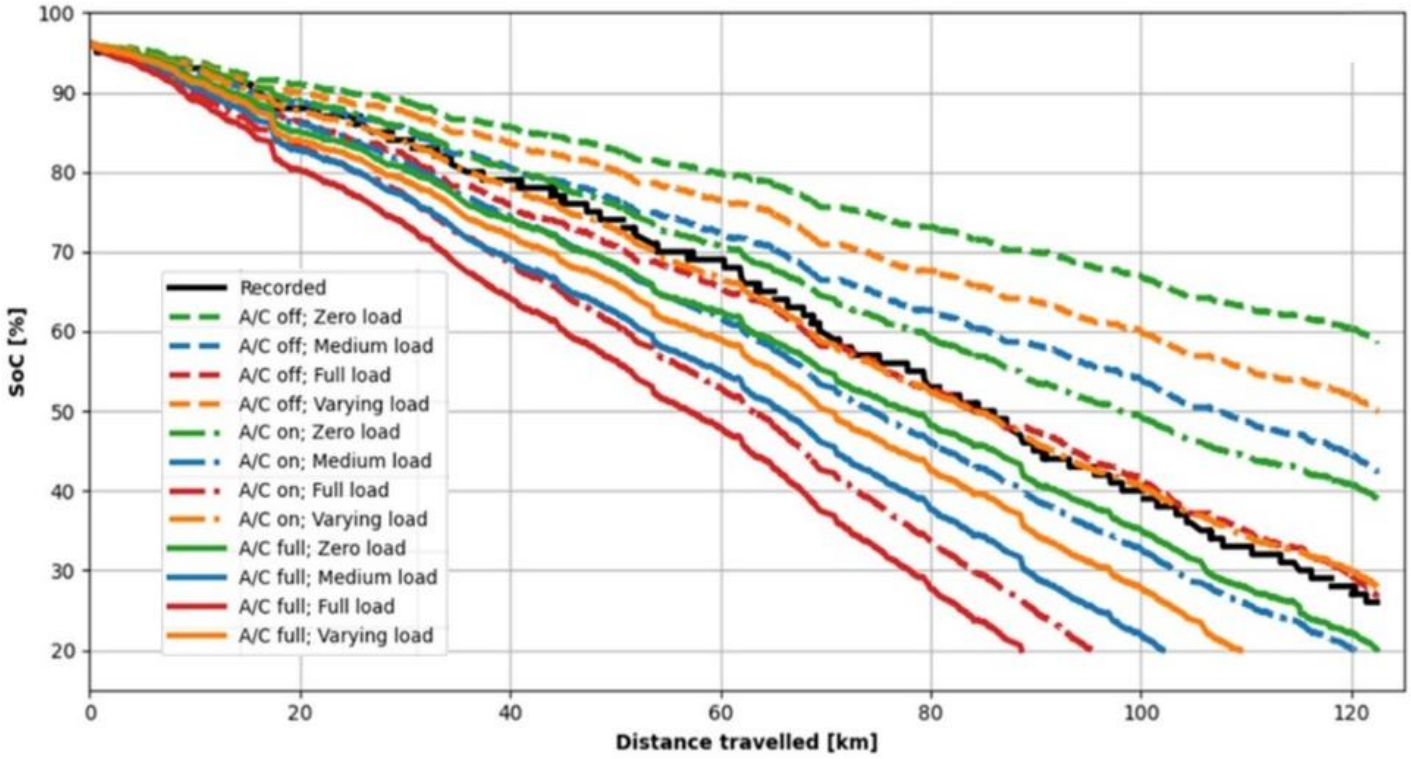


Fig. 13 SoC trajectories obtained for different levels of A/C and ridership load

Table 2 Specific energy consumptions for different levels of A/C and ridership load

A/C state	Passengers load			
	Zero	Medium	Full	Varying
On	0.87 kWh/km	1.24 kWh/km	1.59 kWh/km	1.06 kWh/km
Off	1.31 kWh/km	1.77 kWh/km	2.24 kWh/km	1.56 kWh/km
Full*	1.74 kWh/km	2.09 kWh/km	2.41 kWh/km	1.95 kWh/km

3.3.2 Sensitivity analysis with respect to period of daily operation

In order to gain insight into daily variation of the e-bus specific energy consumption, the simulation has been conducted over each individual trip along the day and separately for each driving direction. The model is reset to its nominal setting corresponding to actual (varying) A/C and ridership loads (as in Fig. 11). The obtained simulation and related recorded values of specific energy consumptions are shown in Fig. 14 for individual and combined driving directions. The same figure shows the corresponding average vehicle velocity data. The summarised results are plotted in Fig. 15. The results correspond only to actual driving missions, i.e. the dwelling periods at the end stations (and resulted A/C load) are disregarded.

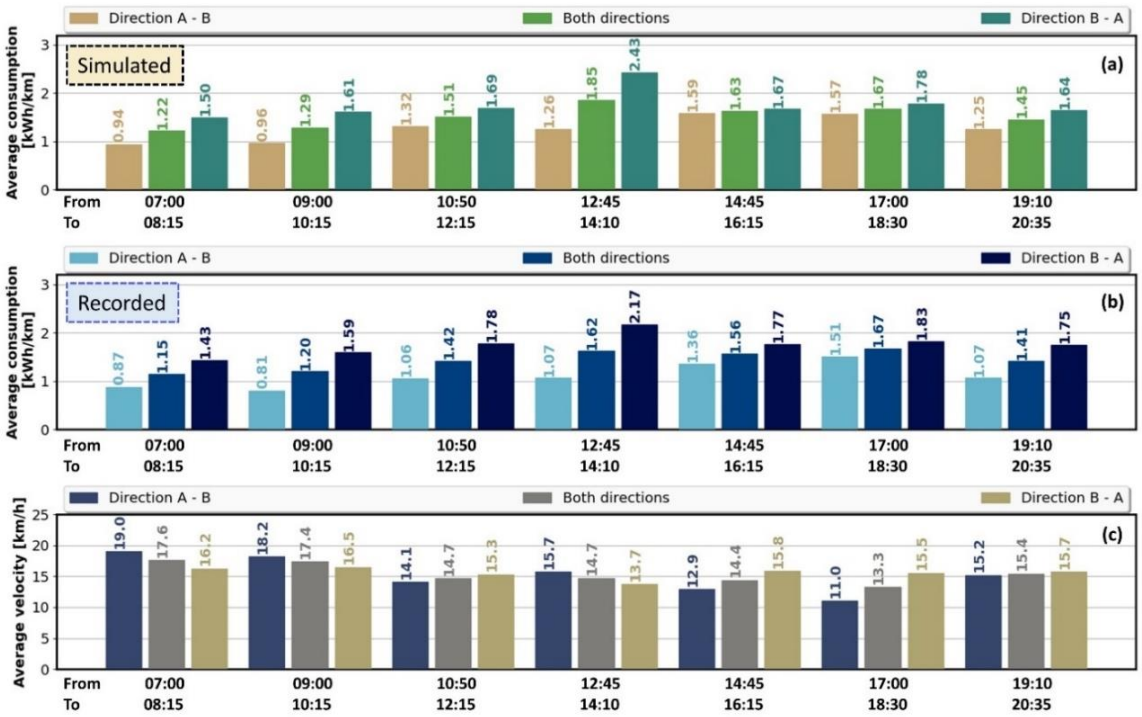


Fig. 14 Specific energy consumption values calculated per trip (and per direction) based on simulation results (a) and recorded data (b), and corresponding average values of recorded vehicle velocity (c)

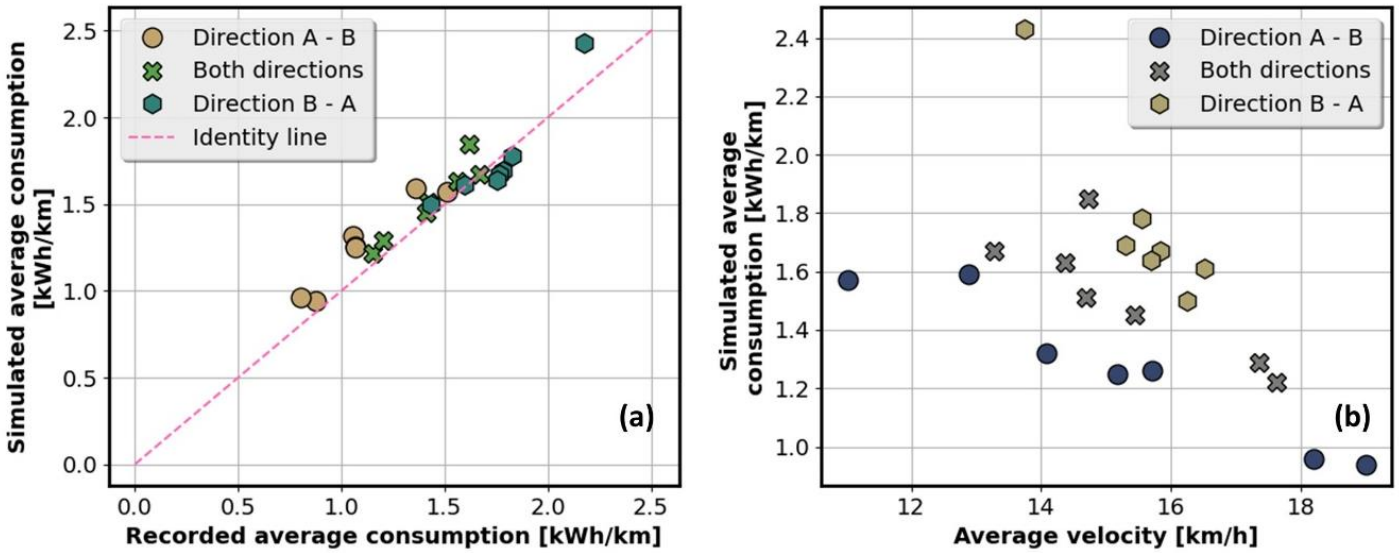


Fig. 15 Simulated vs. recorded values of specific energy consumption (a) and simulated specific energy consumption vs. average vehicle velocity (b)

The specific energy consumption varies significantly due to the effect of road slope (the consumption is lower for mostly downhill driving in Direction A, Fig. 14). The consumption variation is significant even for combined (two-way) trips (light green bar in Fig. 14a), which is due to due to the varying ambient, ridership and traffic conditions (cf. Fig. 3 and Fig. 14). The traffic condition influence is substantiated by clear correlation between the specific energy consumption and the average vehicle velocity, as shown in Fig. 14 and more clearly in Fig. 15b. The individual direction specific consumptions vary in the range from around 0.9 to 2.4 kWh/km, while for the two-way trips they fall in the range from 1.2 to 1.8 kWh/km. The good modelling accuracy is confirmed by fine agreement between the simulation-obtained and recorded value plots in Figs. 14a and 14b. This is better illustrated in Fig. 15a in terms of good alignment of simulation vs. recorded values with the ideal 1:1 line. Quantitatively, the plot in Fig. 15a is represented by the Pearson's correlation coefficient of 0.95 and the coefficient of determination is R^2 = 0.85, which are quite close to the ideal value of 1.

3.4 Data collection for data-driven e-bus modelling

3.4.1 Data collection framework

In the absence of a wide set of recorded e-bus energy consumption data, the framework depicted in Fig. 16 has been employed to generate the data needed for data-driven modelling. Initially, high-sampling-rate (1 Hz) data were acquired for a 12m electric city e-bus operating across a day on several

routes in the City of Jerusalem. The acquired data were employed for parametrization and validation of a physical e-bus model running on the same 1 Hz sampling rate (Subsections 3.2 and 3.3).

At the same time, low-sampling-rate (approx. around 0.25 Hz) data were collected from a fleet of around 300 conventional buses operating on 29 routes in Jerusalem over the period of one month. The recorded low-sampling-rate data were then transformed into the corresponding set of representative high-sampling-rate driving cycles corresponding to trips between two end stations. Those driving cycles were then fed to the developed physical e-bus model to obtain the energy consumption data. The transformation was based on the Markov chain synthesis method proposed in [7].

Finally, a wide set of trip-based statistical features (e.g., mean velocity, number of bus station stops, average ridership, trip duration, initial SoC, etc.) have been extracted from the synthetic driving cycles. They are paired with the physical model-based simulation data on energy consumption to form a dataset employed for the development of data-driven model in Subsections 3.5 and 3.6.

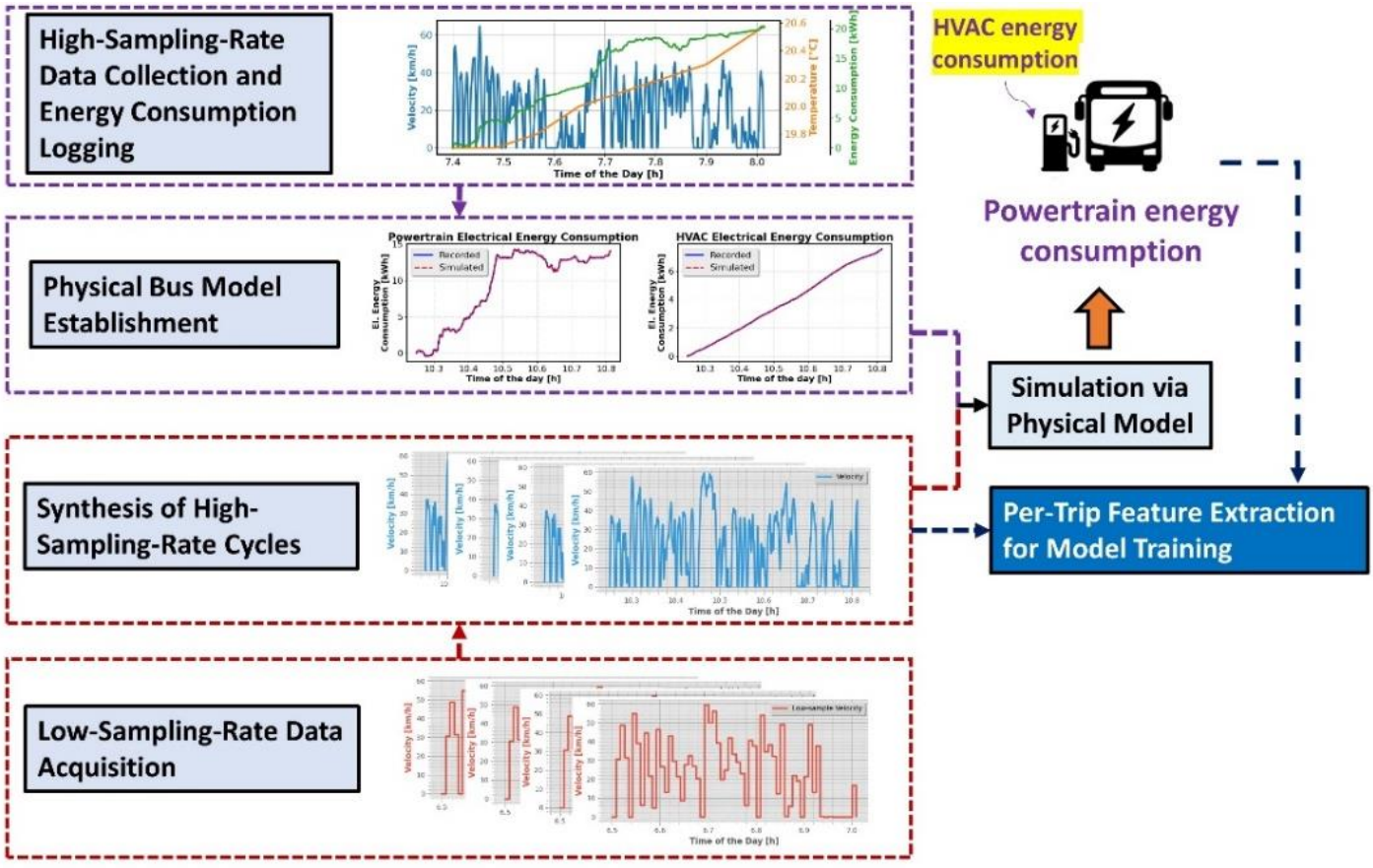


Fig. 16 Illustration of data collection framework

3.4.2 Generation of extrapolation datasets

In total, 4057 synthetic driving cycles were generated based on the Markov chain approach [7]. Each cycle was unique not only with respect to route (considering diverse road and traffic conditions, including varying road grades), but also with respect to time of trip (considering fluctuating traffic and ridership conditions). Additionally, each driving cycle had a distinct initial battery state of charge (SoC).

To rigorously assess the extrapolation ability of the data-driven model (i.e., its generalization properties), four additional sets of driving cycles were derived from the basic set of synthetic driving cycles (Set #1):

- Set #2: Faster and shorter trips: For each trip, the mean velocity of every bus station-to-station segment is amplified by 50%, and the travelled distances are randomly reduced.
- Set #3: Flat roads: the road slope is set to zero.
- Set #4: Steeper roads scenario: the road grade profile is scaled up by 50%.
- Set #5: Faster trips: The mean velocity of each station-to-station segment is amplified by 50%.

Fig. 17 shows histograms of the main driving cycles features for all the five individual datasets and a data set composed of the individual ones (an aggregate dataset). The corresponding histogram of powertrain energy consumption per trip is given, as well. When compared to the basic dataset #1, the modified datasets extend the range of features, thus making the aggregate dataset wider and flatter.

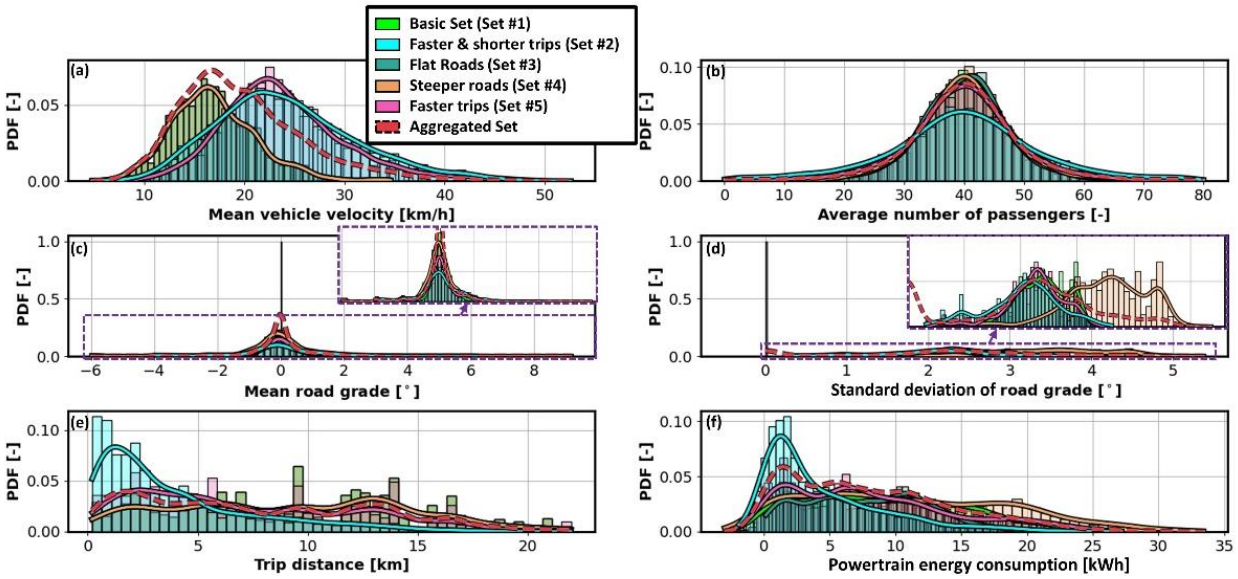


Fig. 17 Distributions of main features of standard, modified, and aggregate driving cycle sets and corresponding distribution of powertrain energy consumption

3.5 Feature selection for data-driven e-bus powertrain modelling

Feature selection is an integral component of machine learning and data analytics. It is aimed at enhancing the model accuracy and simplicity by identifying and retaining only the most relevant features. The presented feature selection method corresponds to e-bus powertrain (and auxiliary devices) modelling only, because HVAC modelling represents an independent and straightforward trip-based modification of the approach presented in Subsections 3.2 and 3.3 (see Subsection 3.6).

3.5.1 Performance metrics and validation strategy

Two metrics are employed to energy consumption model residuals to evaluate the modelling accuracy [8]: (i) root mean square error ($RMSE$) and (ii) coefficient of determination (R^2). To reduce the number of model inputs, the powertrain energy consumption is normalized with respect to travelled distance. The output predicted by such a normalized model (i.e., specific energy consumption in kWh/km) is in the final stage multiplied by the travelled distance to calculate the absolute energy consumption in kWh. The model performance metrics R^2 and $RMSE$ metrics are computed with respect to final model output, i.e. the absolute energy consumption.

In the model evaluation, a five-fold cross-validation method has been applied to the basic dataset (Set #1, Subsection 3.4), as depicted in Fig. 18. The basic dataset is randomly partitioned into five sections, termed *folds*. In each iteration of this method, a single fold was designated for model validation, with the remaining four folds serving for training. This process yields individual scores $R^2_{tr,i}$ and $R^2_{val,i}$, i = 1,...,5, for training and validation in each iteration, from which lumped/average scores R^2_{tr} and R^2_{val} are derived (Fig. 18).

In the sixth iteration, the model is trained on the whole (unpartitioned) basic dataset. The obtained model is then applied to the extrapolation datasets (Sets #2-#5), thus resulting in the validation scores $R^2_{s,j}$, j = 2,...,5 (Fig. 18). Finally, the combined validation score R^2_{total} is obtained from the residuals calculated by merging the predicted outputs from the validation iterations ($\hat{y}_{val,i}$ for $i = 1, \dots 5$) with the predicted values for the extrapolation sets ($\hat{y}_{s,j}$ for $j = 2, \dots 5$), and subtracting them with their true-value counterparts. The described validation process (Fig. 18) is applied when evaluating both R^2 and $RMSE$ metrics.

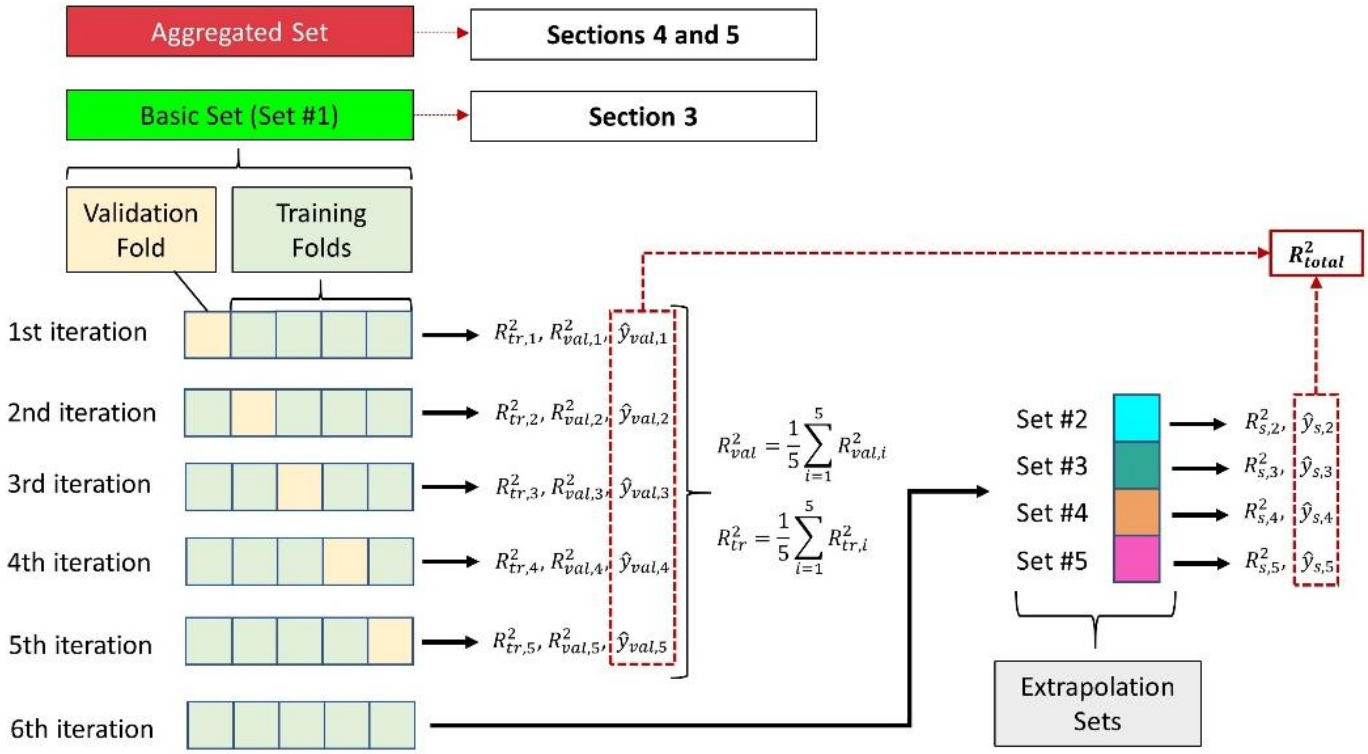


Fig. 18 Schematic representation of model validation strategy

3.5.2 Quadratic regression model

Feature selection have been applied by using the following linear-in-parameter quadratic model:

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_1^2 + \beta_6 X_2^2 + \beta_7 X_3^2 + \beta_8 X_4^2 + \beta_9 X_1 X_2 + \beta_{10} X_1 X_3 + \beta_{11} X_1 X_4 + \beta_{12} X_2 X_3 + \beta_{13} X_2 X_4 + \beta_{14} X_3 X_4, \quad (3.9)$$

where $\hat{y}$ is the dependent variable (here specific powertrain consumption), $X_1, X_2, \dots, X_n$ are the predictor variables (with n = 4 in the example of Eq. (3.9)), β_0 is the is the y-intercept parameter, and $\beta_1, \beta_2, \dots, \beta_m$, $m = 2n + \frac{n(n-1)}{2}$, are the model parameters corresponding to individual features.

The considered predictor variables include (see the dark blue block in Fig. 16): total number of route stations $N_{stations}$, number of stations that the bus actually stopped at, N_{stops}, ratio of stopping to total stations $\rho_{stops} = \frac{N_{stops}}{N_{stations}}$, mean velocity μ_v, average ridership $\bar{n}_{pass}$ and standard deviation of ridership σ_{pass}, trip duration t_{trip}, trip distance d_{trip}, initial state of charge SoC_{init}, mean road grade μ_{rg}, and standard deviation of road grades σ_{rg}. With this set of n = 11 predictor variables, the number of quadratic model features equals m = 77.

3.5.3 Feature selection

LASSO (Least Absolute Shrinkage and Selection Operator) technique [9] applies a penalty to the absolute values of regression parameters β_i, $i = 1, \dots, m$, thus encouraging parameters corresponding to non-influential features to diminish (see Fig. 19). This shrinkage mechanism is controlled by the penalty coefficient lambda λ. As λ grows, more model parameters converge to zero.

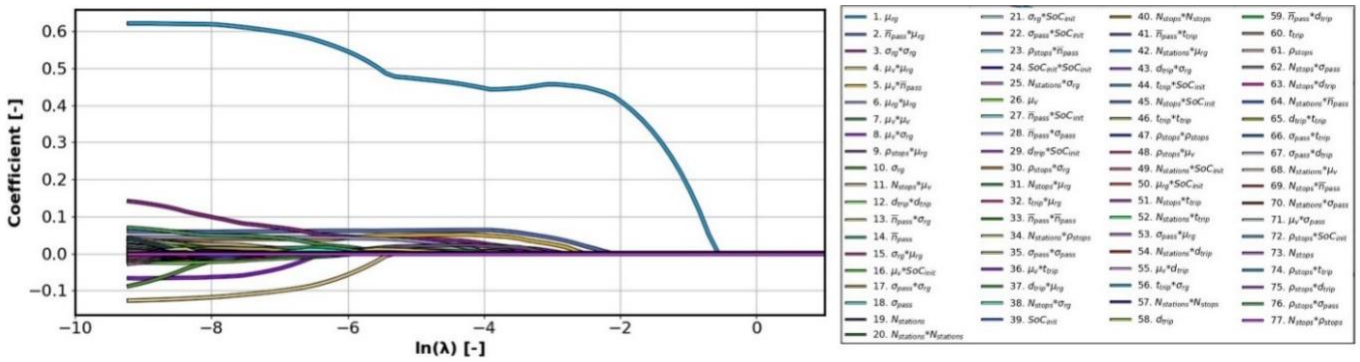


Fig. 19 Illustration of LASSO feature selection technique in particular case of n = 11 predictor variables and m = 77 features of energy consumption quadratic regression model

Random forest importance approach assigns importance scores to features based on their frequency in splitting data, indicating their contribution to the prediction accuracy. This relative feature importance is illustrated in Fig. 20.

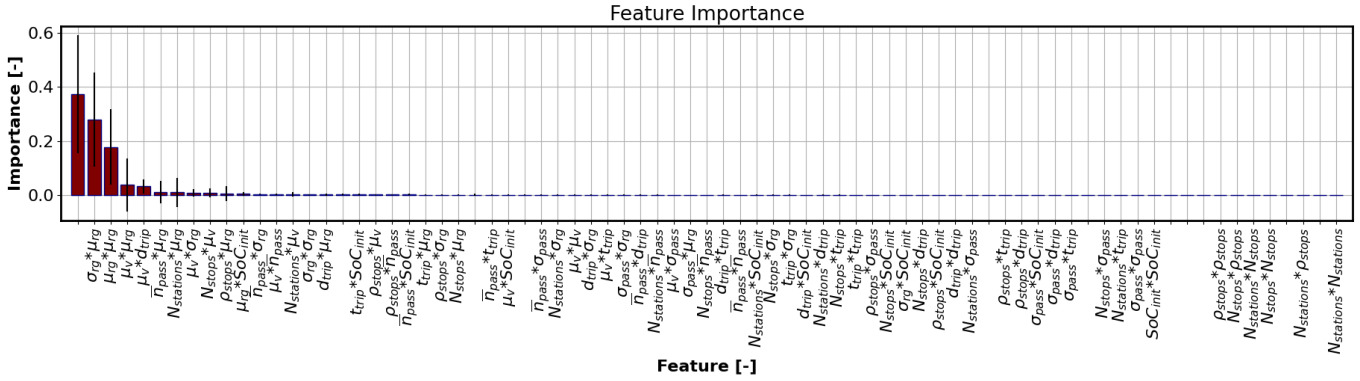


Fig. 20 Feature importance distribution as determined by Random forest importance analysis

The quadratic regression model has been re-trained by sequentially adding individual features based on their significance ranking provided by LASSO and Random forest importance approaches. The results are shown in Fig. 21 based on the R^2_{total} validation metrics introduced in Subsection 3.5.1. They indicate that the LASSO approach achieves peak performance with a smaller number of features compared to Random forest importance method.

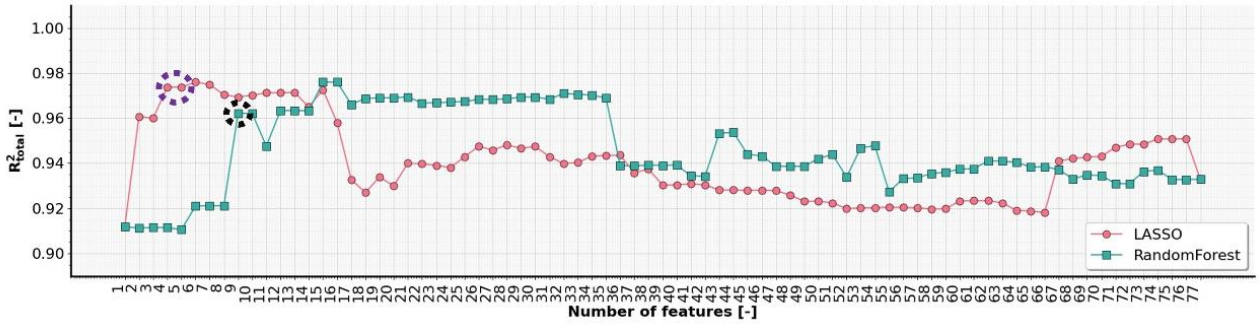


Fig. 21 Comparative plots of aggregate R^2 values for LASSO and random forest importance feature selection methods

Wrapper methods are used in model optimization for systematic feature selection. These methods select the best feature subsets by building and evaluating models [9]. **Forward feature selection, Backward feature elimination, and Stepwise regression** are characteristic methods from this category. Each method identifies an optimal set of regression model features based on the Bayesian Information Criterion (BIC):

$$BIC = k\ ln(\sigma^2) + (m+1)ln\ (k), \tag{3.10}$$

where $m+1$ represents the number of model parameters (including intercept), k signifies the number of observations (sample size), and σ^2 represents the average of the squared differences between the observed values and the values predicted by the model, quantifying the model prediction error. A lower BIC index suggests a better model fit.

To determine the optimal set of features, with a focus on both performance and number of terms, specific thresholds for each method were fine-tuned. Forward Feature Selection begins with no features, and continues with progressively adding them based on model fit improvement until the BIC value increase surpasses a threshold of 100. Backward Feature Elimination begins with all features and removes them to improve the model, while stopping when the BIC falls below the threshold of 150. Stepwise Regression combines both methods, adjusting features based on fit with the threshold of 450 and the removal threshold of 400.

Best subset method searches through all combinations of features to identify the optimal model subset. Due to the high computational demand, the number of predictor variables is reduced to $n = 4$ variables highlighted by feature selection results in Figs. 19 and 20: mean road grade, standard deviation of road grade, average number of passengers, and mean velocity. This leads to the quadratic regression model given by Eq. (3.9) and having $m = 14$ features. Consequently, 16,383 distinct linear regression models can be produced. The performance of each model is depicted in Fig. 22 by a point, which gives the values of validation metrics R^2_{total} and $RMSE_{total}$ defined in Subsection 3.5.1.

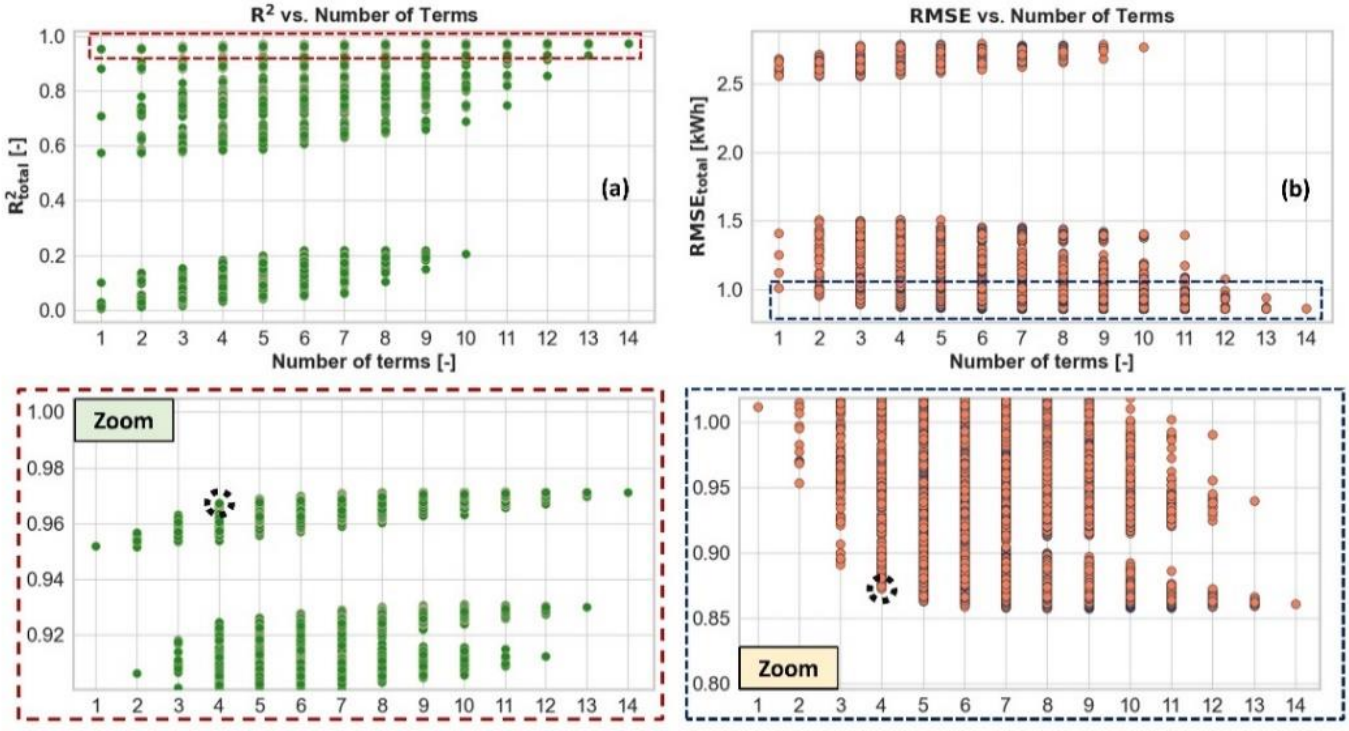


Fig. 22 Best subset method validation results

3.5.4 Comparative analysis of model gained by various feature selection methods

Different feature selection methods presented in Subsection 3.5.3 yield multiple candidate feature sets, which are summarized in Table 3. Four candidate sets, including from 3 to 6 features, are identified by the Best Subset method as a good trade-off of modelling accuracy and simplicity (see Fig. 22a). Although the LASSO metrics peak is at 6 features, and Random Forest at 15, simpler sets close to these peaks are preferred, influenced by the Best Subset approach emphasis on fewer features. So, LASSO models with 4 and 5 features are selected in Table 3, while Random Forest highlights a 9-feature model (see marked sets in Fig. 21). The wrapper methods are represented by a single optimal configuration each.

Out of the total of 10 configurations listed in Table 3, the four-feature one given by the best subset method (denoted in bold in Table 3 and marked in Fig. 22) has been selected for further analysis in Subsection 3.6. This is because its score R^2_{total} = 0.9755 nearly matches the top score R^2_{total} = 0.9763 of the best-subset model with six features. Moreover, minimal variance in R^2 (and $RMSE$) among different data sets points to a consistent performance of the selected model, alongside with a good interpretability (e.g., there is only a single interaction term - the one between mean velocity and average ridership).

Table 3 Comparative performance metrics of optimal models obtained by various feature selection methods

Number of features	Selected features	R^2_{tr} $RMSE_{tr}$	R^2_{val} $RMSE_{val}$	$R^2_{s,2}$ $RMSE_{s,2}$	$R^2_{s,3}$ $RMSE_{s,3}$	$R^2_{s,4}$ $RMSE_{s,4}$	$R^2_{s,5}$ $RMSE_{s,5}$	R^2_{total} $RMSE_{total}$
LASSO								
4	$\mu_{rg}, \sigma^2_{rg}, \mu_v * \mu_{rg}, \mu_v{}^*\bar{n}_{pass}$	0.9777 0.8873	0.9776 0.8862	0.9650 0.8720	0.9829 0.5911	0.9575 1.5901	0.9592 1.2487	0.9738 1.0126
5	$\mu_{rg}, \sigma^2_{rg}, \mu_v{}^*\mu_{rg}, \mu_v{}^*\bar{n}_{pass}, \bar{n}_{pass}{}^*\mu_{rg}$	0.9776 0.8883	0.9776 0.8881	0.9656 0.8649	0.9816 0.6124	0.9568 1.6032	0.9601 1.2350	0.9737 1.0153
Random Forest importance								
9	$\mu_{rg}, \mu^2_{rg}, \sigma_{rg}{}^*\mu_{rg}, N_{stations}{}^*\mu_{rg}, \mu_v{}^*\mu_{rg}, \mu_v{}^*d_{trip}, N_{stops}{}^*\mu_v, \bar{n}_{pass}{}^*\mu_{rg}, \mu_v{}^*\sigma_{rg}$	0.9752 0.9346	0.9750 0.9381	0.9645 0.8783	0.9312 1.1850	0.9374 1.9290	0.9576 1.2731	0.9621 1.1896
Forward selection								
8	$\mu_{rg}, \mu_v{}^*\bar{n}_{pass}, \sigma^2_{rg}, \mu^2_{rg}, \mu_v{}^*\mu_{rg}, \bar{n}_{pass}{}^*\mu_{rg}, \sigma_{rg}, t_{trip}{}^*\mu_{rg}$	0.9787 0.8673	0.9785 0.8682	0.9636 0.8895	0.8790 1.5719	0.9625 1.4927	0.9556 1.3017	0.9638 1.1652
Backward Elimination								
10	$\mu_{rg}, \mu^2_v, \mu_v{}^*\bar{n}_{pass}, \mu_v{}^*\sigma_{rg}, \mu_v{}^*\mu_{rg}, \bar{n}_{pass}{}^*\mu_{rg}, \bar{n}_{pass}{}^*SoC_{init}, \sigma_{pass}{}^*\sigma_{rg}, \sigma^2_{rg}, \mu^2_{rg}$	0.9781 0.8787	0.9780 0.8788	0.9656 0.8640	0.9464 1.0462	0.9617 1.5088	0.9571 1.2799	0.9710 1.0761
Stepwise Regression								
6	$\mu_{rg}, \mu_v{}^*\bar{n}_{pass}, \sigma^2_{rg}, \mu^2_{rg}, \mu_v{}^*\mu_{rg}, \bar{n}_{pass}{}^*\mu_{rg}$	0.9783 0.8755	0.9782 0.8750	0.9647 0.8757	0.9840 0.5719	0.9660 1.4222	0.9569 1.2825	0.9760 0.9839
Best Subset								
3	$\mu_{rg}, \sigma^2_{rg}, \mu_v{}^*\bar{n}_{pass}$	0.9778 0.8860	0.9777 0.8849	0.9662 0.8567	0.9828 0.5923	0.9574 1.5919	0.9591 1.2504	0.9739 1.0104
4	$\boldsymbol{\mu_{rg}, \mu^2_{rg}, \sigma^2_{rg}, \mu_v{}^*\bar{n}_{pass}}$	**0.9784** **0.8727**	**0.9784** **0.8721**	**0.9639** **0.8855**	**0.9825** **0.5978**	**0.9666** **1.4091**	**0.9546** **1.3161**	**0.9755** **0.9922**
5	$\mu_{rg}, \mu^2_{rg}, \mu_{rg}{}^*\sigma_{rg}, \sigma^2_{rg}, \mu_v{}^*\bar{n}_{pass}$	0.9786 0.8694	0.9785 0.8690	0.9642 0.8817	0.9821 0.6047	0.9681 1.3774	0.9547 1.3153	0.9759 0.9862
6	$\mu_{rg}, \bar{n}_{pass}, \mu^2_{rg}, \mu_{rg}{}^*\mu_v, \sigma^2_{rg}, \mu^2_v$	0.9781 0.8782	0.9781 0.8782	0.9666 0.8521	0.9823 0.6010	0.9662 1.4178	0.9582 1.2630	0.9763 0.9817
Note: All $RMSE$ values are given in kWh.								

3.6 Trip-based data-driven e-bus model

In Subsection 3.5, powertrain model features were selected (see Table 3), and the model was trained and validated on the basic dataset and then tested on four separate (extrapolation) datasets. Herein, a combined/aggregate dataset (see Fig. 17) is used for both training and validation (the training/validation folds in Fig. 18 are taken from the aggregate dataset). This approach aims to improve the modelling accuracy and allows for a direct performance comparison between the linear regression model and more complex machine learning algorithms, which often perform well at interpolation but face challenges with extrapolation. The training and validation metrics ($R^2_{tr},\ R^2_{val}, RMSE_{tr}, RMSE_{val}$) are obtained by using 5-fold cross-validation, as illustrated in Fig. 18.

3.6.1 Powertrain model

Training of the quadratic regression model, selected in Subsection 3.5 (see bold row of Table 3) and given by:

$$\frac{E_{pt}}{d_{trip}} = \beta_0 + \beta_1\,\mu_{rg} + \beta_2\,\mu_{rg}^2 + \beta_3\,\sigma_{rg}^2 + \beta_4\,\mu_v \bar{n}_{pass}, \tag{3.11}$$

on the aggregate dataset yields the performance metrics listed in the first row of Table 4. These metrics are nearly identical to the one listed in Table 3, thus highlighting the model robustness and generalizability.

To potentially improve the modelling accuracy, alternative machine learning algorithms were evaluated on the aggregate dataset and compared with the quadratic regression model (3.11). Most of these algorithms are set to use the individual predictor variables rather than quadratic and interaction terms/features present in the model (3.11) (see second column of Table 4). This is because these advanced algorithms should automatically detect/realize inherent interactions between individual predictor variables.

The evaluated machine learning algorithms and their main design parameters are summarized in what follows.

1. *LASSO Regression*: The parameter λ is set in the range from 0.0001 to 0 with increments of 0.00001.
2. *Ridge Regression*: The parameter λ is varied in the same range as for LASSO Regression.
3. *Decision Trees*: The maximum depth parameter ranges from 10 to 100, with increments of 1.
4. *Random Forest*: The number of estimators is in the range from 4 to 200, with increments of 1.
5. *Gradient Boosting*: The number of estimators is set in the same way as with Random forest method.

6. *K-nearest Neighbours*: The algorithm is set with neighbours ranging from 1 to 200.
7. *Support Vector Regression*: Various kernels, including Radial Basis Function, and 1st-, 2nd-, and 3rd-order polynomial are examined.
8. *Multilayer Perceptron (MLP) Neural Networks*: The number of layers and nodes varies from 1 to 4 and 16 to 512, respectively.
9. *1D Convolution Neural Networks*: The same architecture parameters are considered as with MLP neural network, all with the stride of 1.

Table 4 displays the best-performing configurations for each algorithm. Evidently, the advanced regression techniques do not considerably surpass the quadratic regression model when the validation performance is concerned, which is evidenced by the R^2_{val} index differing only at the third decimal place. Moreover, the advanced techniques typically perform poorly when tested on extrapolation datasets (see, e.g., results for high-order models in Table 3). So, even when the advanced models are trained on the aggregate dataset as done in Table 4, they may considerably underperform the quadratic regression model for real-world scenarios not fully captured by the aggregated dataset. Hence, due to its simplicity and strong performance, the quadratic regression model is recommended in applications.

It has been demonstrated in Table 3 and 4 that the regression model is characterised by a high R^2 score on different sets of seen and unseen data (at least 0.97, meaning that 97% of the variability in the dependent variable can be explained by the predictor (independent) variables). In an attempt to analyse the possible root causes of the remaining modelling errors and potentially enhance the model performance, additional features have been derived from the driving cycles used in the model development phase. In addition to the four selected predictor variables (see Table 4), the mean positive (μ_{a^+}) and negative (μ_{a^-}) accelerations, as well as their standard deviations ($\sigma_{a^+}, \sigma_{a^-}$) and the standard deviation of velocity (σ_v) are employed as influential variables related to vehicle dynamics. By using this extended set of predictor variables, an MLP neural network model with four hidden layers has been implemented.

Table 4 Comparative performance metrics of different machine learning algorithms using previously selected features

Number of features	Features/ Predictor variables	$RMSE_{tr}$ [kWH]	$RMSE_{val}$ [kWh]	R^2_{tr}	R^2_{val}
Quadratic Regression					
4	$\mu_{rg}, \mu_{rg}^2, \sigma_{rg}^2, \mu_v{}^*\bar{n}_{pass}$	0.9922	0.9922	0.9756	0.9756
LASSO Regression					
4	$\mu_{rg}, \mu_{rg}^2, \sigma_{rg}^2, \mu_v{}^*\bar{n}_{pass}$	0.9922	0.9922	0.9756	0.9756
RIDGE Regression					
4	$\mu_{rg}, \mu_{rg}^2, \sigma_{rg}^2, \mu_v{}^*\bar{n}_{pass}$	0.9922	0.9922	0.9756	0.9756
Decision Trees					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.0079	1.3208	1.0000	0.9558
Random Forest					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.3518	0.95	0.9969	0.9771
Gradient Boosting					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.9124	0.9399	0.9789	0.9776
K-nearest Neighbors					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.8868	0.9828	0.9801	0.9760
Support Vector Regression					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.9448	0.9477	0.9774	0.9772
MLP Neural Networks					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.9450	0.9473	0.9774	0.9772
1D Convolution Neural Networks					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.9581	0.9582	0.9767	0.9767

The corresponding modelling results shown in Table 5 indicate that the validation index R^2_{val} increases from 0.9772 to 0.9890. This reveals that (i) the limited performance of the models from Table 4 is more because of the limited set of features than the limited model structure, and (ii) the model with trip-based features can closely match the original, high-sampling-rate physical model, provided that the trip-based feature set is rich enough. However, despite the commendable performance, practical application of the model based on the additional, acceleration-based features is constrained by limited data availability. Namely, the typical bus tracking data are sampled too slowly to consistently capture the fast transients of vehicle acceleration signals. Hence, the quadratic regression model (3) remains to be recommended for application due to low data demands, simplicity, and still favourable accuracy (Table 5).

Table 5 Comparison of model performance with enhanced feature set

Number of features	Features/ Predictor variables	$RMSE_{tr}$ [kWh]	$RMSE_{val}$ [kWh]	R^2_{tr}	R^2_{val}
Quadratic Regression					
4	$\mu_{rg}, \mu_{rg}^2, \sigma_{rg}^2, \mu_v{}^*\bar{n}_{pass}$	0.9922	0.9922	0.9756	0.9756
MLP Neural Networks					
4	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$	0.9450	0.9473	0.9774	0.9772
9	$\mu_{rg}, \sigma_{rg}, \mu_v, \bar{n}_{pass}$, $\sigma_v, \mu_{a^+}, \mu_{a^-}, \sigma_{a^+}, \sigma_{a^-}$	0.6994	0.6996	0.9892	0.9890

3.6.2 HVAC system model

The HVAC power consumption regression model developed and indirectly experimentally validated within the physical e-bus model (Subsection 3.2) has a quadratic form gained by a feature selection method for four inputs: ambient temperature T_a, solar irradiation $\dot{Q}_{sol}$, ridership n_{pass}, and vehicle velocity v_{veh}:

$$P_{HVAC} = \beta_0 + \beta_1 T_a + \beta_2 \dot{Q}_{sol} + \beta_3 n_{pass} + \beta_4 v_{veh} + \beta_5 T_a v_{veh} + \beta_6 \dot{Q}_{sol}^2, \quad (3.12)$$

For integration into the trip-based data-driven model, the features of the HVAC model are averaged on a per-trip basis. This modification is justified by two assumptions: (i) the ambient conditions, such as solar irradiation and temperature, remain approximately constant during a relatively short bus trip, and (ii) the velocity and ridership variables, which may significantly change during the trip, are of

secondary influence on the HVAC consumption when compared to the influence of ambient condition variables. To further suppress the influence of velocity and ridership variations on mean value model accuracy, it is good to avoid the nonlinear terms present in model (3.12). It has been shown that this intervention does not considerably deteriorate the accuracy of physical model, but notably improves the accuracy of the mean value model, which is, thus, formulated as:

$$P_{HVAC} = \beta_0 + \beta_1 \bar{T}_a + \beta_2 \bar{\dot{Q}}_{sol} + \beta_3 \bar{n}_{pass} + \beta_4 \mu_v, \tag{3.13}$$

where the mean predictor variables are calculated over the trip, i.e., the driving cycle. The HVAC energy consumption per trip is then determined as:

$$E_{HVAC} = t_{trip} P_{HVAC}, \tag{3.14}$$

where t_{trip} is the trip duration.

The mean value HVAC model (3.14) has been tested against the original model (3.12) by using the five-fold cross-validation method illustrated in Fig. 18. The corresponding R^2_{val} value is 0.999 and an $RMSE_{val}$ is only 0.128 kWh. This confirms that the mean value HVAC system model can be used with a negligible loss of accuracy.

3.6.3 Analysis of model residuals

A practical analysis of the model residuals is carried out separately for powertrain and HVAC models, as well as for the full vehicle model. The analysis results relate to the validation dataset aggregated from validation folds in Fig. 18.

Powertrain model. An essential step in evaluating regression models involves examining the spread of residuals against the predicted values, which should be distributed around a horizontal zero-value line without forming any distinct patterns [10]. The residual plot of the powertrain quadratic regression model from Table 4 is shown in Fig. 23a. It indicates a slight slope of -0.015 kWh/kWh around the zero-value line, thus confirming the model consistency. Fig. 23b shows that the model predictions scatter closely around the ideal, identity line.

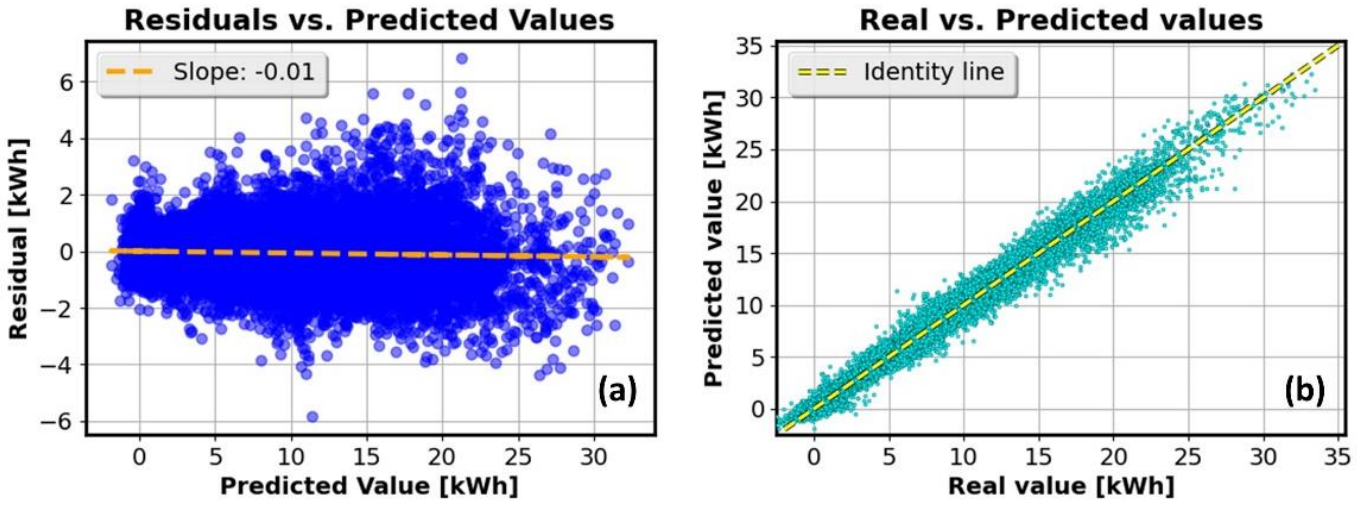


Fig. 23 Powertrain model residuals plotted vs. predicted values (a) and model predicted vs. true value plot (b)

The normality of residuals is another model assessment criterion. Fig. 24a demonstrates that, despite the p-value being lower than the normality threshold, the residuals exhibit an unbiased, symmetric distribution resembling the normal distribution. The distribution of relative residuals, shown in Fig. 24b, indicate that a great majority of relative residuals (actually 90% of them, see Table 6) fall below the margin of 8%. The Q-Q plot in Fig. 24c provides further illustration of the residual distribution normality by plotting the residuals in a manner that should form a straight line if they are normally distributed. Fig. 24d shows a heat plot of the residual versus true value. It reveals that the higher relative residuals are associated with lower predicted values, which is apparently due to the nature of relative residual calculation that tends to be more sensitive to smaller values. Table 6 provides a summarized residual statistic.

Table 6 Characterization of absolute and relative residual distributions of powertrain model

	Mean	Std.	1%	5%	10%	15%	25%	50%	75%	85%	90%	95%	99%
Absolute [kWh]	-0.06	0.85	-2.31	-1.43	-1.02	-0.78	-0.47	-0.06	0.31	0.62	0.87	1.35	2.56
Relative [%]	-0.50	6.93	-18.70	-11.51	-8.52	-6.75	-4.43	-0.45	3.46	5.87	7.65	10.72	17.83

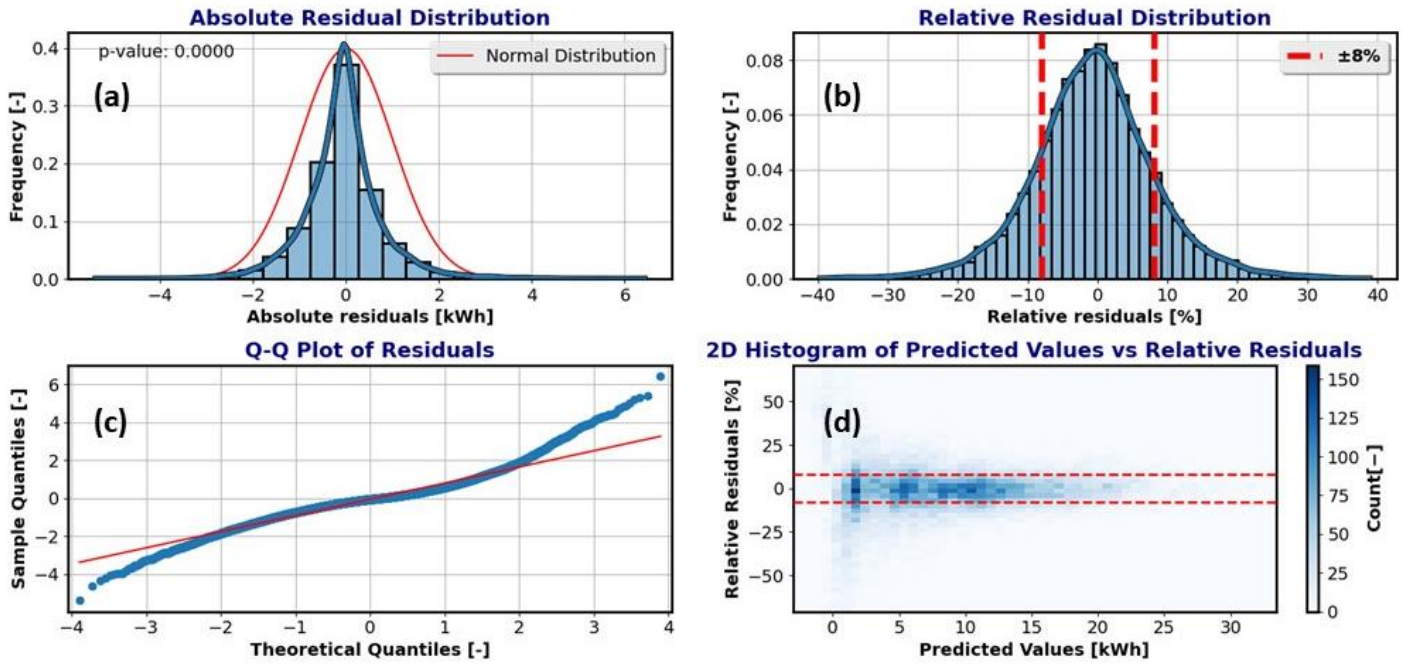


Fig. 24 Characteristic powertrain model residual plots

HVAC model. Fig. 25 shows the main residual plots of the HVAC model given by Eqs. (3.13) and (3.14), while the corresponding statistics is given in Table 7. The 90% of residuals fall below the absolute and relative margins of 0.16 kWh or 3.74%, respectively, which confirms the good modelling accuracy.

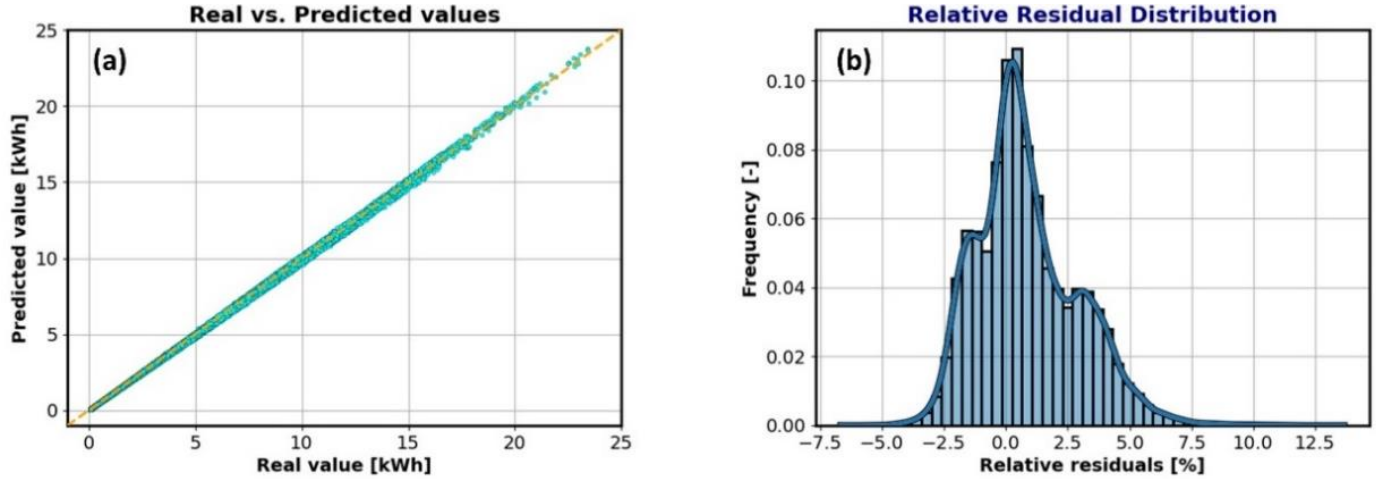


Fig. 25 HVAC model predicted vs. true value plot (a) and corresponding relative residual distribution plot (b)

Table 7 Characterization of absolute and relative residual distributions for HVAC model

	Mean	Std.	1%	5%	10%	15%	25%	50%	75%	85%	90%	95%	99%
Absolute [kWh]	0.02	0.11	-0.25	-0.16	-0.09	-0.05	-0.02	0.01	0.06	0.11	0.16	0.25	0.37
Relative [%]	0.85	2.05	-2.86	-2.05	-1.64	-1.29	-0.51	0.52	2.09	3.19	3.74	4.48	6.28

Full model. Fig. 26 shows the residual analysis results for the full e-bus model (both powertrain and HVAC models). The relative residual distribution is narrower than for the powertrain model itself (cf. Figs. 26b and 24b) due the accuracy contribution of the HVAC submodel. Consequently, the score R^2_{val} of the full model (when validated on the aggregate dataset) lifts from the powertrain model validation value of 0.9756 (Table 4) to 0.9812.

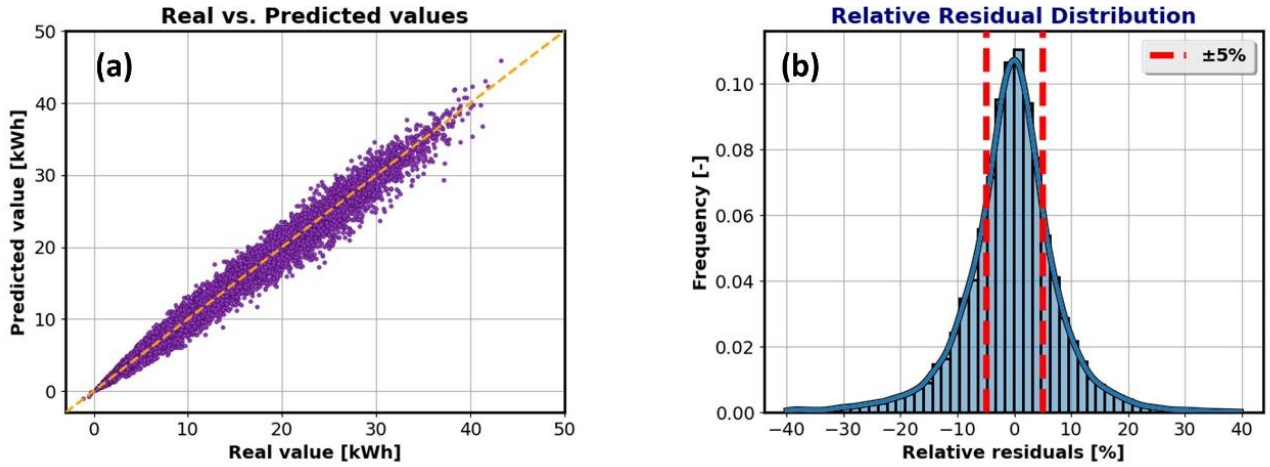


Fig. 26 Full e-bus model predicted vs. true value plot (a) and corresponding relative residuals distribution plot (b)

3.7 Short conclusion

A data-driven regression model for predicting the electric city bus battery energy consumption has been built up. The model has been parameterized and validated based on a comprehensive data set obtained by simulating an experimentally validated physical model over a wide set of naturalistic city bus driving cycles. The model relies on typically available trip-related data, as opposed to the physical model that requires high sampling rate driving cycle data. It consists of independent powertrain and HVAC submodels. For the powertrain, a feature selection method has been used to find an optimal quadratic regression model for the specific energy consumption (in kWh/km), where the selected features include the mean road grade and its square, the road grade standard deviation square, and

the product of mean velocity and ridership. The model performance (characterized by the validation R^2 value of 0.976) is comparable with more complex methods such as neural networks and gradient boosting, but with the added advantage of greater simplicity and robustness.

The original HVAC system model with four inputs (ambient temperature, solar irradiance, vehicle velocity, and ridership) has been reformulated to have (i) a mean value form to be applicable to trip-based inputs and (ii) a linear structure to suppress the influence of velocity and ridership variation on the mean value modelling accuracy. When validating the full model on an aggregate dataset, it registered a notable R^2 score of 0.981, thus confirming its capability to accurately describe the energy consumption patterns.

The developed e-bus model provides a solid basis for accurate and computationally efficient description and simulation of city bus fleets for electrification planning purposes. The presented approach of modelling the 12m fully electric city bus can be applied to other bus sizes (e.g., 18m) and types of city buses (e.g., HEV, PHEV, and H2 buses).

4 Charging configuration optimization

4.1 Introduction

An integral part of the transport system electrification planning is optimization of charging configuration. When concerning an e-bus transport system, charging configuration optimization aims at selecting terminals to serve as fast charging stations and selecting the number of chargers per each terminal to minimize investment cost and service delay.

To this end, this section 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. The charging terminals are selected for (i) the predefined e-bus fleet defined by the number of buses, bus type, and battery capacity and (ii) predefined bus lines, schedules, and timetables. The optimization objectives to be minimized include the number of charging terminals, the total number of chargers, and the total (cumulative) service delay with respect to timetables. The optimization is based on the pilOPT multi-objective genetic algorithm (GA) provided in the modeFRONTIER optimization environment, which is connected with a transport system macro-simulation model implemented in programming language Python. To reduce the input space represented by the number of chargers on different terminals, and thus improve convergence properties of the GA, the modified greedy set-cover algorithm is developed and used in pre-optimization. To illustrate the effectiveness of the proposed method/tool, the optimized charging configurations are compared with near-optimal charging configurations previously found through expert knowledge.

The charging configuration optimization tool is demonstrated on a real city bus transport system corresponding to 29 lines/routes, 25 terminals, and 303 buses operating in the city of Jerusalem [7]. The transport system macro-simulation model is built based on real-life travelling time and terminal dwell time data for every route and direction on an hourly basis throughout the operating day. These data have been retrieved from the GPS tracking data collected on the existing (Diesel) bus fleet operating on the given routes. The transport system model also includes the energy consumption maps for e-buses, which have been determined by simulating the physical e-bus model from Section 3 over the high-resolution driving cycle data. 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 system energy consumption, respectively.

The remaining part of this section is structured as follows. Subsection 4.2 provides an overview of the optimization framework. Subsection 4.3 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 Subsection 4.4. Subsection 4.5 presents the overall charging system configuration optimization algorithm, with the results given and discussed in Subsection 4.6. Subsection 4.7 presents concluding remarks.

Note: The work presented in this section has been disseminated through the following conference paper, which also includes a methodology state-of-the-art review and elaborates on the contributions of the approach proposed:

D. Matković, J. Topić, B. Škugor, and J. Deur, "Search Space Reduction-Supported Multi-objective Optimization of Charging System Configuration for Electrified City Bus Transport System", 17th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Paphos, Cyprus, 2022.

4.2 Optimization framework

This subsection provides an overview of the optimization framework shown in Fig. 27 and aimed to determine the optimal charging configuration for an e-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, the design of experiments (DOE) is defined, which sets the initial charging configuration and is generated by pilOPT algorithm. The number of chargers at each 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 a bus route (or line) $r \in [1, N_r]$, where N_r is the number of routes. There are three objectives to be minimized, which are denoted by J_i, $i \in [1, 3]$ (see Fig. 27), 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 constraints, where N_c and N_{ct} are calculated directly from the optimized charging configuration, and D_{tot} is obtained from macro-simulation. The charging configuration is represented by the set [Ch_1,..., Ch_n], i.e. by the number of installed chargers at every terminal. The macro-simulation model parameters include the bus schedule set, S, a deadzone time, T_{dz}, the charging power, P_{ch}, and the e-bus battery energy 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.

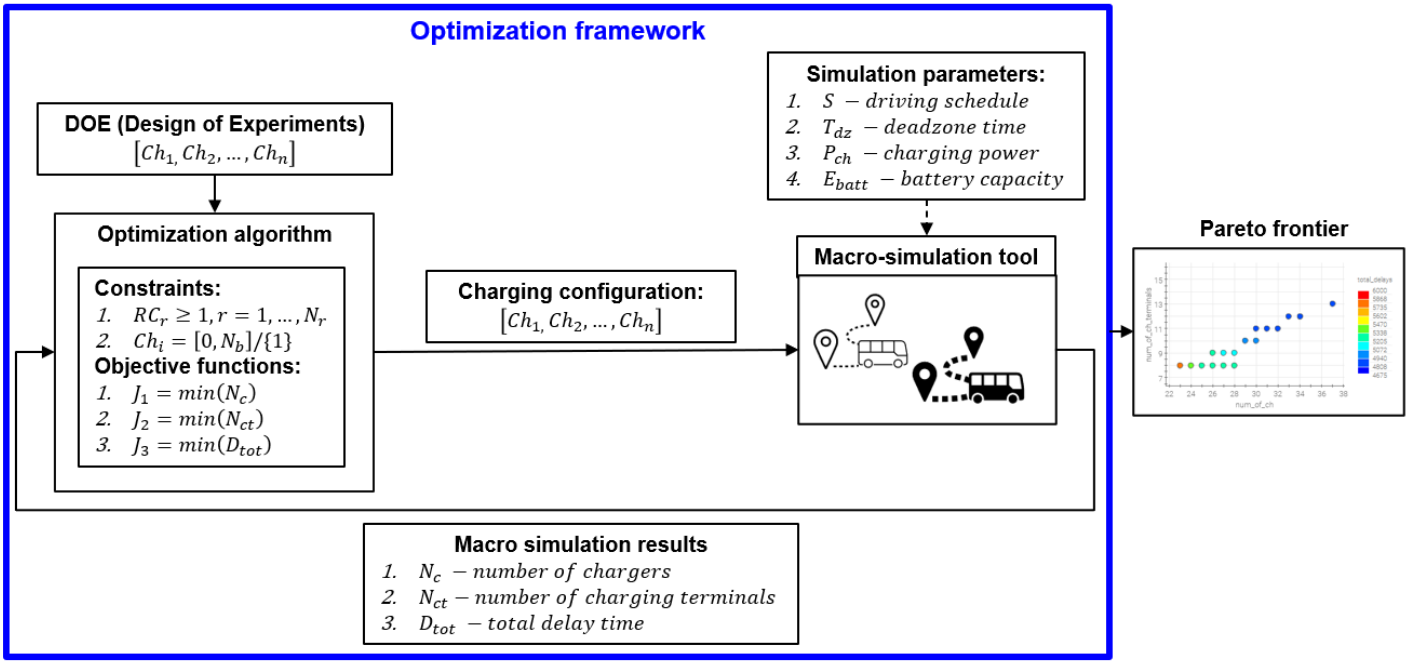


Fig. 27 Block diagram of optimization framework used for optimizing charging configuration

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 to minimize the objective functions including 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 waits, subject to optimization constraints.

4.3 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 step 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 operates only on one of the routes. Simulation outputs are post-processed to obtain detailed transport analysis data for every route and bus, including the dwell time at each terminal, the delay time of driving missions, and a variety of metrics regarding the battery state of charge (SoC; actually a state of energy), energy charged, and bus utilization.

Fig. 28 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 (service trips). A trip 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 ≥ 20% at the charging terminal, the departure is postponed, i.e. a delay occurs. The trip travel time and the energy consumption are obtained from the corresponding maps, which have been determined (i.e., pre-processed) from the recorded driving cycle data and the e-bus physical micro-simulation, and they are stored in a database over different routes and on an hourly basis. The bus battery SoC is updated at the end of a trip 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. the time of the day). After the bus arrives at the terminal, a simple heuristic charging management algorithm is executed.

The charging management algorithm is described by the flowchart shown in Fig. 29. First, when a 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 bus battery is fully charged ($SoC = 95\%$) or a bus with SoC ≥ 20% is scheduled to depart, the bus disconnects from the charger, and the bus with the lowest SoC connects.

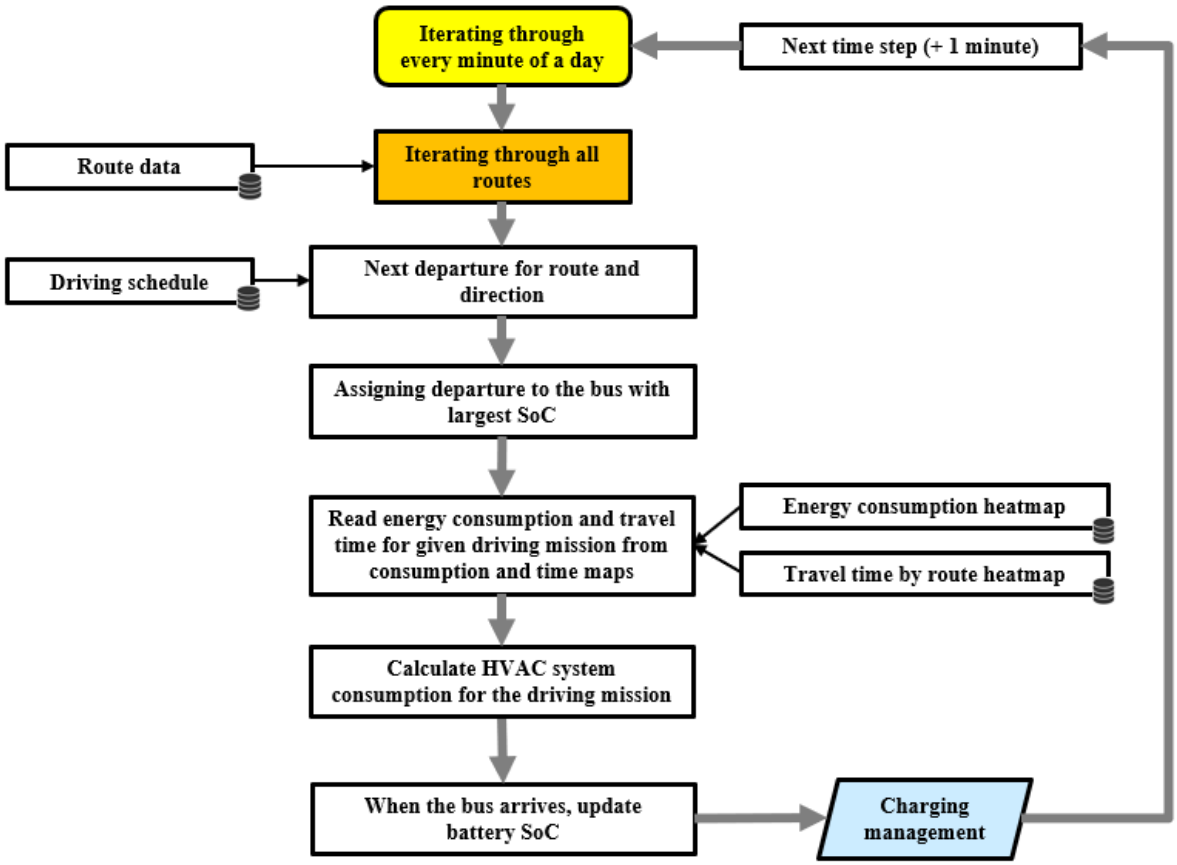


Fig. 28 Flowchart of macro-simulation model

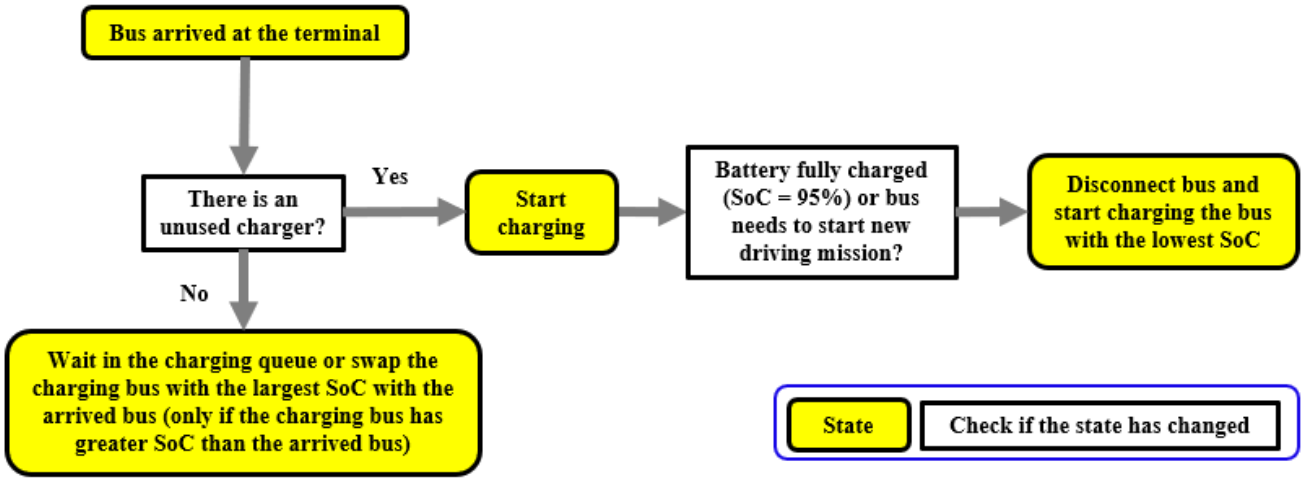


Fig. 29 Flowchart of heuristic charging management algorithm

4.4 "Greedy algorithm" based optimization of charging locations

The e-bus transport system includes 25 terminals, which may or may not include chargers (two options), which results in 2^{25} = 33,554,432 possible charging configuration combinations. Thus, it would be very time-consuming to manually find viable charging configuration combinations to be used in full optimization in Subsection 4.5, while considering the 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 subsection. The search space is reduced by decreasing the number of input variables, in this case the charging terminal candidates, otherwise set to all terminals.

4.4.1 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.

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.

4.4.2 Modified greedy set-cover algorithm for charging candidate optimization

The mathematical formulation of the set-cover problem is as follows:

$$\begin{aligned} &\textit{Given the elements of } U = \{u_1, u_2, \dots, u_n\}, \\ &\textit{Subsets } S_1, S_2, \dots, S_k \subseteq U, \\ &\textit{Weights } w_1, w_2, \dots, w_k, \\ &\textit{find } I \subseteq \{1, 2, \dots, k\}, \\ &\textit{that } \min \textstyle\sum_{i \in I} w_i, \text{s.t.} \bigcup_{i \in I} S_i = U. \end{aligned} \tag{4.1}$$

(4.2)

The greedy set-cover algorithm is shown in Algorithm 1 below and is 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: $C = \emptyset$
while $C \neq U$ **do**
  $D \leftarrow U \backslash C$;
  for $i \leftarrow 1$ *to* n **do**
    $c_i \leftarrow \frac{|S_i \cap D|}{w_i}$;
  end
  $S_i* \leftarrow \max(c_1, ..., c_n)$;
  $C \leftarrow C \cup S_i*$;
end
return C

The proposed, modified greedy set-cover algorithm is a version of the original algorithm, where the 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, as given by Eqs. (4.1) and (4.2), 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 $w_1, w_2, \dots, w_k$, (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: $C = \emptyset$
$i \leftarrow 0$;
$cs_{min} \leftarrow n$
$C_{min} = \emptyset$;
while $i < 20$ **do**
  $w_i, ..., w_n \leftarrow \mathcal{N}(\mu, \sigma^2)$;
  $I = \emptyset$;
  while $I \neq U$ **do**
    $D \leftarrow U \backslash I$;
    for $i \leftarrow 1$ *to* n **do**
      $c_i \leftarrow \frac{|S_i \cap D|}{w_i}$;
    end
    $S_i* \leftarrow \max(c_1, ..., c_n)$;
    $I \leftarrow I \cup S_i*$;
  end
  if $I \notin C$ **then**
    $C \leftarrow [C, I]$;
    $cs_{min} = min(cs_{min}, |I|)$;
  end
end
for $I \in C$ **do**
  if $|I| = cs_{min}$ **then**
    $C_{min} \leftarrow [C_{min}, C]$;
  end
end
return C_{min}

4.4.3 Charging candidate optimization results

Fig. 30 shows the assignment of terminals to the routes of considered city bus system. The goal is to find the minimum charging configuration combinations satisfying the route coverage constraint meaning that each route is covered by at least one charging terminal (Subsection 4.2).

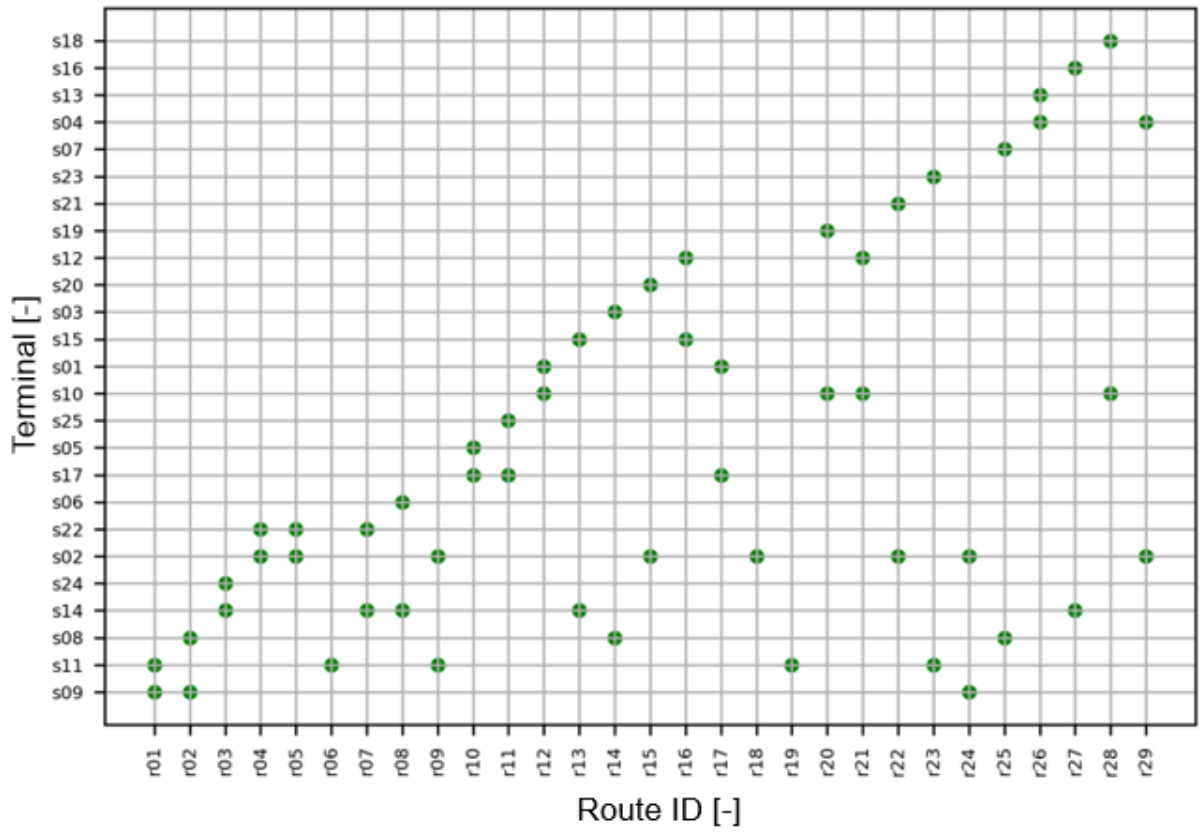


Fig. 30 Route and belonging terminals of the considered city bus system

Table 8 shows the combinations of minimum charging 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 was succeeded to be found manually, while the modified greedy set-cover algorithm has managed to find four distinct combinations, including the manually found one. The reduced input space-based optimizations (Subsection 4.5) consider the union of charging terminals determined by the greedy set-cover algorithm (marked green in Table 8). That said, the number of input variables (set by default to $[Ch_1,...,Ch_n]$ in Fig. 27), i.e. terminals decreases from n = 25 to 10, which is a significant improvement in terms of search space reduction.

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.

Table 8 Charging candidate optimization results obtained by expert knowledge (i.e, 'manually') and application of modified greedy set-cover algorithm

Configuration \ Terminal	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

4.5 Optimization of overall charging system configuration

This subsection presents details of the overall, multi-objective optimization framework built around modeFRONTIER genetic algorithm pilOPT. Fig. 31 shows the modeFRONTIER optimization scheme, which includes inputs that represent charging configuration (marked green), and outputs that are used in constraints and objective functions evaluation (marked red). The next subsections explain in detail each component of the optimization scheme.

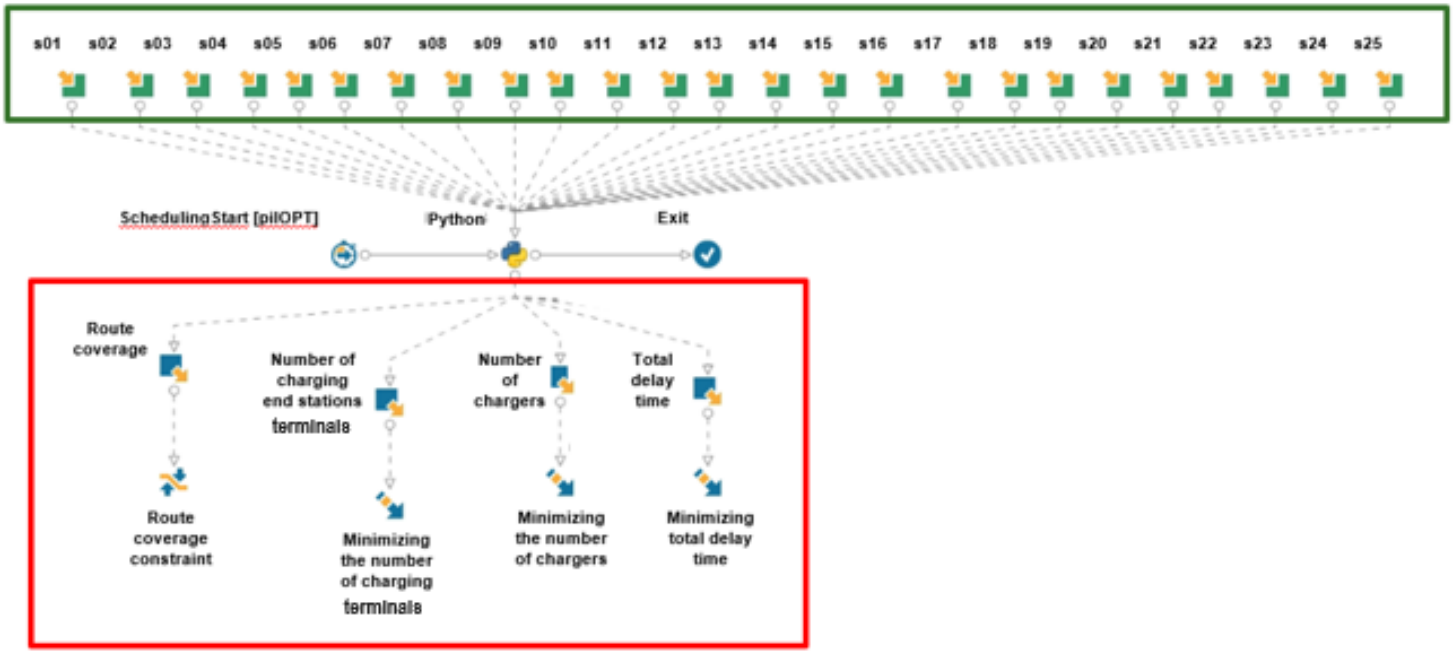


Fig. 31 modeFRONTIER scheme of overall charging configuration optimization

4.5.1 Objective functions

As outlined in Subsection 4.2, 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}):

$$\min N_{cs} \tag{4.3}$$

$$\min N_{ch} \tag{4.4}$$

$$\min D_{tot} \tag{4.5}$$

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 (Subsection 4.3).

4.5.2 Optimization problem constraints

As discussed in Subsection 4.2, the optimization constraints are formulated as:

$$Ch_i = [0, N_b]/\{1\},\ i = 1,\ldots,n \tag{4.6}$$

$$RC_r \geq 1,\ r = 1,\ldots,N_r \tag{4.7}$$

The constraint (4.6) 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 (4.7) 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.

4.5.3 Optimization scenarios

Table 9 overviews the scenarios for which the optimization and related analyses will be carried out 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 operational 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 a reduced input

space, where the number of charging terminal candidates is reduced from 25 to 10, as discussed in Subsection 4.4. 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 20,000.

Table 9 Overview of charging configuration optimization scenarios

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

4.5.4 Comparative analysis

Table 10 shows which charging configuration combinations from Table 8 are found in which optimization scenario from Table 9. The labels Feasible and Pareto optimal designate whether the solution is feasible (in terms of satisfying the constraints) and 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.

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 8, *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 Fig. 32 indicates that the terminals *s04* and *s12* indeed 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*.

Table 10 Overview of the charging configuration combinations found in each scenario, both according to all feasible and Pareto solutions

Charging configuration combination / Scenario	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	+ / -	+ / -	+ / -	+ / +

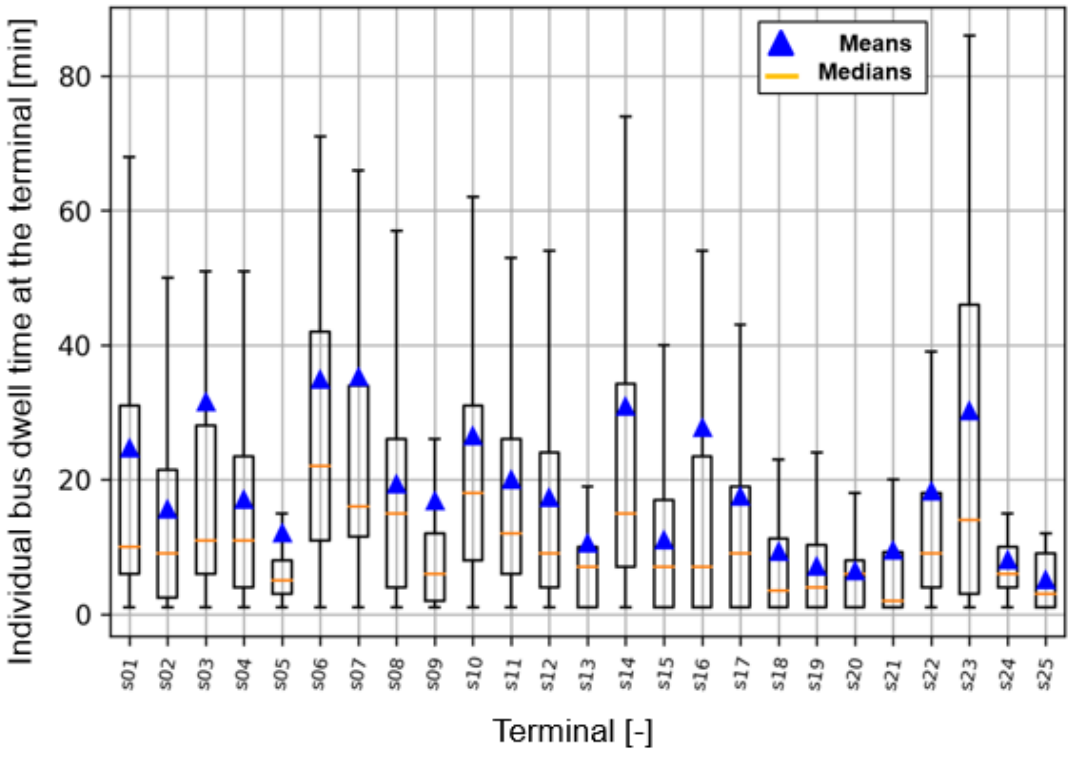


Fig. 32 Terminal dwell time statistics

4.6 Results and discussion

In this subsection, the results for optimization scenarios defined in Table 9 are presented and discussed. First, optimization results are given, and they are then supplemented by detailed macro-simulation results.

4.6.1 Optimization results

The first scenario from Table 9 is the “Autonomous complete space” scenario, which concerns the complete (unreduced) input space and autonomous mode of pilOPT algorithm. The optimization resulted in 11,723 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 Fig. 33a. This solution agrees with the results presented in Subsection 4.4 and Table 8, i.e. the full optimization finds the same minimum number of charging terminals as greedy algorithm did. The Pareto frontier in Fig. 33a suggests that the total transport system delay time, as the third objective, can be reduced (blue tones) if the number of charging terminals and/or the number of chargers is increased.

When reducing the input space (Fig. 33b, Scenario 2), 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 corresponding number of iterations is also significantly lower (4,475 vs. 11,723), 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. 11,723; Scenario 3), the Pareto frontier shown in Fig. 33c 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 20,000, the optimization results in the Pareto frontier shown in Fig. 33d (Scenario 4). 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 as impractical, and it may be concluded that the previous scenario (Scenario 3) could not be further improved. Its

characteristic charging configurations marked by black circles in Fig. 33c will be analysed in detail in Subsection 4.6.3.

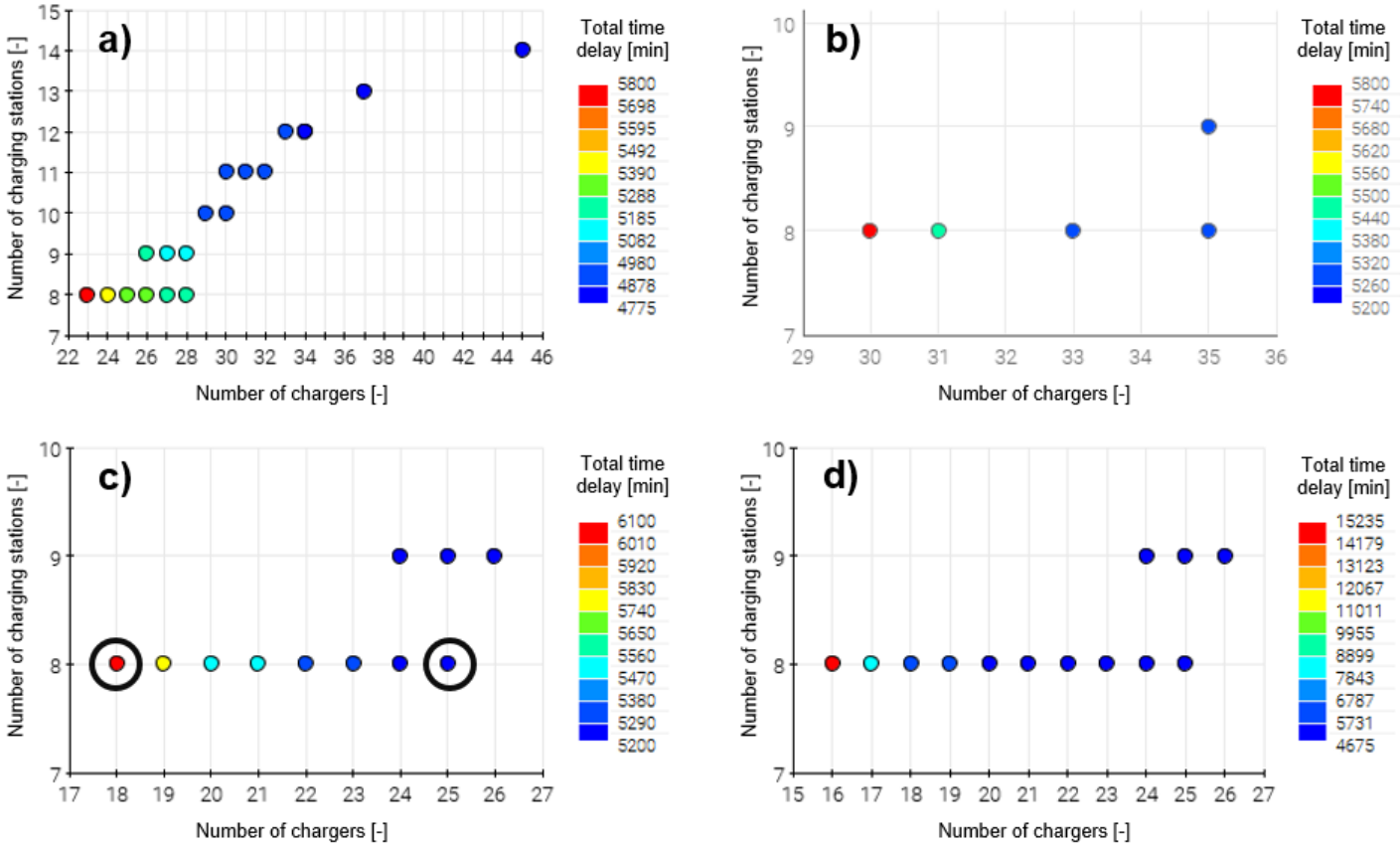


Fig. 33 Pareto frontiers obtained for different optimization scenarios from Table 10: a) Autonomous complete space, b) Autonomous reduced space, c) Self-initialized reduced space, and d) Self-initialized reduced space II

4.6.2 Optimization procedure

Based on the results from the previous subsection, this subsection formalises the optimization steps, as shown in Fig. 34 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 Subsection 4.4, **(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.

4.6.3 Simulation outcomes for Pareto optimal solutions

The optimal configurations obtained in Subsection 4.6.1 (see Fig. 33c) based on the procedure summarized in Subsection 4.6.2 (i.e., Fig. 34) 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 8). 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.).

The macro-simulation results presented in Fig. 35 contain 6 plots. The first plot shows the e-bus SoC values at the end of the day (i.e., the final SoC, SoC_f) 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.

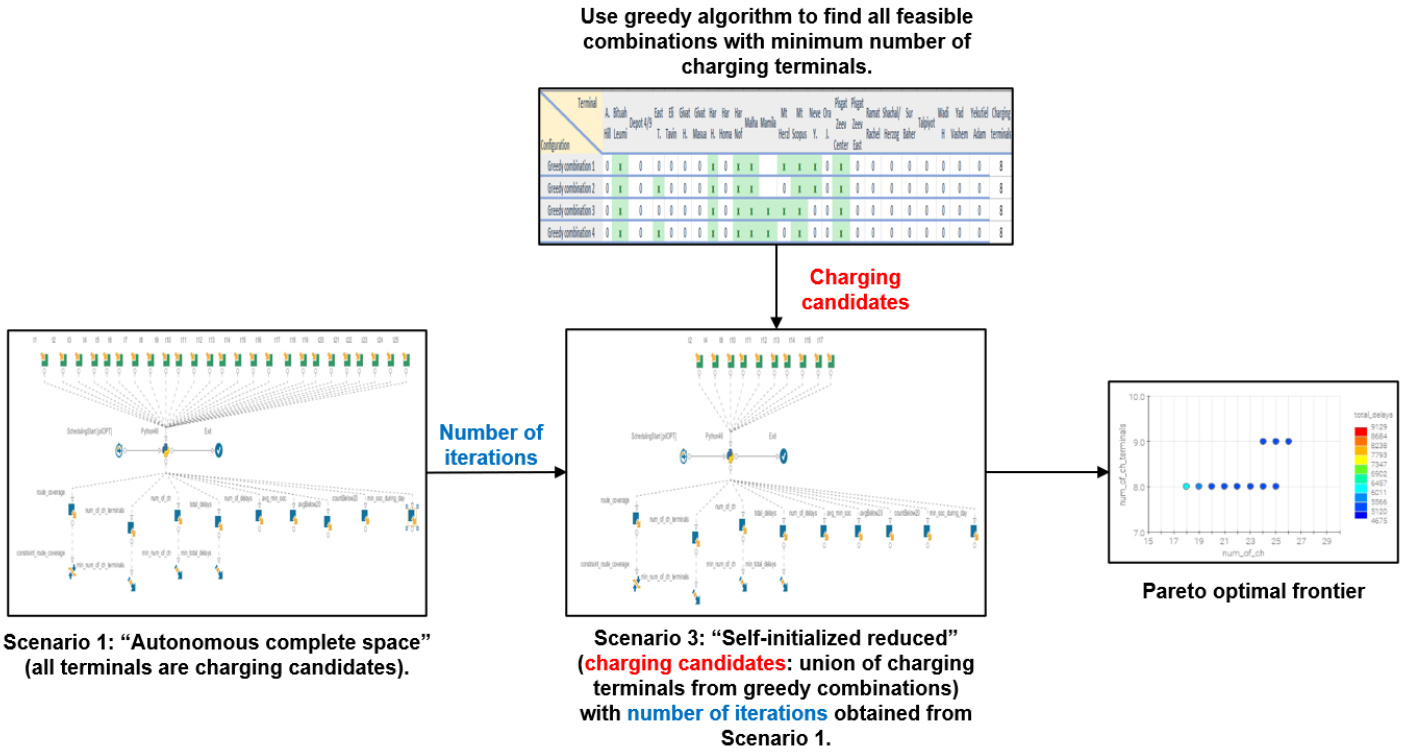


Fig. 34 Optimal charging configuration optimization setup

As discussed in Subsection 4.6.1, the Pareto frontier gathered from the "Self-initialized reduced space" scenario (Scenario 3), shown in Fig. 33c, resulted in configurations that had [18, 25] chargers distributed on 8 charging terminals, which corresponded to the minimum number of charging terminals when satisfying the route coverage constraint. For the sake of simplicity, only configurations with upper and lower bands of the number of chargers (designated by circles in Fig. 33c) have been simulated and are discussed below. Fig. 35 presents the macro-simulation results for the lower-band configuration with 18 charging terminals.

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; e.g., 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.

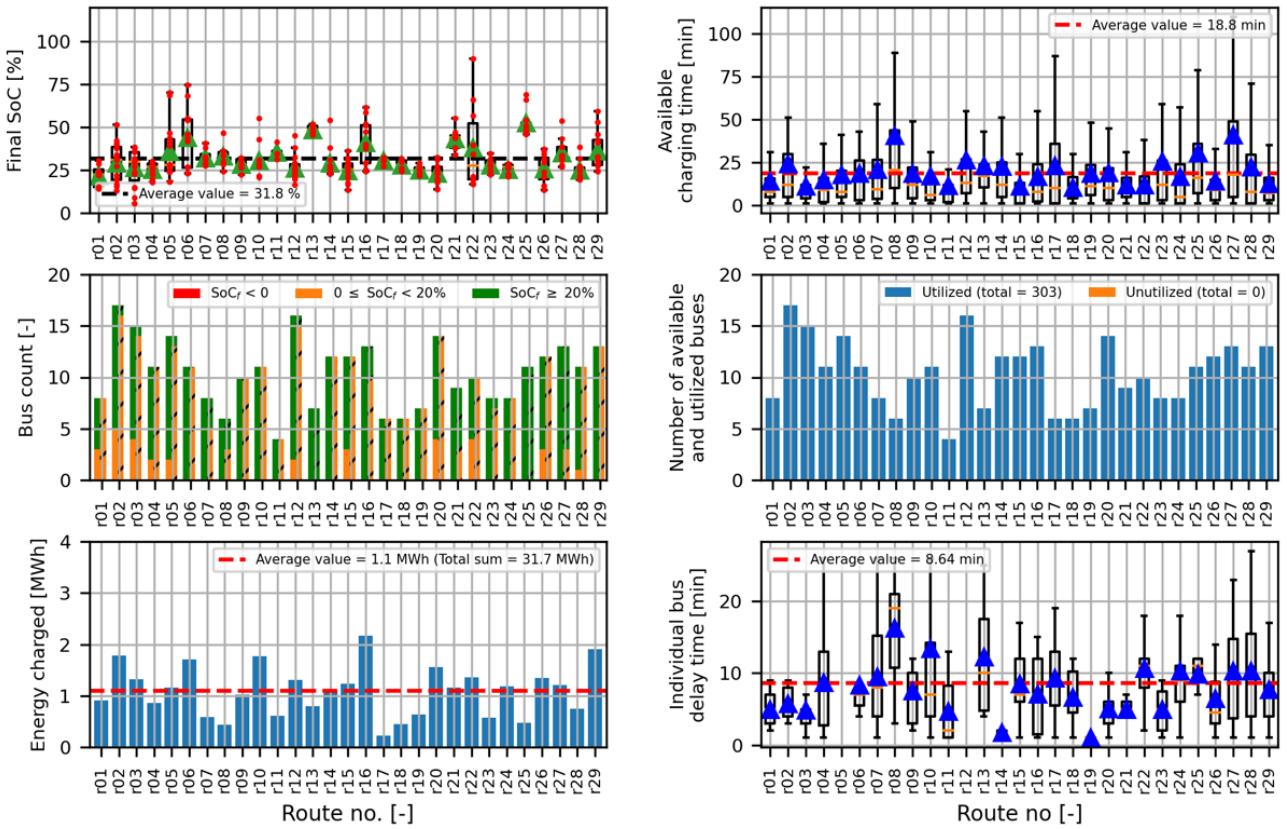


Fig. 35 Macro-simulation results for optimal charging configuration related to 8 charging terminals and 18 chargers (see left-hand side circle in Fig. 33c)

Fig. 36 shows macro-simulation results related to optimal charging configuration with 25 chargers (see upper-band circle in Fig. 33c). 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 (Fig. 36).

The configurations found through expert knowledge (i.e., "manually") are listed in Table 11 based on Table 8 and variation of total number of chargers. Table 11 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 Fig. 35.

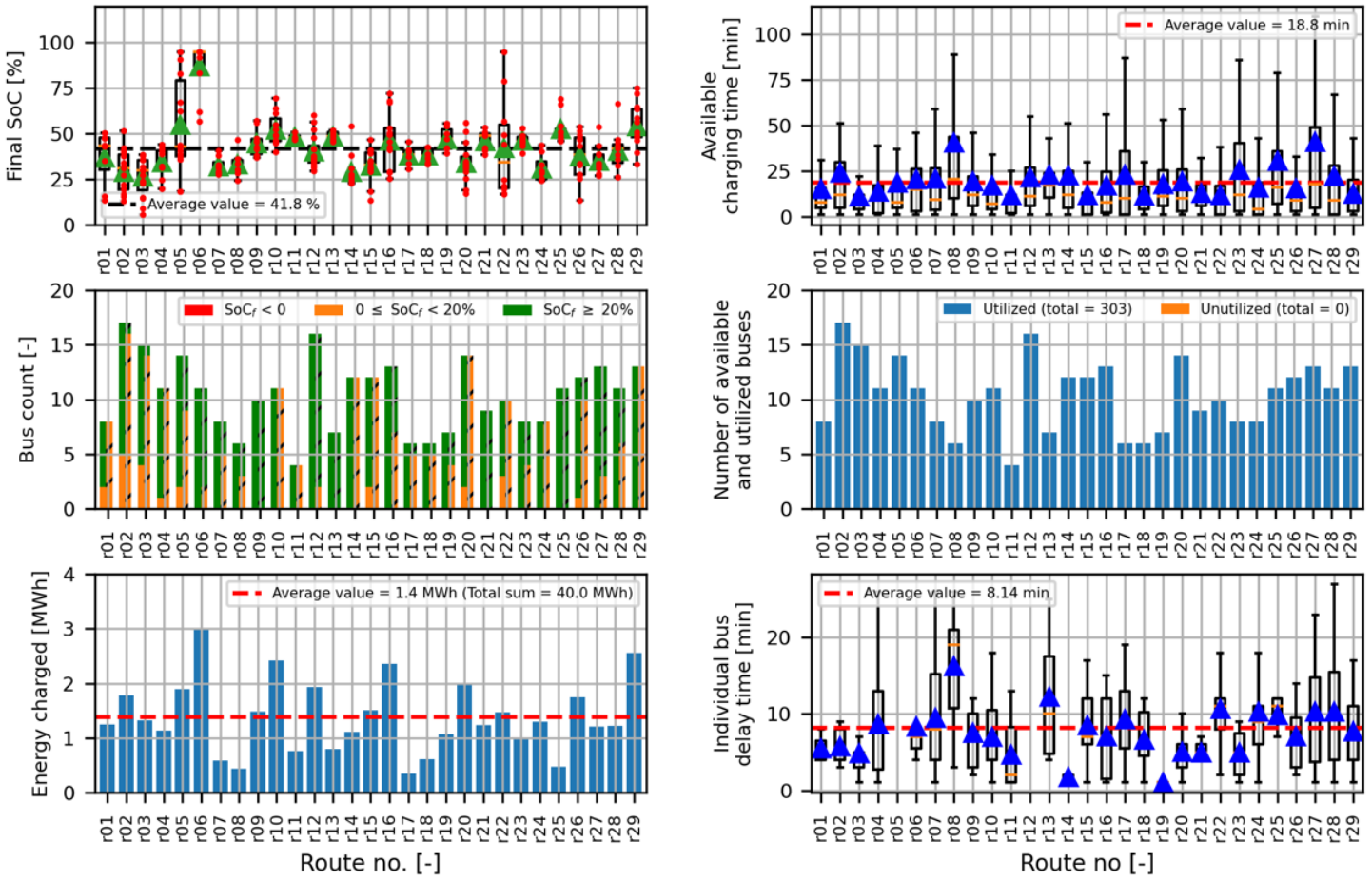


Fig. 36 Macro-simulation results for optimal charging configuration related to 8 charging terminals and 25 chargers (see right-hand side circle in Fig. 33c)

Table 11 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 12 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 and delay statistics.

Table 12 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

4.7 Short 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 Fig. 34, 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, **(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), and **(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 caused by charging restrictions.

The selected Pareto optimal charging configurations have been compared with the ones found based on expert knowledge. It has been demonstrated 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.

5 Optimal charging management

5.1 Introduction

The proposed hierarchical EV fleet charging management is conceived to optimize the charging power time profiles at two levels: (i) aggregate level, and (ii) distributed level of individual EVs (see illustration in Fig. 37). The charging power profile on the aggregate level is meant to be optimized in a receding horizon manner (a model predictive control approach, MPC) by using a simplified and numerically efficient aggregate battery-based EV fleet model. The obtained optimal aggregate charging power is then distributed over individual EVs in each time step by using a heuristic algorithm based on charging priorities. The main advantage of this hierarchical approach is in simplicity of implementation and excellent scalability to relatively large EV fleets (e.g., e-hubs; i.e., its computational complexity is invariant to the number of EVs within a fleet).

The optimization on the aggregate level assumes the availability of predictions of different quantities over a prediction horizon, such as arrival time of EVs (i.e., starting time of their charging) and related battery state-of-charge (SoC), electricity price, power production from renewable energy sources (RES), electricity consumption of other consumers within a local micro-grid, and similar. For instance, in the case of e-bus fleets, the arrival time can be predicted from known driving schedules and

historical and actual traffic data. Similarly, historical data and external condition prediction (e.g., weather forecast) can be used in connection with machine learning techniques to provide internal and RES electricity productions, as well as electricity prices.

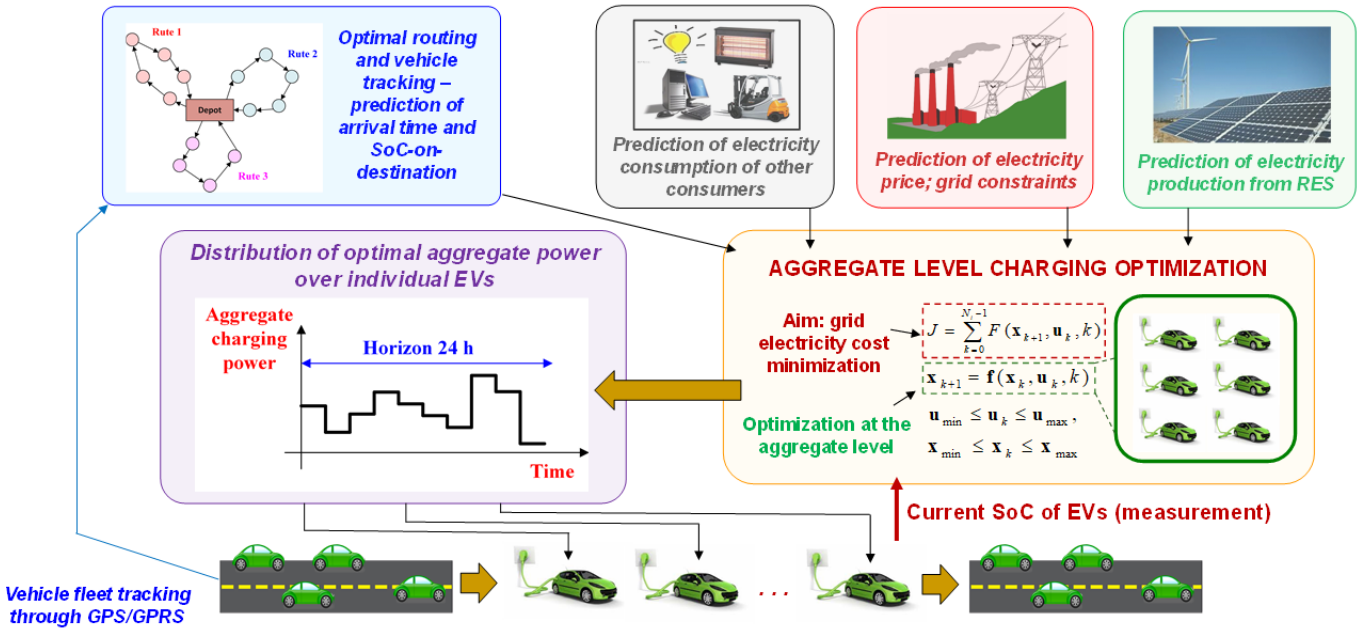


Fig. 37 Concept of hierarchical EV fleet charging management framework

5.2 EV fleet models

Two types of EV fleet models are formulated and used [12]: (i) aggregate, and (ii) distributed; where the first one considers all EVs within fleet as a single aggregated battery, while the second one models each EV battery separately. The batteries are modelled as energy storages with the state-of-energy (SoE) and the charging power as their state and control variables, respectively.

5.2.1 Aggregate EV fleet model

Dynamics of the aggregate EV fleet model is described by the following state equation [12]:

$$SoE_{agg}(k+1) = SoE_{agg}(k) + SoE_{in,avg}(k)\frac{n_{in}(k)}{N_v} - SoE_{out,avg}(k)\frac{n_{out}(k)}{N_v} + \eta_{ch}\frac{P_{c,agg}(k)\Delta T}{N_v E_{max,ind}}, \quad k = 0,1,\dots,N_t-1, \tag{5.1}$$

where k is the discrete time step, N_t is the total number of time steps, $SoE_{in,avg}$ and $SoE_{out,avg}$ are average SoE values of EVs connecting to the grid and disconnecting from the grid within k^{th} step, respectively, with the corresponding number of EVs denoted by n_{in} and n_{out}, respectively, N_v is the

total number of EVs within the fleet, $P_{c,agg}$ is the aggregate charging power, $E_{max,ind}$ is the energy capacity of the individual battery (expressed in Wh; $N_v E_{max,ind}$ is the energy capacity of all batteries within the fleet), and ΔT is the time discretization (expressed in hours, [h]). The aggregate SoE state variable SoE_{agg} is defined as the normalized average energy of connected EVs:

$$SoE_{agg}(k) = \frac{\sum_{i=1}^{N_v} E_{c,i}(k)}{N_v E_{max,ind}}, \tag{5.2}$$

where $E_{c,i}$ is the battery energy (in Wh) of i^{th} EV, which equals the actual battery energy if EV is connected within the k^{th} time step, while it is zero, otherwise. The lower limit on SoE_{agg} is zero, while the upper limit is set to be dependent on the number of EVs connected to the grid (n_c):

$$0 \leq SoE_{agg}(k) \leq \frac{n_c(k)}{N_v} \leq 1. \tag{5.3}$$

The aggregate charging power is limited in the range from zero (only the one-direction power flow is enabled, i.e., from a grid to EVs) to the charging power capacity of connected EVs:

$$0 \leq P_{c,agg}(k) \leq n_c(k) P_{cmax,ind}, \tag{5.4a}$$

where $P_{cmax,ind}$ is the maximum charging power of individual EV. Additionally, the aggregate charging power is limited by the fixed upper constraint:

$$P_{c,agg}(k) \leq P_{c,agg,\max}, \tag{5.4b}$$

to account for the grid power limit.

5.2.2 Distributed EV fleet model for offline charging power optimization

The model structure given by the state equation (5.1) may also be used to model the individual (i^{th}) EV battery within a distributed EV fleet model as:

$$SoE_i(k+1) = SoE_i(k) + SoE_{in,i}(k) n_{in,i}(k) - SoE_{out,i}(k) n_{out,i}(k) + \eta_{ch} \frac{P_{c,i}(k) \Delta T}{E_{max,ind}}, \tag{5.5}$$

with related constraints:

$$0 \leq SoE_i(k) \leq n_{cb,i}(k), \quad n_{cb,i} \in \{0, 1\}, \tag{5.6}$$

$$0 \leq P_{c,i}(k) \leq n_{cs,i}(k) P_{cmax,ind}, \quad n_{cs,i} \in [0, 1], \tag{5.7}$$

where N_v from Eq. (5.1) is now set to 1 and thus omitted in Eq. (5.5), $SoE_{in,i}$ and $SoE_{out,i}$ are SoE values of the i^{th} EV when it connects to and disconnects from the grid, respectively, and $n_{in,i}$ and $n_{out,i}$ are binary variables taking the value of 1 if connection/disconnection of i^{th} EV takes place within the k^{th} step, and 0, otherwise. The state variable SoE_i is defined similarly to the definition of SoE_{agg} in (5.2): $SoE_i(k) = E_{c,i}(k)/E_{max,ind}$, where $E_{c,i}$ equals zero if i^{th} EV is disconnected. The variable n_c from Eqs. (5.3) and (5.4) is replaced by $n_{cb,i}(k)$ in Eq. (5.6) and by $n_{cs,i}$ in Eq. (5.7), where $n_{cb,i}$ represents the binary variable taking the value of 1 if the i^{th} EV is partially or fully connected within k^{th} step, and 0, otherwise, while $n_{cs,i}(k)$ represents a share of EV connection time within the k^{th} step (e.g., $n_{cs,i} = 0.1$ means that a related EV was connected 10% of time step duration ΔT).

5.2.3 Distributed EV fleet model for simulation study

To strictly satisfy the lower SoE constraint in Eq. (5.6), $0 \leq SoE_i(k)$, within the EV fleet simulation model, the state equation of distributed model (5.5) is modified as:

$$SoE_i(k+1) = \begin{cases} SoE_{int,i}(k), & \text{for } n_{out,i}(k) = 0, \\ 0, & \text{for } n_{out,i}(k) = 1, \end{cases} \tag{5.8a}$$

where SoE_i at k+1 step takes an intermediate value $SoE_{int,i}$ if the EV is not disconnected at the k^{th} step, while it equals 0, otherwise. The intermediate SoE value incorporates the SoE contributions brought by EV connection to the grid ($SoE_{in,i}$) and charging with the power $P_{c,i}$ (cf. Eq. (5.5)):

$$SoE_{int,i}(k) = SoE_i(k) + SoE_{in,i}(k) n_{in,i}(k) + \eta_{ch} \frac{P_{c,i}(k)\Delta T}{E_{max,ind}}. \tag{5.8b}$$

The SoE on departure, $SoE_{out,i}$, is updated in the (k+1)th step to $SoE_{int,i}(k)$ only if a new driving mission starts at the k^{th} step ($n_{out,i}(k) = 1$):

$$SoE_{out,i}(k+1) = \begin{cases} SoE_{out,i}(k), & \text{for } n_{out,i}(k) = 0, \\ SoE_{int,i}(k), & \text{for } n_{out,i}(k) = 1. \end{cases} \tag{5.9}$$

On the other hand, the SoE of an EV arriving from a driving mission and connecting to the grid in the k^{th} step, $SoE_{in,i}$, i.e. when $n_{in,i}(k) = 1$ holds, is calculated as a function of the SoE at previous departure SoE (i.e., $SoE_{out,i}(k)$) and a travelled distance $d_i(k)$ of that driving mission:

$$SoE_{in,i}(k) = \begin{cases} 0, & \text{for } n_{in,i}(k) = 0, \\ f_{SoE}\left(SoE_{out,i}(k), d_i(k)\right), & \text{for } n_{in,i}(k) = 1. \end{cases} \tag{5.10}$$

The upper constraints on individual charging powers are set to:

$$P_{c,\max,i}(k) = \min\left(n_{cs,i}(k)P_{c\max,ind}, \frac{1 - SoE_i(k) - SoE_{in,i}(k)n_{in,i}(k)}{\eta_{ch}\Delta T}E_{\max,ind}\right), \tag{5.11}$$

where the first term within the operator min(.) corresponds to the upper constraint of Eq. (5.7), while the second one is to satisfy the upper SoE limit from Eq. (5.6) based on Eq. (5.8b) with $SoE_{int,i}$ limited to 1 (recall that the lower SoE limit from (5.6) is ensured through the modified state equation (5.8)).

For the purpose of post-analysis, the SoE and charging power values of individual EVs from the distributed model can be aggregated for each time step *k* as:

$$SoE_{agg}(k) = \sum_{i=1}^{N_v} SoE_i(k) \Big/ N_v, \tag{5.12}$$

$$P_{c,agg}(k) = \sum_{i=1}^{N_v} P_{c,i}(k). \tag{5.13}$$

The charging power can be supplied from the grid (P_g) or from the local renewable energy sources (RES; P_{res}), with the priority of charging being given to RES while covering the eventual power deficit from the grid:

$$P_g(k) = \begin{cases} P_{c,agg}(k) - P_{res}(k), & \text{for } P_{c,agg}(k) - P_{res}(k) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.14}$$

5.3 Offline charging management optimization

The main aim of EV fleet charging optimization is to minimize the cost of energy drawn from the grid:

$$C_{batt} = \sum_{k=0}^{N_t-1} C_{el}(k)\frac{P_g(k)\Delta T}{1000}, \tag{5.15}$$

where $C_{el}(k)$ is the electricity unit price time profile (given in EUR/kWh), while the term $P_g(k)\Delta T/1000$ denotes the grid-supplied charging energy increment in the k^{th} step (expressed in kWh). The charging optimization relies on the aggregate EV fleet model, and as such it is subject to the SoE dynamics equation (5.1), and SoE and charging power inequality constraints (5.3) and (5.4). Additionally, it is required that the final SoE, $SoE_{agg}(N_t)$, is equal to a pre-determined target value SoE_{final}, which is set to be equal to the initial SoE value, $SoE_{init} = SoE_{agg}(0)$:

$$SoE_{final} = SoE_{init},$$

to satisfy the charge sustaining condition.

The optimization problem is solved by using the dynamic programming (DP) algorithm proposed in [13], which provides a globally optimal solution for a general, non-convex optimization problem with non-convex cost function and constraints. The aforementioned SoE constraints are accounted for within the DP formulation via soft constraints $L(k)$ added to the cost function (5.15) as:

$$J = \sum_{k=0}^{N_t-1} \underbrace{C_{el}(k)\frac{P_g(k)\Delta T}{1000}}_{F(k)} + L(k), \tag{5.16a}$$

$$\begin{aligned} L(k) = {} & K_{g,1}\big(SoE_{agg}(k+1)-1\big)H\big(SoE_{agg}(k+1)-1\big) \\ & + K_{g,2}\Big(-SoE_{agg}(k+1)\Big)H\Big(-SoE_{agg}(k+1)\Big) \\ & + K_{g,3}\left(SoE_{agg}(k+1)-\frac{n_c(k+1)}{N_v}\right)H\left(SoE_{agg}(k+1)-\frac{n_c(k+1)}{N_v}\right) \\ & + K_{g,4}H\Big(SoE_{final}-SoE_{agg}(k+1)\Big)H(k-N_t+1), \end{aligned} \tag{5.16b}$$

where the function *H*(.) represents the Heaviside function defined as: $H(z) = 0$ for $z < 0$ and $H(z) = 1$ for $z \geq 0$. Relative importance of the individual terms/constraints are given via related weighting factors $K_{g,i}$, i = 1,...,4, which are all set to high values to enforce constraint satisfaction if possible. The physical SoE constraints $0 \leq SoE_{agg} \leq 1$ are posed to have the highest priority by setting $K_{g,1} = K_{g,2} = 10^9$, while the remaining weighting factors are set as $K_{g,3} = 10^7$ and $K_{g,4} = 10^8$. The aggregate charging power constraints given by Eq. (5.4) are implemented as hard constraints within the DP algorithm. Namely, the aggregate charging power $P_{c,agg}$ is iterated over its predefined discrete values, from zero until reaching $n_c(k)P_{cmax,ind}$ or $P_{c,agg,\max}$, which results in strict satisfaction of these constraints.

The DP procedure [13] consists of two distinctive phases: (i) backward-in-time optimization of charging power to minimize the cost (5.16), and (ii) forward-in-time reconstruction of the optimal SoE and charging power time profiles $SoE_{agg}(k)$ and $P_{c,agg}(k)$, starting from the pre-determined initial SoE condition $SoE_{agg}(0) = SoE_{init}$. Since the DP algorithm requires discrete state and control variables, the originally continuous variables, SoE_{agg} and $P_{c,agg}$, are uniformly quantized into certain number of discrete values. The phase (i) starts from the last time step $N_t - 1$ and iterates backward-in-time until reaching the initial time step 0 (i.e., $k = \{N_t - 1, \dots, 1, 0\}$), while minimizing the cumulative cost function:

$$J_k\big(SoE_{agg,k}\big) = \min_{P_{c,agg,k}} \big\{F\big(SoE_{agg,k}, P_{c,agg,k}, k\big) + J_{k+1}\big(SoE_{agg,k+1}\big)\big\} \tag{5.17}$$

by iterating over discrete aggregate charging power values $P_{c,agg,k}$. The optimal cumulative cost function J_k and the related charging power $P_{c,agg,k}$ in the k^{th} time step are found and stored for each discrete value of $SoE_{agg,k}$. The optimal cumulative cost at the $(k+1)^{th}$ time step, $J_{k+1}(SoE_{agg,k+1})$, is obtained by means of linear interpolation if $SoE_{agg,k+1}$ falls between two discrete grid values of SoE. The SoE in $(k+1)^{th}$ step is obtained by the state equation (5.1) in dependence on the current SoE, $SoE_{agg,k}$, and the charging power $P_{c,agg,k}$. The forward-in-time reconstruction is then performed starting from a pre-defined initial SoE value ($SoE_{agg}(0) = SoC_{init}$), for which the optimal charging power obtained in the backward phase is restored and applied to the state equation (5.1) to get $SoE_{agg,1}$, where the linear interpolation is again applied. The forward procedure is iteratively repeated until the last time step $N_t - 1$.

Apart from the aggregate EV fleet model, the formulated optimization problem given by the cost function (5.16) and solved by the DP algorithm can also be used for charging optimization of individual EVs represented by the model (5.5). Namely, SoC_{agg} from (5.16) is replaced with SoE_i for i^{th} EV charging optimization, N_v is set to 1, and $n_{cb,i}$ from (5.6) is used instead of n_c. While minimizing the cost (5.17) in the DP backward-in-time phase, the individual EV charging power is iterated from 0 until reaching $n_{cs,i}(k)P_{cmax,ind}$ (see (5.7)). The charging optimization is performed separately for each individual EV, as the joint DP optimization of all EVs would not be feasible due to the increased number of state and control variables for the distributed model and consequently prohibitive increase in computational complexity of the DP algorithm. To provide such a decoupled optimization approach, the grid power-related upper constraint (5.4b) on the aggregate charging power is omitted. Although this makes the formulation unrealistic (if the constraint turns out to be violated), the approach can be used for the purpose of benchmarking the aggregated model-based optimization against the more direct distributed model-based one (both implemented without the aggregate charging power constraints).

5.4 Online charging management

5.4.1 Model predictive control

Model predictive control (MPC) is an advanced control technique which combines an optimization-based open-loop control with a closed-loop feedback control. It effectively handles multi-input/multi-output (MIMO) systems, where constraints on state and control variables can explicitly be imposed. MPC is typically executed in a receding horizon manner, while taking into account the current measured (or estimated) process state variables (feedback part), and solving its optimization problem (optimal control) in each sampling time step on the prediction horizon. As its name suggests, while

solving the optimization problem, MPC relies on a model to predict the system (process) behaviour on the prediction horizon. For full performance, external variables should be predicted, as well, based on an external/environment model.

The online EV fleet charging management, executed on the aggregated level, as shown in Fig. 37, is based here on the receding horizon MPC framework (denoted as MPC-REC). The control variable optimization problem is solved within MPC by using the DP algorithm used in offline optimization (Subsection 5.3), with the main difference that it is now run online on the receding horizon of length N_p. The sampling time is set to 15 min (ΔT = 0.25 h), while the prediction horizon is one day (N_p= 96 sampling steps). The MPC optimization problem including cost function and constraints is formulated as (cf. Eq. (5.16)):

$$J = \sum_{j=0}^{N_p-1} \underbrace{C_{el}(j|k)\frac{P_g(j|k)\Delta T}{1000}}_{F(j|k)} + L(j|k), \tag{5.18a}$$

$$\begin{aligned} L(j|k) = {} & K_{g,1}\big(SoE_{agg}(j+1|k)-1\big)H\big(SoE_{agg}(j+1|k)-1\big) \\ & + K_{g,2}\Big(-SoE_{agg}(j+1|k)\Big)H\Big(-SoE_{agg}(j+1|k)\Big) \\ & + K_{g,3}\left(SoE_{agg}(j+1|k)-\frac{n_c(j+1|k)}{N_v}\right)H\left(SoE_{agg}(j+1|k)-\frac{n_c(j+1|k)}{N_v}\right) \\ & + K_{g,4}H\Big(SoE_{final}-SoE_{agg}(j+1|k)\Big)H\big(j-N_p+1\big), \end{aligned} \tag{5.18b}$$

$$0 \le P_{c,agg}(j|k) \le n_c(j|k)P_{cmax,ind}, \tag{5.18c}$$

$$P_{c,agg}(j|k) \le P_{c,agg,\max}, \tag{5.18d}$$

$$\begin{aligned} SoE_{agg}(j+1|k) = {} & SoE_{agg}(j|k) + SoE_{in,avg}(j|k)\frac{n_{in}(j|k)}{N_v} - \\ & SoE_{out,avg}(j|k)\frac{n_{out}(j|k)}{N_v} + \eta_{ch}\frac{P_{c,agg}(j|k)\Delta T}{N_v E_{max,ind}}, \end{aligned} \tag{5.18e}$$

where k denotes the current simulation time step, and j is the time step on the prediction horizon (relative to the current step k; $j = 0, 1, \dots, N_p - 1$). The state variable SoE_{agg} is predicted on the horizon by using the prediction model (5.18e), which is set to have the same structure as the simulation model (5.1). Since the final step is generally outside of the receding horizon, the requirement on the final SoE is omitted here by setting $K_{g,4} = 0$. As in the case of offline DP (Subsection 5.3), the aggregate charging power constraints (5.18c) and (5.18d) are implemented as hard constraints. The DP optimization provides a sequence of optimal aggregate charging power

values $P_{c,agg}(j|k)$ ($j = 0,1,\dots,N_p-1$), and only the first element $P_{c,agg}(0|k)$ is applied in the current, k^{th} sampling step, while the remaining ones are discarded.

Another MPC approach considered performs the optimization on a shrinking horizon (denoted as MPC-DIM), which gradually diminishes as time progresses towards the end time of a day (set typically in the early morning period where the transport system is typically at rest preparing for the next day). Thus, the time-varying length of MPC-DIM prediction horizon $N_{p,dim}(k)$ is set as

$$N_{p,dim}(k) = N_p - k + \left\lfloor \frac{k}{N_p} \right\rfloor N_p, \qquad k = \{0,\dots,N_t-1\}, \tag{5.19}$$

where N_p is the fixed horizon length (equal to N_p = 96), and $\lfloor x \rfloor$ is a mathematical operator providing an integer counterpart of a real number x. Note that the full prediction horizon of length N_p rebuilds when a new day starts. MPC-DIM relies on the same optimization problem (5.18) as MPC-REC, but with the final SoE condition included (i.e., $K_{g,4} = 10^8$ is set in Eq. (5.18b) instead of being equal to zero), as its final step is now contained in the prediction horizon.

The MPC-DIM approach is deemed as a reasonable alternative option since the fleet driving schedules are planned offline one day ahead. Apart from that, the MPC-DIM is characterized by an improved computational efficiency since its prediction horizon length is shorter in average when compared to MPC-REC, and thus related optimization executes faster.

5.4.2 Preparation of MPC input distributions

The following input time profiles of individual EVs denoted by the subscript $i = 1,2,\dots,N_v$ should be predicted over the prediction horizon $j = 0,1,\dots,N_p-1$: $n_{in,i}(j|k)$, $n_{out,i}(j|k)$, $SoE_{in,i}(j|k)$, and $SoE_{out,i}(j|k)$, to serve as a basis for calculating the following input time profiles of the aggregate battery model (5.18e) needed for MPC optimization: $n_{in}(j|k)$, $n_{out}(j|k)$, $n_c(j|k)$, $SoE_{in,avg}(j|k)$, and $SoE_{out,avg}(j|k)$. While the arrival and departing times of each EV, $n_{in,i}(j|k)$ and $n_{out,i}(j|k)$, may be predicted from the planned driving schedules, the SoE of the arriving EVs, $SoE_{in,i}(j|k)$, should be predicted by using a transport energy demand model (below denoted by $f_{SoE}(.)$).

To maximize the vehicle range and also simplify the energy demand model, it may be assumed that EV batteries are always fully charged when disconnecting from the grid and departing, i.e., $SoE_{out,i}(j|k) = 1$ when $n_{out,i}(j|k) = 1$ [12, 13]. However, it may happen that an EV is parked and connected to the grid for a relatively short amount of time between two driving missions and cannot be fully charged under present charging power limit of $P_{cmax,ind}$. To satisfy the departure schedule, it disconnects from the charger before the battery is full, and eventually rely on fast charging on road if

the energy charged (at depot or e-hub) is not high enough to cover the trip energy demand. Thus, $SoE_{out,i}$ profiles should carefully be prepared to have the maximal possible values of 1, if possible, while not violating the individual charging power limit. For this purpose, the distributed model (5.8)-(5.11), whose equations are rewritten below, is evaluated over the prediction horizon ($j = 0,1, \dots, N_p - 1$; in the recursive sense) for the scheduled profiles $n_{in,i}(j|k)$ and $n_{out,i}(j|k)$, the maximum charging power $P_{cmax,ind}$, and the known initial conditions: $SoE_i(0|k)$, $SoE_{in,i}(0|k)$, $n_{in,i}(0|k)$, and $SoE_{out,i}(0|k)$.

$$SoE_i(j+1|k) = \begin{cases} SoE_{int,i}(j|k), & \text{for } n_{out,i}(j|k) = 0, \\ 0, & \text{for } n_{out,i}(j|k) = 1, \end{cases} \tag{5.20a}$$

$$SoE_{int,i}(j|k) = \min\left(1, SoE_i(j|k) + SoE_{in,i}(j|k) n_{in,i}(j|k) + \eta_{ch} \frac{P_{cmax,ind} \Delta T}{E_{max,ind}}\right), \tag{5.20b}$$

$$SoE_{in,i}(j|k) = \begin{cases} 0, & \text{for } n_{in,i}(j|k) = 0, \\ f_{SoE}\left(SoE_{out,i}(j|k), d_i(j|k)\right), & \text{for } n_{in,i}(j|k) = 1, \end{cases} \tag{5.20c}$$

$$SoE_{out,i}(j+1|k) = \begin{cases} SoE_{out,i}(j|k), & \text{for } n_{out,i}(j|k) = 0, \\ SoE_{int,i}(j|k), & \text{for } n_{out,i}(j|k) = 1. \end{cases} \tag{5.20d}$$

Note that the expression (5.20b) effectively saturates SoE to the maximum possible value of 1 if it is reached prior to vehicle departure (under consistent application of the maximum charging power). The obtained SoE value at departures is used as the input for the transport demand model $f_{SoE}(\cdot)$ in Eq. (5.20c) to predict the SoE at arrival (i.e., return to depot) and connections to the grid (when $n_{in,i}(j|k) = 1$).

The number of arriving and departing EVs for the aggregate battery prediction model are calculated by summing up the individual profiles:

$$n_{in}(j|k) = \sum_{i=1}^{N_v} n_{in,i}(j|k), \tag{5.21a}$$

$$n_{out}(j|k) = \sum_{i=1}^{N_v} n_{out,i}(j|k). \tag{5.21b}$$

Similarly, the aggregate battery SoE time profiles are calculated by averaging the SoE profiles of individual EVs:

$$SoE_{in,avg}(j|k) = \frac{\sum_{i=1}^{N_v} SoE_{in,i}(j|k) n_{in,i}(j|k)}{\sum_{i=1}^{N_v} n_{in,i}(j|k)}, \tag{5.22a}$$

$$SoE_{out,avg}(j|k) = \frac{\sum_{i=1}^{N_v} SoE_{out,i}(j|k) n_{out,i}(j|k)}{\sum_{i=1}^{N_v} n_{out,i}(j|k)}. \quad (5.22b)$$

The number of EVs parked and connected to the grid within each time step, needed for establishing the upper power limit (5.4a), is determined from the time profiles of number of arriving and departing EVs:

$$n_c(j+1|k) = n_c(j|k) + n_{in}(j|k) - n_{out}(j|k), \qquad j = 0,1,\dots,N_p - 1, \quad (5.23)$$

where the initial condition $n_c(0|k)$ corresponds to the known, current number of connected EVs. Additional time profiles needed for MPC optimization, are related to the electricity price $C_{el}(j|k)$ and the RES power production $P_{res}(j|k)$, and they should also be predicted, for instance based on historical data and meteorological forecasts.

5.4.3 Distribution of aggregate charging power to individual vehicles

The aggregate charging power $P_{c,agg}(k)$, obtained by MPC in the k^{th} time step, should be distributed to the connected individual EVs. For this purpose, a rule-based algorithm is established which prioritizes to charge EVs with lower SoE and lower remaining connection times (i.e., sooner departure). The related procedure, summarized below and formulated in Algorithm 3, is generally iterative since saturation of individual charging power due to the upper limits (5.11) may inhibit one-shot aggregate power distribution.

The procedure starts by calculating the lower and upper individual charging power limits, $P_{c,\min,i}(k)$ and $P_{c,\max,i}(k)$, where $P_{c,\max,i}(k)$ is given by Eq. (5.11), while $P_{c,\min,i}(k)$ is determined according to the requirement that each EV is targeted to have the maximum possible SoE (equal to 1) each time when disconnecting from the grid (leading to the maximum vehicle range). More specifically, $P_{c,\min,i}(k)$ is derived from Eq. (5.8b) under the assumption that i^{th} EV will be charged with the maximum power $P_{cmax,ind}$ from the following $(k+1)^{\text{th}}$ time step until the end of connection time $t_{c,i}$.

$$1 = SoE_i(k) + SoE_{in,i}(k) n_{in,i}(k) + \eta_{ch} \frac{n_{cs,i}(k) P_{c,i}(k) \Delta T + \left(t_{c,i}(k) - n_{cs,i}(k)\Delta T\right) P_{cmax,ind}}{E_{max,ind}}. \quad (5.24)$$

Eq. (5.24) is solved for $P_{c,i}(k)$ to get the minimum charging power $P_{c,min,i}(k)$ in the k^{th} step under which the i^{th} EV battery can still be fully charged until departure:

$$P_{c,\min 0,i}(k) = \frac{1}{n_{cs,i}(k)\Delta T}\left(\frac{E_{max,ind}}{\eta_{ch}}\left(1 - SoE_i(k) - SoE_{in,i}(k)n_{in,i}(k)\right)\right. \\ \left. - \left(t_{c,i}(k) - n_{cs,i}(k)\Delta T\right)P_{cmax,ind}\right). \tag{5.25}$$

The upper charging power constraint (5.11) is set to have priority over the lower constraint (5.25), i.e., the maximum charging power constraint cannot be violated, while the SoE at departure can be lower than 1 if the battery cannot be fully charged due to short connection/parking time. To this end, the lower limit $P_{c,min,i}(k)$ of each EV is saturated to $P_{c,max,i}(k)$ as:

$$P_{c,\min,i}(k) = \begin{cases} P_{c,\min 0,i}(k), & \text{for } P_{c,\min 0,i}(k) \le P_{c,max,i}(k), \\ P_{c,max,i}(k), & \text{for } P_{c,\min 0,i}(k) > P_{c,max,i}(k). \end{cases} \tag{5.26}$$

The individual charging power values are then initialized to their lower limit values:

$$P_{c,i}(k) = \begin{cases} P_{c,min,i}(k), & \text{for } P_{c,min,i}(k) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.27}$$

They are rescaled by the factor $P_{c,agg,max}/\sum_{i=1}^{N_v} P_{c,i}(k)$ if their sum exceeds the allowed aggregate charging power $P_{c,agg,max}$ given by Eq. (5.4b) (i.e., if $\sum_{i=1}^{N_v} P_{c,i}(k) > P_{c,agg,max}$). The remained aggregate charging power is then calculated as:

$$P_{c,agg,r}(k) = P_{c,agg}(k) - \sum_{i=1}^{N_v} P_{c,i}(k), \tag{5.28}$$

which is distributed over individual EVs according to shares $p_i(k)$, set to be proportional to the deviation of corresponding SoE from 1 (i.e., from being fully charged), and inversely proportional to the remaining connection time $t_{c,i}(k)$:

$$p_i(k) = \frac{1 - SoE_i(k) - SoE_{in,i}(k)n_{in,i}(k) - \eta_{ch}\dfrac{P_{c,i}(k)\Delta T}{E_{max,ind}}}{t_{c,i}(k)}. \tag{5.29}$$

These shares are calculated only for those EVs connected to the grid, $n_{cb,i}(k) = 1$, for which the currently designated charging power values $P_{c,i}(k)$ are lower than the related maximum values $P_{c,\max,i}(k)$, $P_{c,i}(k) < P_{c,\max,i}(k)$ (i.e., those that can still accommodate additional charging power). For other EVs, they are preset to zero, $p_i(k) = 0$. Then, the calculated shares are normalized:

Algorithm 3. Algorithm of aggregate charging power distribution over individual EVs.

- Calculate charging power limits $\boldsymbol{P_{c,\min,i}(k)}$ and $\boldsymbol{P_{c,\max,i}(k)}$ for connected EVs ($\boldsymbol{n_{cb,i}(k)=1}$) according to Eqs. (5.26) and (5.11), respectively.
- Set priority to upper power limit: $\boldsymbol{P_{c,\min,i}(k) \leftarrow \min\left(P_{c,\min,i}(k), P_{c,\max,i}(k)\right), \forall i.}$
- Set individual charging powers at lower limits (if being larger than zero): $\boldsymbol{P_{c,i}(k) \leftarrow \max\left(0, P_{c,\min,i}(k)\right), \forall i.}$
- If $\sum_{i=1}^{N_v} P_{c,i}(k) > P_{c,agg,max}$ holds, rescale individual charging powers as $\boldsymbol{P_{c,i}(k) \leftarrow P_{c,i}(k)\frac{P_{c,agg,max}}{\sum_{i=1}^{N_v} P_{c,i}(k)}}$, to satisfy upper constraint on aggregate charging power (5.4b).
- Calculate remained aggregate charging power: $\boldsymbol{P_{c,agg,r}(k) = P_{c,agg}(k) - \sum_{i=1}^{N_v} P_{c,i}(k)}$.
- Initialize $\boldsymbol{p_i(k)=1, \forall i.}$

while $\boldsymbol{\sum_{i=1}^{N_v} p_i(k) \neq 0}$ **and** $\boldsymbol{P_{c,agg,r}(k) > 0}$

- Calculate shares $\boldsymbol{p_i(k)}$ for all EVs ($i = 1, 2, \ldots, N_v$) by using Eq. (5.29).
- Calculate normalized shares: $\boldsymbol{\bar{p}_i(k) = \begin{cases} p_i(k)/\sum_{i=1}^{N_v} p_i(k), & \textbf{for } \sum_{i=1}^{N_v} p_i(k) > 0, \\ 0, & \textbf{otherwise.} \end{cases} \forall i.}$
- Distribute remained charging power $\boldsymbol{P_{c,agg,r}(k)}$ in dependence on $\boldsymbol{\bar{p}_i(k)}$ as:

$$\boldsymbol{P'_{c,i}(k) = P_{c,i}(k) + \bar{p}_i(k) P_{c,agg,r}(k), \forall i.}$$

- Saturate individual charging powers with respect to their upper limits:

$$\boldsymbol{P_{c,i}(k) = \min\left(P'_{c,i}(k), P_{c,\max,i}(k)\right), \forall i,}$$

- Update remained aggregate charging power as:

$$\boldsymbol{P_{c,agg,r}(k) = P_{c,agg}(k) - \sum_{i=1}^{N_v} P_{c,i}(k)}$$

end while

$$\bar{p}_i(k) = \begin{cases} \dfrac{p_i(k)}{\sum_{i=1}^{N_v} p_i(k)} & \text{if } \sum_{i=1}^{N_v} p_i(k) > 0, \\ 0, & \text{otherwise,} \end{cases} \quad \forall i. \tag{5.30}$$

and as such they are used for distributing the remained aggregate charging power $P_{c,agg,r}(k)$:

$$P'_{c,i}(k) = P_{c,i}(k) + \bar{p}_i(k) P_{c,agg,r}(k), \forall i. \tag{5.31a}$$

$$P_{c,i}(k) = \min\left(P'_{c,i}(k), P_{cmax,i}(k)\right), \forall i. \tag{5.31b}$$

Note that $P'_{c,i}(k)$ determined by Eq. (5.31a) is used to update $P_{c,i}(k)$ in Eq. (5.31b) (as the final control input over its preliminary value given by Eq. (5.27) and used in (5.31a), see Algorithm 3). The distribution procedure represented by Eqs. (5.28)-(5.31) is iteratively repeated until the remained

aggregate power $P_{c,agg,r}(k)$ given by Eq. (5.28), which is yet to be distributed, is brought to zero, or all shares $p_i(k)$ become zero ($\sum_{i=1}^{N_v} p_i(k) = 0$ in Eq. (5.29), i.e., there are no EVs available for charging).

The presented distribution algorithm can be applied both in an offline and online manner. In the offline case, the whole aggregate power sequence is obtained offline (e.g., by DP optimization and aggregate battery model) and then it is distributed over individual EVs step-by-step by using Algorithm 3 (no feedback present). In the online case, the distribution algorithm is performed after getting the optimal charging power $P_{c,agg}(0|k)$ by executing the MPC algorithm in the actual, k^{th} sampling step, and using it to determine the individual charging power values in the same sampling step employing Algorithm 3 (feedback is present through the MPC path).

5.4.4 Baseline (dumb) charging strategy

A so-called dumb charging strategy is introduced to serve as a baseline for verification of the developed MPC charging strategy. The idea is to charge the aggregate battery as soon as possible, without accounting for electricity price or production from RES. The dumb strategy is applied to both aggregate and distributed models.

For the aggregate model, the aggregate charging power that brings the current aggregate SoE to SoE_{final} (typically set to 1) is calculated by rearranging (5.1) as:

$$P'_{c,agg}(k) = \frac{SoE_{final} n_c(k+1) - SoE_{agg}(k) N_v - SoE_{in,avg}(k) n_{in}(k) + SoE_{out,avg}(k) n_{out}(k)}{\eta_{ch} \Delta T} E_{max,ind} \tag{5.32}$$

This power is then saturated to avoid violation of the aggregate charging power limits defined by (5.4):

$$P_{c,agg}(k) = \min\left(P'_{c,agg}(k), \min\left(n_c(k) P_{cmax,ind}, P_{c,agg,max}\right)\right). \tag{5.33}$$

By targeting $SoE_{final}\, n_c(k+1)/N_v$ as the aggregate SoE value in each time step, *dumb charging strategy* achieves $SoE_{agg}(k) = SoE_{final}$ at the end time N_t where $n_c(N_t) = N_v$. Thus, it should be directly comparable in terms of charging cost with other charging strategies (e.g., offline DP and MPCs) by setting them to target the same final SoE (i.e., SoE_{final}).

For the distributed model, the dumb charging strategy sets individual charging power values in each time step k to their maximum values $P_{c,\max,i}(k)$ given by Eq. (5.11) if not violating the upper limit on the aggregate charging power given by Eq. (5.4b) (see the first condition below); otherwise, they are set to values obtained by scaling down $P_{c,\max,i}(k)$ in the way that satisfies the aggregate power limit (second condition below):

$$P_{c,i}(k) = \begin{cases} P_{c,\max,i}(k), & \text{for } \sum_{i=1}^{N_v} P_{c,\max,i}(k) \le P_{c,agg,\max} \\ P_{c,\max,i}(k)\dfrac{P_{c,agg,\max}}{\sum_{i=1}^{N_v} P_{c,\max,i}(k)}, & \text{otherwise.} \end{cases}, \forall i. \quad (5.34)$$

5.5 Case study

5.5.1 Parametrization of EV fleet models

EV fleet models described in Subsection 5.2 are parameterized by using the data recorded for a delivery vehicle fleet of a local retail company [14]. The data were recorded on ten mid-size Diesel engine-propelled delivery trucks by using GPS/GPRS equipment over a three-month period. The vehicles mission was to deliver cargo from a distribution centre (a depot; DC) to different sales centres. These trucks were virtually converted to extended range electric vehicles (EREV) with similar power and torque characteristics as in the real trucks [15, 20]. EREVs (denoted as EVs hereafter for the sake of brevity) were used instead of pure battery electric vehicles (BEV) to overcome limited range of BEVs, i.e., to be able to cover all recorded driving missions (both short- and long-distance ones). It was assumed that their charging could take place only at the DC during their parking periods between two driving missions. Thus, the recorded GPS positions were used to detect time periods when vehicles had been located within the DC and thus hypothetically connected to the grid and available for charging. From this data, the following time profiles of EV fleet models from Subsection 5.1 could be derived: n_{in}, n_{out}, n_c, $n_{cb,i}$, $n_{cs,i}$, $n_{in,i}$, $n_{out,i}$, where $i = 1,2,\dots,N_v$.

The above-described setup has been adopted for the case study presented herein, with the following notes: (i) a one week period is selected out of the total three-month period, and (ii) the time discretization is reduced from 1 h to 15 minutes, i.e., ΔT = 0.25 h. To ensure that each day starts and ends with all vehicles being parked within the DC, the week is set to start at 5 a.m. of the first recording day. This is illustrated by the time profile of the number of vehicles within the DC, $n_c(t)$, shown in in Fig. 38, which is determined from $n_{in}(k)$ and $n_{out}(k)$ (in the same way as defined by Eq. (5.23) for the MPC horizon):

$$n_c(k+1) = n_c(k) + n_{in}(k) - n_{out}(k), \qquad k = 0,1,\dots,N_t - 1, n_c(0) = N_v. \quad (5.35)$$

The profiles shown in Fig. 38 reveal a repetitive fleet activity over workdays (from Monday to Friday) with the peak activity occurring around 10 a.m., and the reduced activity appearing over weekend days (Saturday and Sunday).

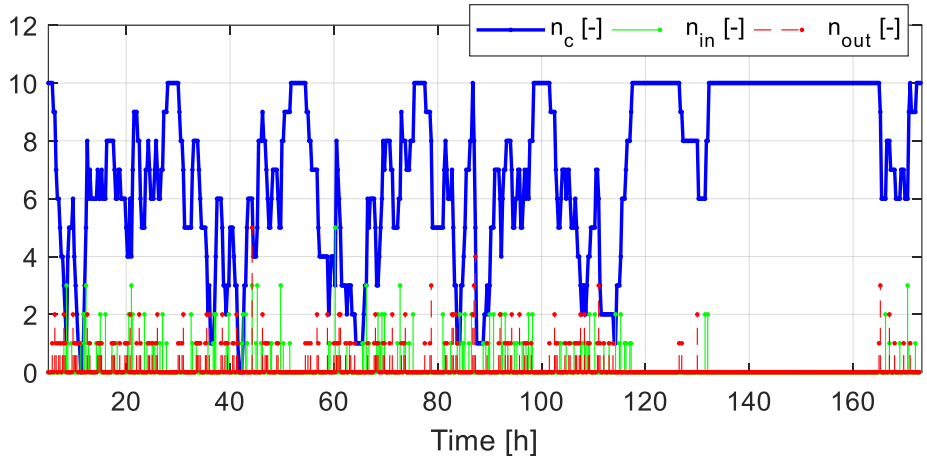


Fig. 38 Time profiles of aggregate EV fleet model related to number of connected EVs (n_c), number of arriving EVs (n_{in}), and number of departing EVs (n_{out}) over a one-week period (total number of EVs within fleet is 10, N_v = 10)

The backward-looking type of EREV model given in series configuration is shown in Fig. 39 (see [20] and [15] for more details). The main propulsion comes from the bigger electric machine (denoted as Motor), while the internal-combustion engine drives the generator to sustain the battery SoC to its lower limit value, thus providing the vehicle range extension. The battery capacity is set to 72.67 kWh. The control strategy is assumed to operate in the so-called CD/CS (Charge Depleting/Charge Sustaining) regime, where CD corresponds to pure electric driving (i.e., $\omega_e = 0$ and $\tau_e = 0$), while CS relates to hybrid driving. The control in CS regime is based on an equivalent consumption minimization strategy (ECMS), which sets the engine operating point, ω_e and τ_e, to minimize an equivalent fuel consumption cost.

The recorded driving cycles are divided into ten groups with respect to their travelled distance, and for each group one statistically representative synthetic driving cycle is generated by using Markov chain methodology [14]. The EREV model is simulated over each synthetic driving cycle for nine different initial battery state-of-charge (SoC) values, $SoC_{out} = \{0.2, 0.3, \dots, 1\}$, resulting in a grid (10x9) of SoC-at-destination (SoC_{in}) and fuel consumption (V_f) values shown in Fig. 40. Note that the simulation model concerns a more accurate vehicle battery model expressed in SoC rather than SoE state variable, while the optimization and control algorithms are based on SoE state equations, where the SoE is equalized with the SoC when parameterizing the vehicle energy demand model for optimization/control. Note also that SoC_{in} ends up around 0.3 for larger travelled distances and/or lower initial SoCs, which corresponds to lower limit of SoC sustained within the CS regime.

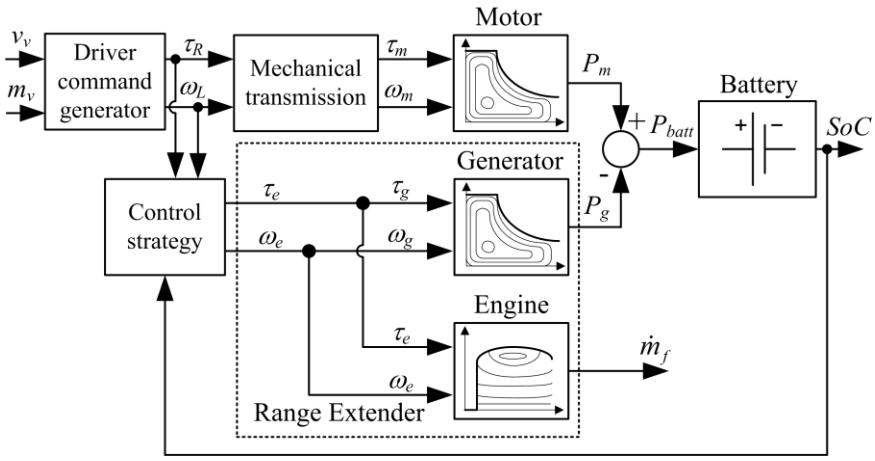


Fig. 39 Block diagram of backward-looking model of Extended Range Electric Vehicle (EREV)

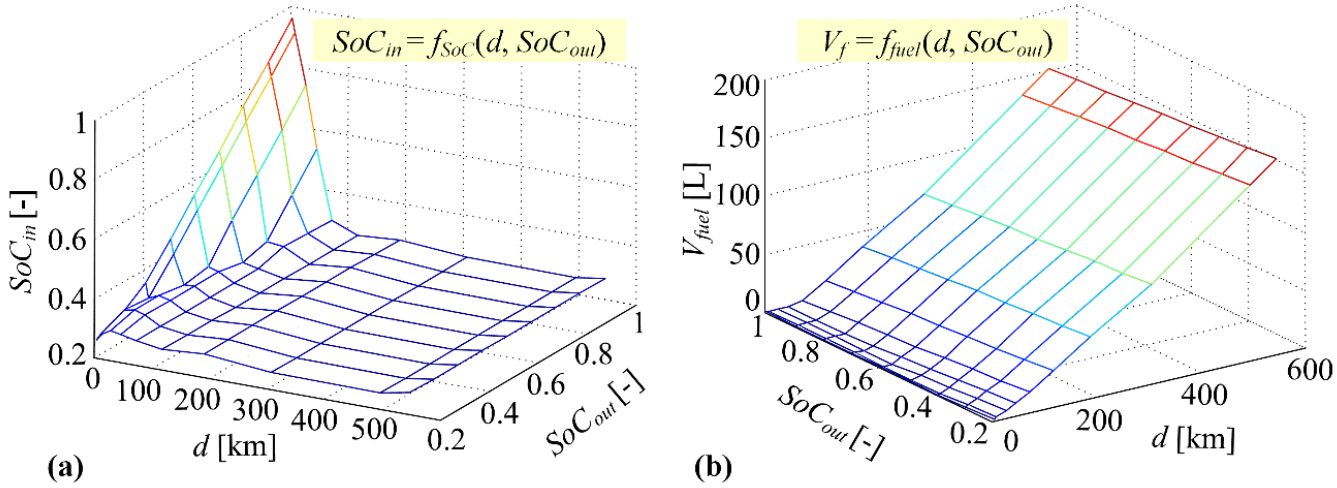


Fig. 40 Response surface-based transport demand model providing SoC-at-destination (SoC_{in}) when arriving to DC (a), and fuel consumption (V_{fuel}) of related driving mission of length d

The SoE time profiles of individual EVs, $SoE_{in,i}(k)$ and $SoE_{out,i}(k)$, are derived by using the transport demand model from Fig. 40 and the expressions (5.20). According to Eq. (5.20), SoE of departing EVs, $SoE_{out,i}$, is set to 1 whenever it is possible if it is not limited by the individual charging power limit. The average SoE profiles of departing and arriving EVs, required by the aggregate battery model, are calculated from individual profiles by using Eq. (5.22) and they are shown in Fig. 41.

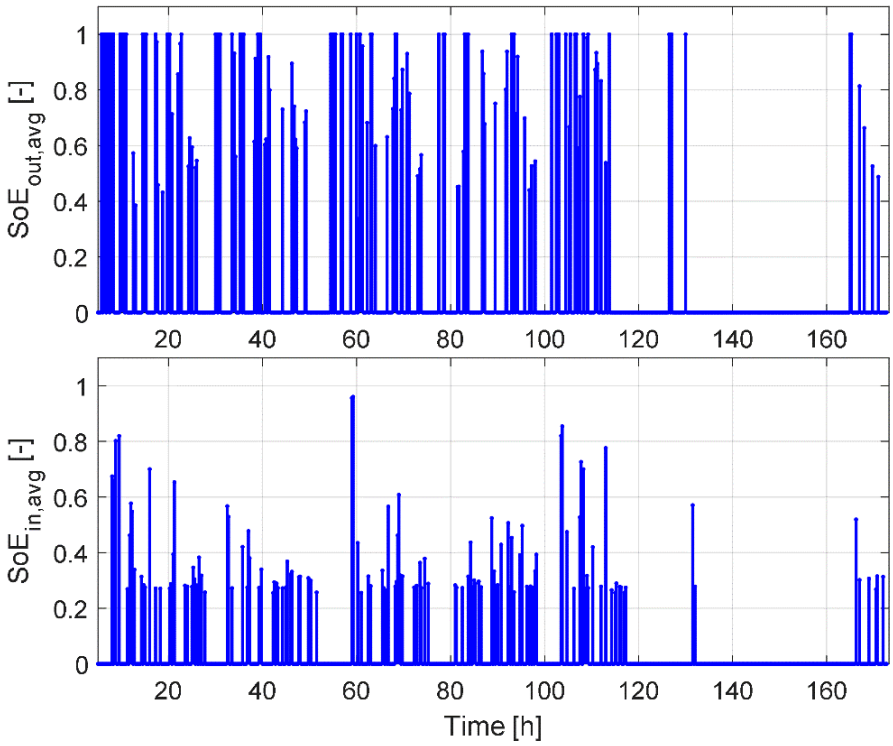


Fig. 41 Average SoE of EVs departing from DC ($SoE_{out,avg}$) and arriving to DC ($SoE_{in,avg}$)

The two-tariff electricity price model present in Croatia is represented by the plot shown in Fig. 42. The RES power time-profile shown in Fig. 43 relates to power production from solar panels hypothetically installed on the DC roofs. It was obtained from irradiation of global radiation (kW/m^2) taken from Meteonorm software for the particular location of DC, which is multiplied by the assumed solar panel surface equal to 2000 m^2 [20].

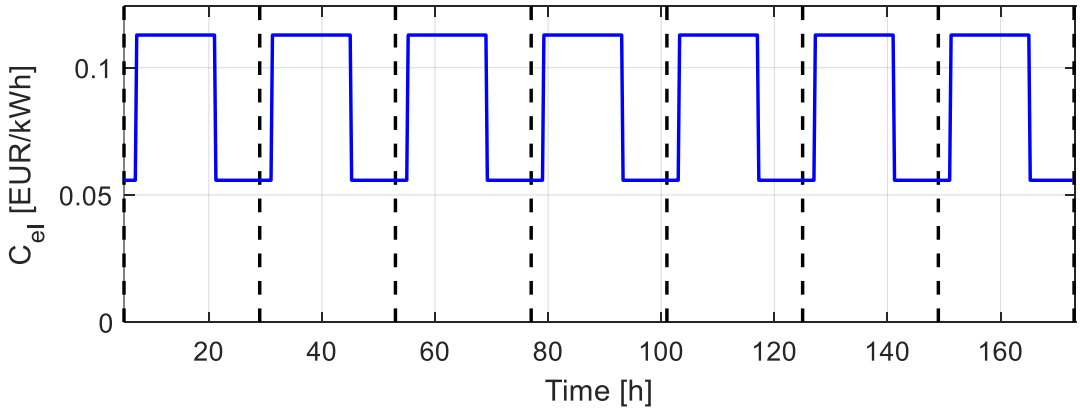


Fig. 42 Time profile of two-tariff electricity price over one-week period (vertical dashed lines denote boundaries between days starting at 5 a.m.)

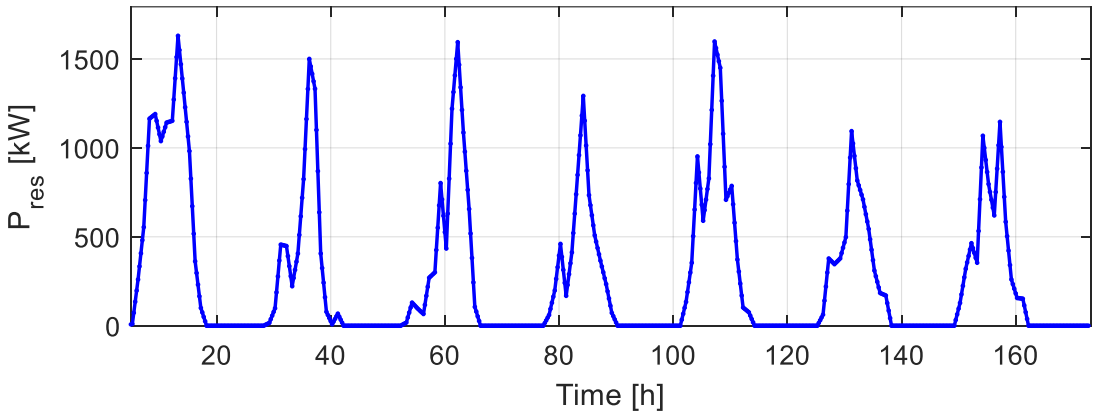


Fig. 43 Time profile of hypothetical power production from solar panels over one-week period

The remaining EV fleet model and simulation parameters are set to: $\Delta T = 0.25$ h, $N_p = 96, N_v = 10$, $E_{max,ind} = 72.67$ kWh, $\eta_{ch} = 0.92$, $SoE_{final} = 0.95$, $P_{c,agg,\max} = 150$ kW, $N_t = 672$, $P_{cmax,ind} = 25$ kW.

To solve the DP optimization problem for the case of the aggregate battery model, the aggregate SoE and charging power are quantized as $SoE_{agg} \in \{0, 0.01, \dots, 1\}$ and $P_{c,agg} \in \{0, 2, \dots, 150\}$ kW, respectively (see Subsection 5.3). In the case of separate DP optimizations for individual EVs, these quantization levels are set to: $SoE_{agg} \in \{0, 0.01, \dots, 1\}$ and $P_{c,agg} \in \{0, 0.5, \dots, 25\}$ kW.

5.5.2 Results for case of no RES consideration

Firstly, the aggregate battery model (5.1)-(5.4) is used as an EV fleet simulation model for conducting the offline DP optimization and testing the online MPC charging strategies in the case of no electric power production from RES (see Fig. 44 for illustration of the latter). The MPC charging on receding

horizon is denoted as MPC-REC, while the one related to diminishing horizon is referred to as MPC-DIM. The offline DP and the baseline (dumb) charging strategy relying on Eq. (5.33) are denoted as DP-OFF and DUMB, respectively.

As MPC-REC does not have requirement on SoE at the end of prediction horizon, it is switched to MPC-DIM when its prediction horizon reaches the end of simulation time (i.e., when reaching the last day of a week), to force SoE to reach the same final target value of 0.95 as in the case of other charging optimization/strategies, and thus facilitate comparative analyses.

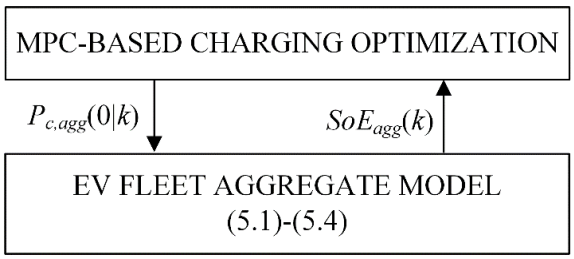


Fig. 44 Online MPC applied to aggregate EV fleet simulation model

Figs. 45 and 46 show the obtained aggregate SoE and charging power profiles, which reveal that MPC-REC profiles closely align with those of the DP-OFF benchmark. MPC-DIM provides somewhat different profiles, which is because its formulation includes the constraint on final SoE at the end of each day to be equal to 0.95, which is not present in MPC-REC and DP-OFF. However, those differences in time profiles do not cause any notable difference in charging costs of DP-OFF, MPC-REC, and MPC-DIM, as evidenced in Table 13. Note also that all approaches provide the same fuel consumption calculated according to the map from Fig. 40b, which is dictated by the same SoE time profiles, $SoC_{in,avg}$ and $SoC_{out,avg}$, used in the aggregate model.

The DUMB charging approach brings the SoE very close to its upper limit, which is due to charging with the maximum power possible (see Subsection 5.4). Unawareness of electricity price is reflected in relatively high charging power levels in the periods of high electricity cost, which results in approximately 20% higher charging cost when compared to other approaches (Table 13).

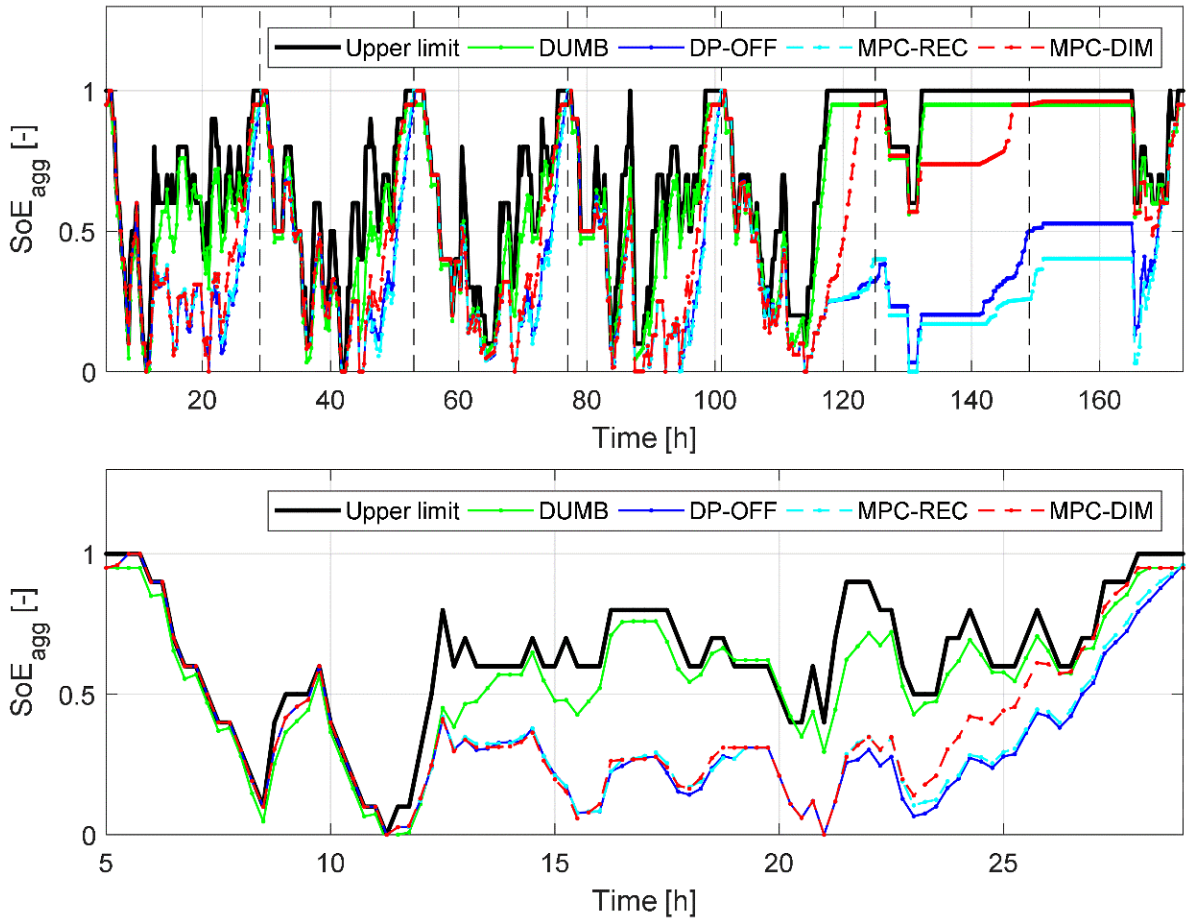


Fig. 45 Aggregate SoE obtained by different charging approaches applied to aggregate battery model, with lower plot representing a zoom-in section of upper plot related to first day profiles

Table 13 Optimization and simulation results obtained for case of aggregate model and no RES considered

Aggregate model, one week	Initial SoC [-]	Final SoC [-]	Fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]
DP-OFF	0.95	0.95	4259.8	6935.0 (0.0%)	515.4 (0.0%)
DUMB	0.95	0.95		6935.0 (0.0%)	617.6 (+19.8%)
MPC-REC	0.95	0.95		6935.0 (0.0%)	515.4 (0.0%)
MPC-DIM	0.95	0.95		6935.0 (0.0%)	515.4 (0.0%)

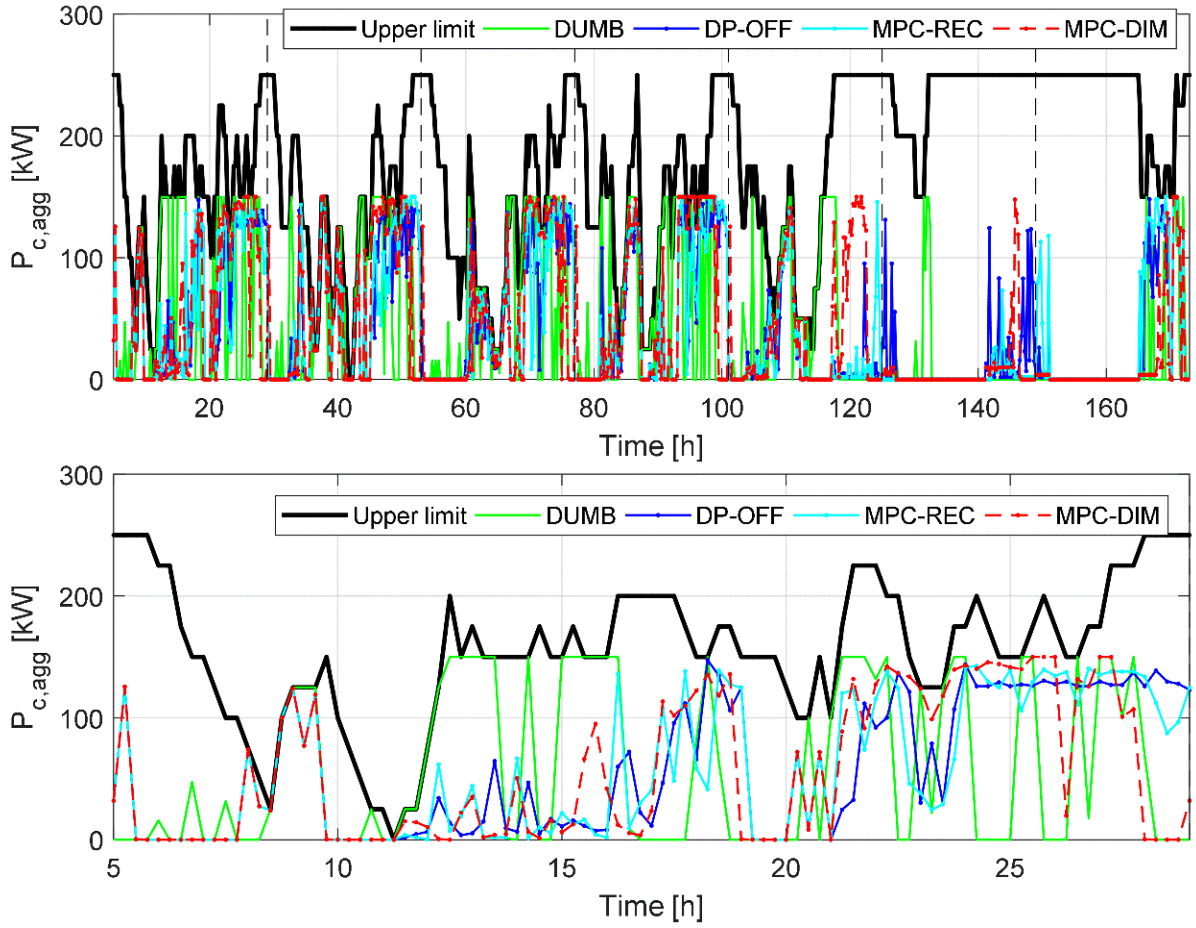


Fig. 46 Aggregate charging power obtained by different charging approaches applied to aggregate battery model, with lower plot representing a zoom-in section of upper plot related to first day profiles

Furthermore, the considered charging approaches have been tested on the distributed vehicle fleet model (5.8)-(5.11). The DP-OFF aggregate charging power profile is distributed over individual EVs by using the distribution algorithm (Algorithm 3 in Subsection 5.4), while DUMB charging is applied directly on the distributed model based on charging power values calculated by Eq. (5.34). Both MPC-REC and MPC-DIM are applied in an online manner while simulating the fleet distributed model, i.e., in each time step the DP optimization is conducted on the prediction horizon by using the aggregate model and the obtained aggregate charging power in the actual time step is distributed over individual EVs by the distribution algorithm (see Fig. 47).

Fig. 48 shows the aggregate SoE and charging power time profiles obtained by DP-OFF prior and after performing distribution, where it can be observed that the distribution does not perturb the

aggregate charging power profile significantly. This is confirmed by a relatively high correlation index of the two power profiles equal to 0.78 (correlation index of SoE profiles is equal to 0.91; its ideal value is 1 on the range [0,1]).

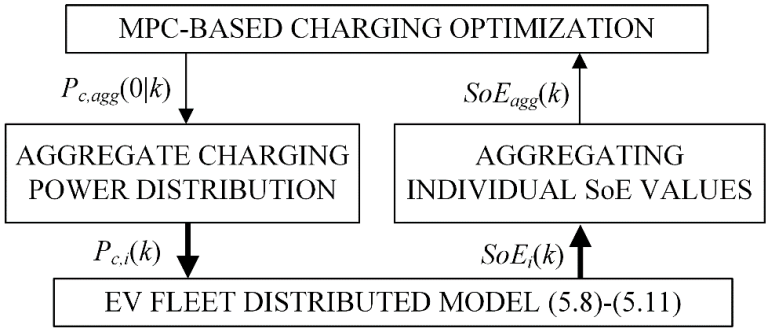


Fig. 47 Online MPC applied to distributed EV fleet simulation model

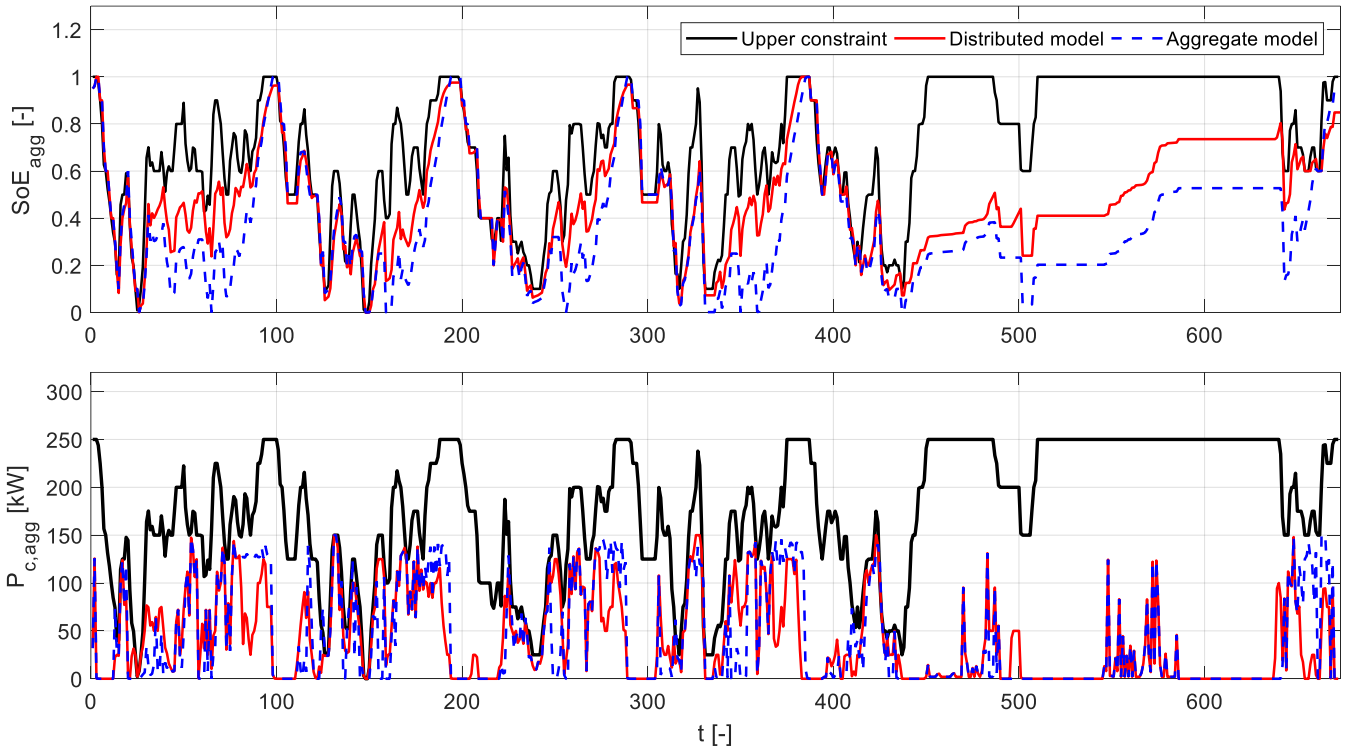


Fig. 48 Comparative plots of aggregate SoE and charging power profiles obtained directly by DP-OFF (blue) and after applying distribution algorithm and aggregating resulting profiles (red)

The first row of Table 14 shows comparative DP optimization results for the cases of using the aggregate model (AGG) and distributing the aggregate charging power over individual EVs (DISTR). These results indicate that distributing of the aggregate power typically results in higher charging costs (by 10%), with negligible reduction of total charging energy (1.5%) and related increase in total fuel consumption (0.2%). MPC-DIM in the case of distributed model ends up with the final SoE at the target value of 0.95 due to its online execution and related feedback effects, which is not the case with DP-OFF-DISTR as its charging power is distributed offline, i.e., in an open-loop manner. Certain differences between the aggregate and distributed models is primarily manifested in charging costs, and they can be attributed to inaccuracies of the aggregate battery model, which cannot fully capture distributed model dynamics.

Table 14 Comparative performance metrics related to results obtained by using aggregate model and distributed model (no RES considered)

One week period	EV fleet model	Initial SoE [-]	Final SoE [-]	Total fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]
DP-OFF	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	515.4 (0.0%)
	DISTR	0.95	0.85	4266.6 (+0.2%)	6830.6 (-1.5%)	565.3 (+9.7%)
DUMB	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	617.6 (0.0%)
	DISTR	0.95	0.96	4273.4 (+0.3%)	6894.0 (-0.6%)	605.1 (-2.0%)
MPC-REC	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	515.4 (0.0%)
	DISTR	0.95	0.95	4267.8 (+0.2%)	6906.1 (-0.4%)	564.1 (+9.5%)
MPC-DIM	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	515.4 (0.0%)
	DISTR	0.95	0.95	4267.8 (+0.2%)	6906.1 (-0.4%)	550.6 (+6.8%)

The DP optimizations are then performed for each EV separately by using the model (5.5)-(5.7), to set a benchmark on the distributed level. The related results are denoted below by the acronym DP-IND. Fig. 49 shows the optimal SoE and charging power profiles for EV #1. It may be observed that SoE increases almost to the upper limit of 1 if the parking period (n_{cb} =1) is long enough (red line in Fig. 49). On the contrary, if this interval is short, the battery is just a partially charged ($SoE < 1$) even though the maximum charging power is applied (see for instance SoE values and charging power when

departing from DC around 40th hour). It should be recalled that the upper limit on the aggregate charging power, given by Eq. (5.4b), is not included here to make the distributed system optimization feasible (see Subsection 5.4), and the obtained results may be somewhat overoptimistic for that reason (due to less constraints involved). However, the aggregate charging power profile obtained from individual profiles by using Eq. (5.13) and shown in Fig. 50 reveals that this violation turns out to occur only in several time steps (out of 672). For that reason, these results may be considered as the (nearly) globally optimal benchmark on the distributed level in the selected settings of the maximum aggregate charging power.

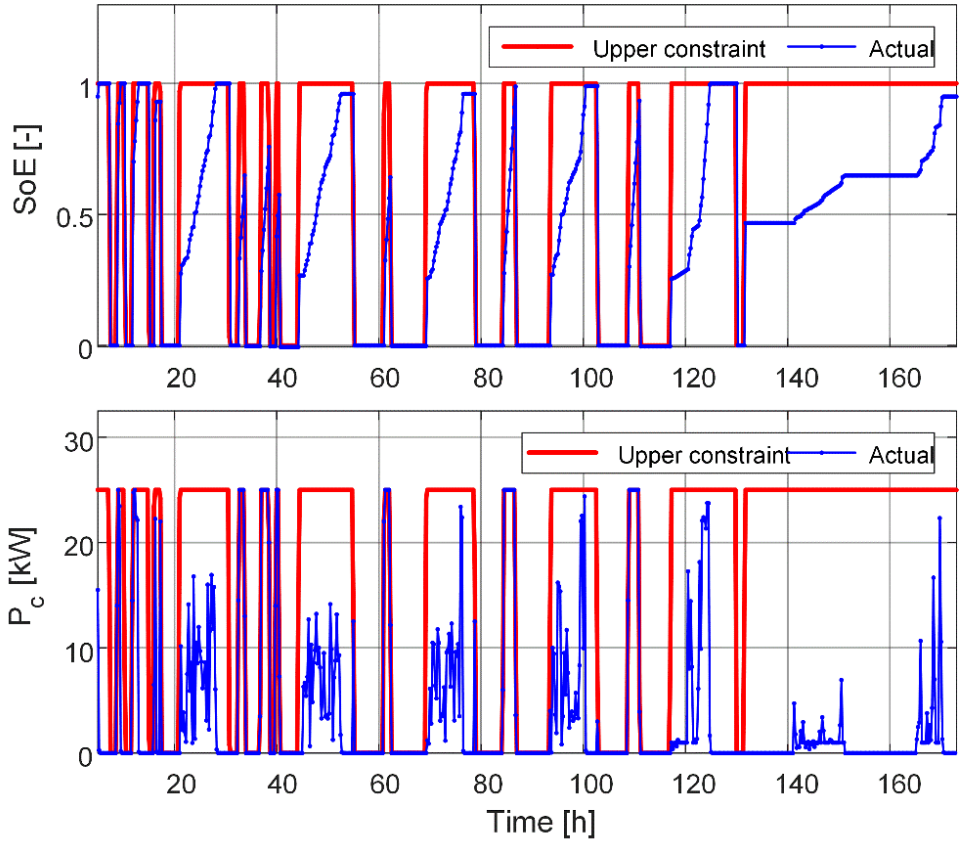


Fig. 49 DP optimized time profiles of SoE and charging power for EV #1 (upper constraint corresponds to ncb from Eq. (5.6))

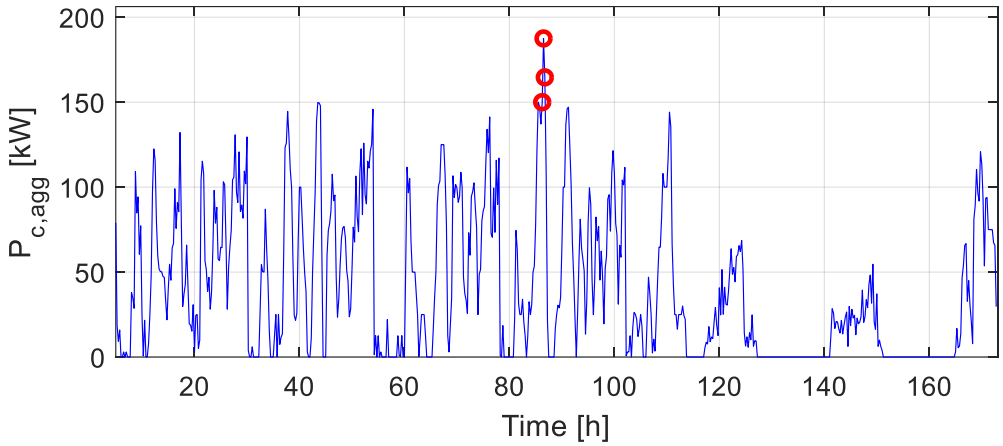


Fig. 50 Aggregate charging power obtained by DP optimization of individual vehicle profiles and aggregating them by using (5.13) (no RES considered; red circles denote points where aggregate charging power exceeds imposed maximum grid power of 150 kW)

Table 15 provides the DP results for each EV separately and the corresponding lump sum results. The lump sum results from Table 15 are included in Table 16 along with the results related to other charging approaches. All methods have very similar fuel consumptions and cumulative charging energies, thus making their charging costs directly comparable. The charging costs of DP-OFF and MPC-REC turns out to be only up to 2% higher than the DP-IND costs, while that of MPC-DIM is even 0.8% lower, confirming that they are all close to the feasible global optimum, despite the fact that they significantly lag the charging costs obtained when applied at the aggregate battery model level (Table 14). On the other hand, when applying DUMB charging, the charging cost becomes higher than the DP-IND cost by 9%. The fact that MPC-DIM provides even lower cost than DP-IND indicates that there is still some room for improvement of DP-IND charging results via using finer quantization of SoE and charging power within DP optimization.

Table 15 Optimization results for each EV obtained by separate DP optimizations (DP-IND; no RES considered)

EV #	Initial SoC [-]	Final SoC [-]	Fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]
1	0.95	0.95	839.4	6884.4	51.9
2	0.95	0.95	303.0	738.6	57.7
3	0.95	0.95	479.6	832.4	65.9
4	0.95	0.95	391.7	813.9	65.5
5	0.95	0.95	518.9	617.0	49.8
6	0.95	0.95	305.2	568.1	46.8
7	0.95	0.95	201.6	583.6	52.1
8	0.95	0.31	587.3	767.0	59.5
9	0.95	0.95	454.0	888.5	68.5
10	0.95	0.95	179.3	446.0	37.2
∑	**9.50**	**8.86**	**4260.0**	**6884.4**	**554.9**

Table 16 Comparative performance metrics related to results obtained by applying different approaches to distributed model (no RES included)

One week period, DISTR model	Total fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]	Specific cost of charging [EUR/kWh]
DP-IND	4260.0 (0.0%)	6884.4 (0.0%)	554.9 (0.0%)	0.0806 (0.0%)
DP-OFF	4266.6 (+0.2%)	6830.6 (-0.8%)	565.3 (+1.9%)	0.0828 (+2.7%)
DUMB	4273.4 (+0.3%)	6894.0 (+0.1%)	605.1 (+9.1%)	0.0878 (+8.9%)
MPC-REC	4267.8 (+0.2%)	6906.1 (+0.3%)	564.1 (+1.7%)	0.0817 (+1.4%)
MPC-DIM	4267.8 (+0.2%)	6906.1 (+0.3%)	550.6 (-0.8%)	0.0797 (-1.1%)

5.5.3 Results for case of RES consideration

When including the power production from RES, the DP-OFF optimization tends to shift charging closer to solar noon where the RES production is around its peak (see Fig. 51). It is interesting to note that the (aggregate) battery is not fully charged at 5 a.m. unlike the case when no RES production is included. This can be explained by the fact that the optimizer leaves the battery at lower SoE value to charge it when RES power is available (note that the local-RES power price is set to zero).

Table 17 gives the charging results for the case of aggregate model (see Fig. 44), where it can be observed that the cost of DUMB approach, and even MPC-DIM, is now multiple times higher than that of DP-OFF. MPC-DIM performs much worse than in the case when no RES was included (cf. Table 13) because of the requirement on the SoE to be equal 0.95 at 5 a.m. of each day, which significantly limits the optimisation freedom in the presence of RES charging potential later in the day. Since MPC-REC does not involve this constraint, its performance does not degrade; in fact, it provides (almost) the same results as DP-OFF.

Table 18 gives comparative results obtained by applying different charging methods for the cases of using aggregate (AGG) and distributed model (DISTR). The fuel consumptions and charging energies are similar in all cases, while the increase in the charging cost when the distributed model is used is more pronounced than in the case of no RES considered (cf. Tables 18 and 14). The only exception is DUMB strategy, whose charging cost is very similar in the case of both scenarios, but it is very high when compared to other charging methods (see also Table 20). These results can be explained by the fact that the relatively narrow RES power production peaks mostly occur around the hours of elevated EVs activity (cf. Figs. 38 and 43), i.e., when EVs are typically less available for charging. For this reason, perturbation of charging power profiles, caused by aggregate charging power distribution to individual EVs, results in decrease of employed RES energy and leads to significantly increased costs.

Table 17 Optimization and simulation results obtained for case of aggregate model and RES considered

Aggregate model, one week	Initial SoC [-]	Final SoC [-]	Fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]
DP-OFF	0.95	0.95	4259.8	6935.0 (0.0%)	99.0 (0.0%)
DUMB	0.95	0.95		6935.0 (0.0%)	324.4 (+228)
MPC-REC	0.95	0.95		6935.0 (0.0%)	99.0 (0.0%)
MPC-DIM	0.95	0.95		6935.0 (0.0%)	244.5 (+147%)

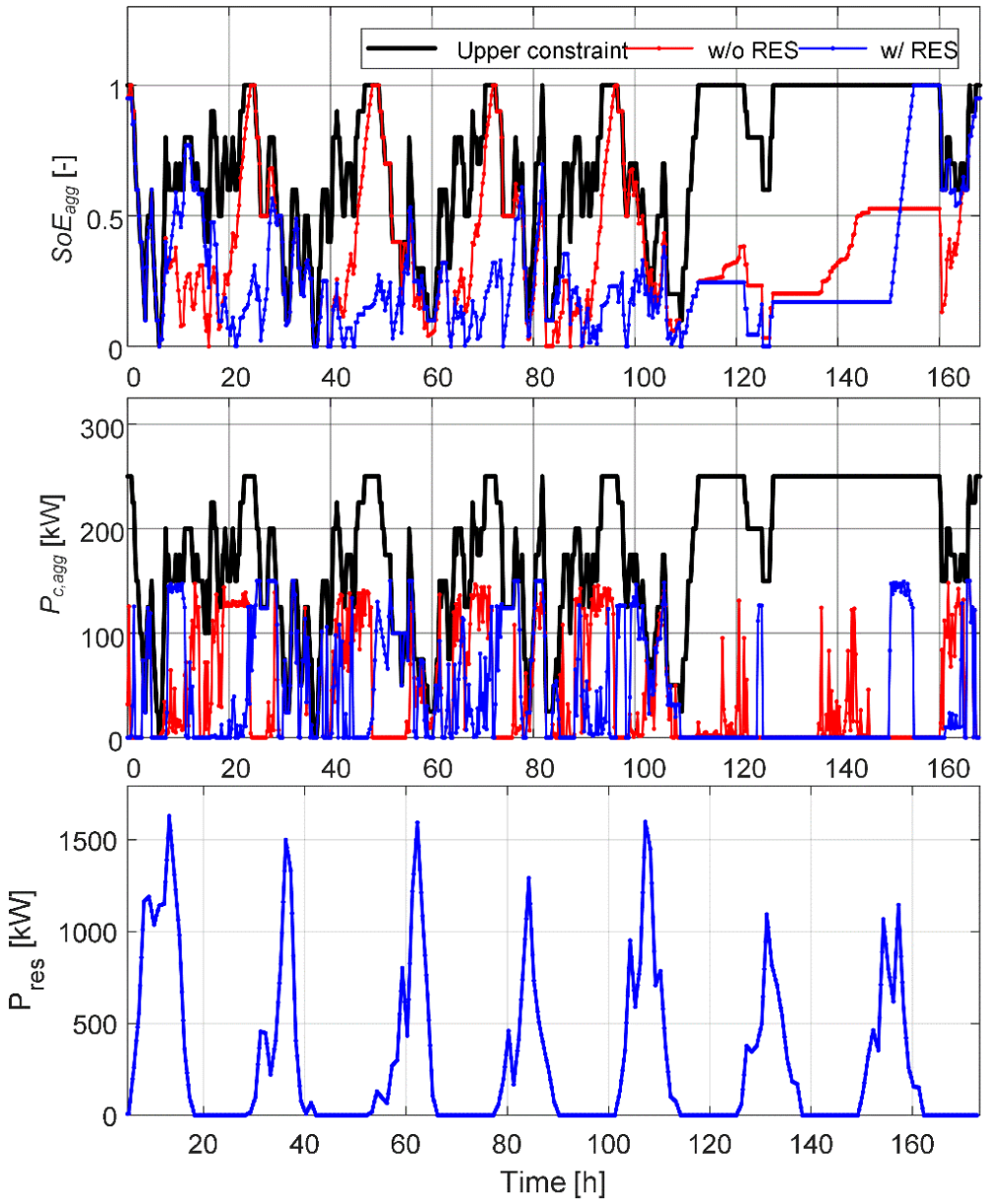


Fig. 51 Comparative DP-OFF optimization results for the cases with and without RES production

Table 18 Comparative performance metrics related to results obtained by using aggregate model and distributed model (RES is considered)

One week period	EV fleet model	Initial SoC [-]	Final SoC [-]	Total fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]
DP-OFF	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	99.0 (0.0%)
	DISTR	0.95	0.85	4262.4 (+0.1%)	6845.5 (-1.3%)	193.2 (+95.2%)
DUMB	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	324.4 (0.0%)
	DISTR	0.95	0.96	4273.4 (+0.3%)	6894.0 (-0.6%)	321.1 (-1.0%)
MPC-REC	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	99.0 (0.0%)
	DISTR	0.95	0.95	4262.4 (+0.06%)	6925.6 (-0.1%)	176.1 (+77.9%)
MPC-DIM	AGG	0.95	0.95	4259.8 (0.0%)	6935.0 (0.0%)	244.5 (0.0%)
	DISTR	0.95	0.95	4262.4 (+0.1%)	6925.6 (-0.1%)	276.8 (+13.2%)

Similarly, as in the case of not using RES, the charging optimization of individual EVs is performed by DP and the distributed model (5.5) (again, denoted as DP-IND), to establish a kind of direct benchmark. The results are presented in Table 19. Implications of not using joint constraint (5.4b) on the aggregate charging power become more pronounced in the no-RES case due to neglecting RES power profile as a shared resource of all EVs (i.e., each EV is set to have the whole RES power profile at its disposal, which is not realistic). The aggregate charging power (Fig. 52), obtained through aggregation of the individual DP-IND profiles, indicates that the aggregate charging power limit (5.4b) is now violated in more time steps than in the case of not using RES (cf. Fig. 50). Nevertheless, the DP-IND results still provide certain orientation on the globally optimal charging cost, and they are, thus, further used in the comparative analyses.

Table 19 Optimization results for each EV obtained by separate DP optimizations (DP-IND; RES is considered)

EV #	Initial SoC [-]	Final SoC [-]	Fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]
1	0.95	0.95	839.4	629.3	10.3
2	0.95	0.95	303.0	738.7	24.9
3	0.95	0.95	479.6	832.4	29.6
4	0.95	0.95	391.7	813.9	26.3
5	0.95	0.95	518.9	617.1	12.2
6	0.95	0.95	305.2	568.2	6.1
7	0.95	0.95	201.6	583.7	13.4
8	0.95	0.31	587.3	767.0	28.2
9	0.95	0.95	454.0	888.5	27.9
10	0.95	0.95	179.3	446.1	5.1
∑	**9.50**	**88.6**	**4260.0**	**6884.9**	**184.0**

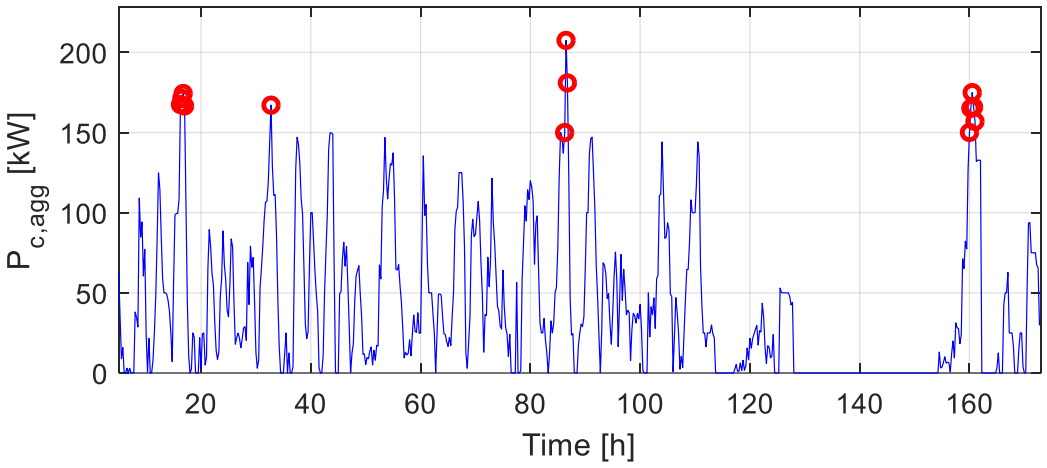


Fig. 52 Aggregate charging power obtained by DP optimization of individual profiles and aggregating them by using Eq. (5.13) for case of RES included (red circles denote points where aggregate charging power exceeds imposed maximum grid power of 150 kW)

The aggregated DP-IND results from Table 19 are used in Table 20 as the benchmark for other charging methods applied on the same distributed model (the results taken from Table 18). The DP-OFF approach gives around 6% higher charging cost than the DP-IND one, which may be attributed to the more constrained optimization in the case of DP-OFF and the suboptimality of the distribution algorithm. MPC-DIM and DUMB methods have again very high charging costs, which is due to the requirement on the final SoE set for each day in the former case, and the algorithm insufficiency (in terms of not reflecting the RES potential) in the latter case. The MPC-REC strategy does not involve the final SoE condition and, thus, significantly outperforms the MPC-DIM method. Furthermore, by incorporating feedback via its online execution, it mitigates the aggregate model deficiencies and, thus, achieves lower cost when compared to DP-OFF method, whose aggregate charging power distribution is performed offline. The fact that MPC-REC, actually, provides 5% lower charging cost when compared to DP-IND again indicates certain room for improvement of DP-IND results via finer quantization of SoE and charging power. Indeed, reducing the quantization step of charging power $P_{c,i}$ from 500 W to 100 W turns out to result in 4.7% cost reduction (from 184.0 to 175.3 EUR), which is now slightly lower than MPC-REC cost.

Table 20 Comparative performance metrics related to results obtained by applying different approaches to distributed model (RES is included)

One week period, DISTR model	Total fuel consumption [L]	Total charging energy [kWh]	Total cost of charging [EUR]	Specific cost of charging [EUR/kWh]
DP-IND	4260.0 (+0.0%)	6884.9 (+0.0%)	184.0 (+0.0%)	0.0267 (+0.0%)
DP-OFF	4262.4 (+0.1%)	6845.5 (-0.6%)	193.2 (+5.0%)	0.0282 (+5.6%)
DUMB	4273.4 (+0.3%)	6894.0 (+0.1%)	321.1 (+74.5%)	0.0466 (+74.5%)
MPC-REC	4262.4 (+0.1%)	6925.6 (+0.6%)	176.1 (-4.3%)	0.0254 (-4.9%)
MPC-DIM	4262.4 (+0.1%)	6925.6 (+0.6%)	276.8 (+50.4%)	0.0400 (+49.8%)

5.6 Short conclusion

An offline optimization tool for EV fleet charging has been first developed based on the dynamic programming (DP) algorithm to set a performance benchmark. An online hierarchical EV charging management method has then been proposed to optimize the aggregate charging power profile by means of a model predictive control (MPC) algorithm and distribute this profile over individual EVs

by using a heuristic allocation algorithm based on charging priorities. The main benefit of the proposed method is that it can be applied to large-size EV fleets, while providing a nearly optimal solution.

The effectiveness of the proposed charging method has been demonstrated through a delivery electric vehicle fleet study, where a Extended Range Electric Vehicle (EREV) is concerned, as it can cover all driving missions involved (short- and long-distance ones). It has been demonstrated that both MPC strategies considered (with receding horizon and shrinking horizon) provide almost the same results on the aggregate battery level when compared to the offline DP benchmark for the case of no power production from renewable energy sources (RES), while the cost is around 17% lower when compared to the baseline (dumb) strategy involving charging EV fleet with a maximum power when possible. When used in combination with the heuristic allocation algorithm within the more realistic distributed vehicle fleet model, both MPC strategies result in charging costs that are close to the DP benchmark obtained on the distributed EV fleet model. At the same time, unlike the offline-applied (open-loop) distribution algorithm, the MPC strategies satisfy the target aggregate battery state-of-energy (SoE) owing to its feedback control character. When including the production from RES, the MPC variant based on receding horizon optimization overperforms its shrinking-horizon counterpart by the large margin. This is because the latter is overly restricted by the battery SoE constraint at the end of diminishing horizon for each operating day, thus not allowing for full exploitation of the RES potential.

In the remaining course of WP2.1 activity, the developed receding horizon-based MPC strategy will be accommodated and demonstrated within the airport e-hub planning case study.

6 E-bus scheduling optimization

6.1 Introduction

Solving an electric bus scheduling problem yields optimal scheduling of electric buses to minimize the fleet size (i.e., the total number of e-buses), while satisfying the predetermined service trips and timetables, and accounting for the e-bus range and charging restrictions. The scheduling involves optimal allocation of electric buses to service trips and the determination of when and where each bus should be charged.

This section presents a thorough approach to e-bus scheduling optimization, which results in a Pareto frontier in two conflicting criteria being minimized: (i) the total number of buses required to serve the predetermined routes and (ii) the excess of distance travelled (so-called deadhead distance). These criteria reflect the city bus fleet investment and operational costs, respectively. The 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 (MILP) and genetic algorithm (GA), 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 underlying MILP formulation has been adopted from [16], and then extended by adding constraints to ensure the buses are fully charged by the end of each day (charge sustaining condition), while locally considering the state of energy for each bus, the rated power of individual chargers, and the specific number of buses that can be charged at each station, thus underscoring the non-uniformity across buses and chargers. The optimization approach is demonstrated on a custom-generated dataset reflecting characteristics of real-world city bus transport systems and is implemented in Python programming language.

The subsequent subsections are organized as follows. Subsection 6.1 introduces the e-bus scheduling framework and formally defines the scheduling problem. Subsection 6.2 presents the MILP formulation, while Subsection 6.3 explores the GA approach. Subsection 6.4 discusses the optimization results and compares the MILP and GA results. Concluding remarks are given in Subsection 6.5.

Note: The work presented in this section has been disseminated through the following conference papers, which also include a methodology state-of-the-art review and elaborates on the contributions of the approach proposed:

Z. Dabčević, B. Škugor, J. Deur, "Pareto Optimization of Electric City Bus Scheduling", 18th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Dubrovnik, Croatia, 2023.

6.2 Problem definition

6.2.1 Electric city bus scheduling framework

Electric bus scheduling poses several challenges, particularly when buses operate on partial charges and require charging at both main depots and designated route stops (end stations). Moreover, 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. 53 illustrates the developed 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 minimum value.

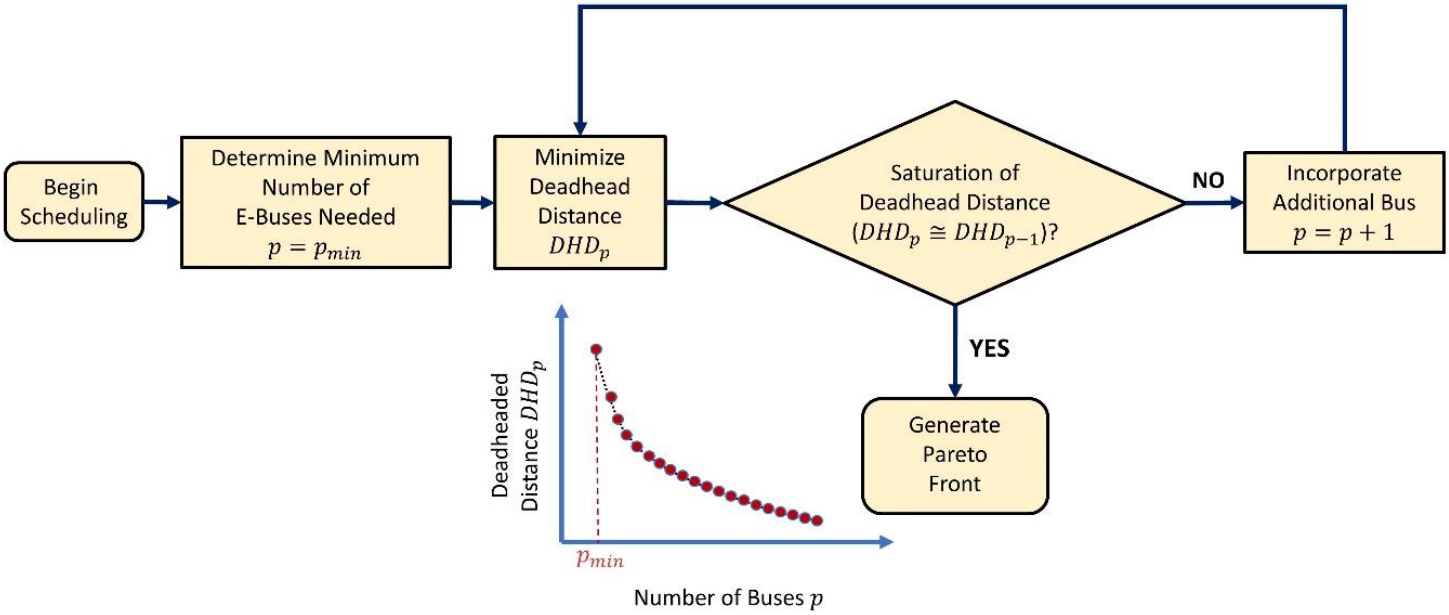


Fig. 53 Flowchart of e-bus scheduling sequential optimization process

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).

6.2.2 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 to the depot. Each service trip, denoted by 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: $s_j + t_j + t_{ij} \leq s_i$.

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 that a charging station can handle at once, based on the 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 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 the trip i has reached the charger r: $s_{rt} \geq s_i + t_i + t_{ri}$,
- $B_c(r,i)$ denotes charging events that occur before the trip i reaches the charger r,
- $F_i(r,t)$ indicates trips starting after the charging event t: $s_i \geq s_{rt} + t_{ir}$,
- $B_i(r,t)$ captures trips ending before the charging event t at the 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. 54 visualizes the role of above decision variables and the overall scheduling mechanism.

*

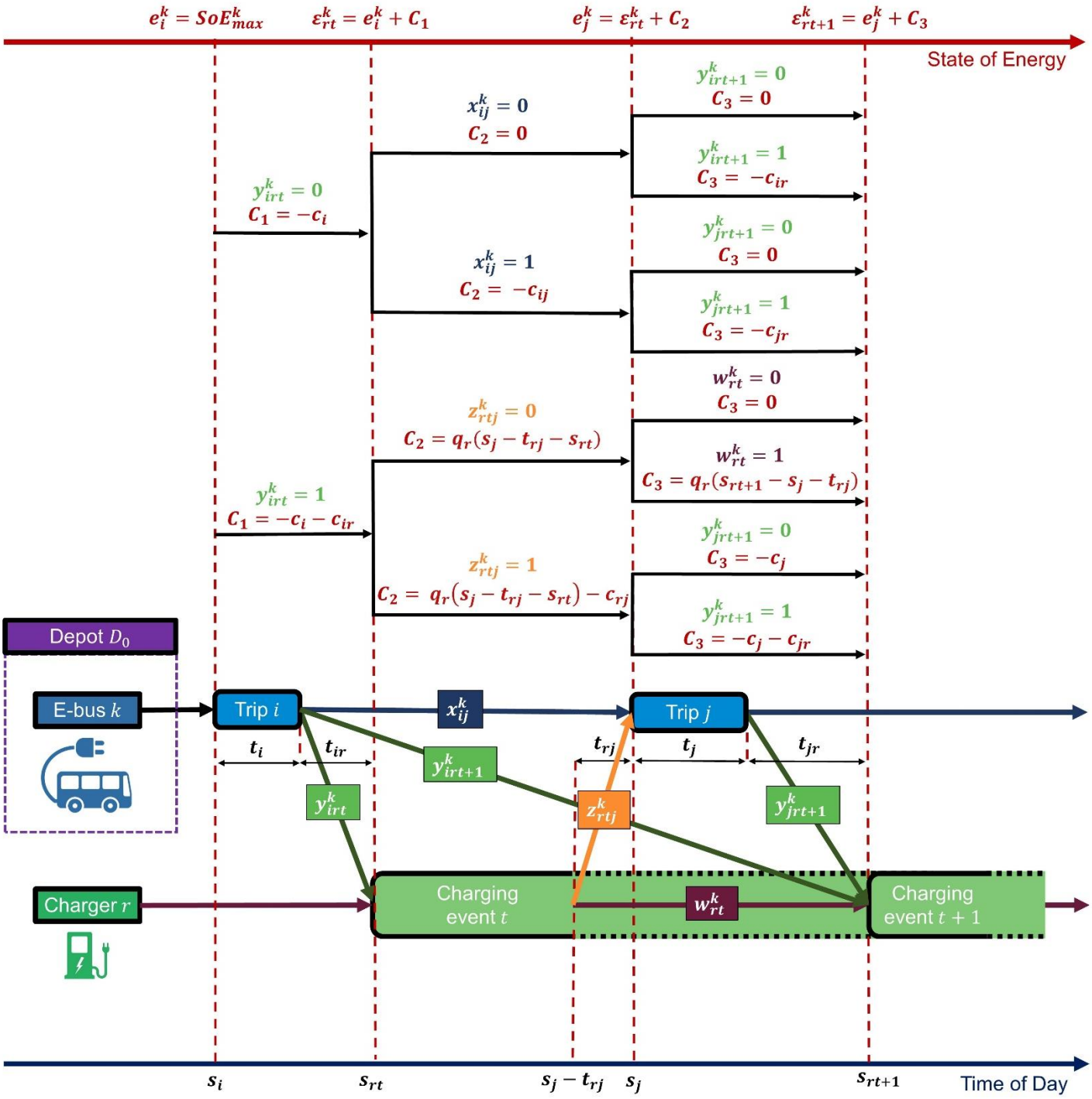


Fig. 54 Visualization of vehicle scheduling problem formulation

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.

6.3 Mixed-integer linear programming formulation

By utilizing mathematical optimization based on the Mixed Integer Linear Programming (MILP) algorithm, a structured approach for solving the bus scheduling problem defined in Subsection 6.2 and Fig. 54 is proposed, which yields Pareto optimal solution in terms of minimization of the total number of buses and the 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 [17]. In this study, the coin-or branch and cut solver, accessible via the PuLP library in Python is utilized to solve the MILP formulation.

6.3.1 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. 53). The total number of buses in the system is determined by those dispatched from the depot, from which the buses are assumed to be exclusively launched (this does not restrict buses from shifting between lines). Therefore, the corresponding objective function counts all the trips j of buses k from the depot D_0, and is formulated as:

$$\min \sum_{k \in K} \sum_{j \in N} x_{D_0 j}^k, \tag{6.1}$$

The second objective relates to minimization of the total deadhead distance (see the third block in Fig. 53), which sums the distances the buses travel outside of regular service. More specifically, 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:

$$\min \sum_{k \in K} \sum_{i \in N} \sum_{j \in F(i)} d_{ij} x_{ij}^k + \sum_{k \in K} \sum_{i \in N} \sum_{t \in F_c(r,i)} d_{ir} y_{irt}^k + \sum_{k \in K} \sum_{r \in R} \sum_{t \in r} \sum_{j \in F_i(r,t)} d_{rj} z_{rtj}^k, \tag{6.2}$$

6.3.2 Vehicle scheduling constraints

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

$$\sum_{k\in K}\sum_{i\in B(j)} x_{ij}^{k} + \sum_{k\in K}\sum_{r\in R}\sum_{t\in B_c(r,j)} z_{rtj}^{k} = 1;\ \forall j \in N. \tag{6.3}$$

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:

$$\sum_{i\in B(j)} x_{ij}^{k} + \sum_{r\in R}\sum_{t\in B_c(r,j)} z_{rtj}^{k} = \sum_{l\in F(j)} x_{jl}^{k} + \sum_{r\in R}\sum_{t\in F_c(r,j)} y_{jrt}^{k};\ \forall j \in N, \forall k \in K. \tag{6.4}$$

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:

$$\sum_{k\in K}\sum_{j\in B_i(r,t)} y_{jrt}^{k} + \sum_{k\in K} w_{rt-1}^{k} \le N_r;\ \forall r \in R, \forall t \in T^r. \tag{6.5}$$

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

$$\sum_{i\in B_i(r,t)} y_{irt}^{k} + w_{rt-1}^{k} = \sum_{j\in F_i(r,t)} z_{rtj}^{k} + w_{rt}^{k};\ \forall r \in R, \forall t \in T^r, \forall k \in K. \tag{6.6}$$

To ensure that the total number of deployed buses matches the given fleet size p in the case of deadhead distance minimization step (Fig. 53), the following constraint is introduced:

$$\sum_{k\in K}\sum_{j\in N} x_{D_0 j}^{k} = p. \tag{6.7}$$

6.3.3 Energy consumption constraints

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

$$e_{D_0}^{k} = SoE_{max}^{k};\ \forall k \in K. \tag{6.8}$$

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:

$$e_i^k \geq SoE_{min}^k + c_i + \sum_{j \in F(i)} x_{ij}^k c_{ij} + \sum_{r \in R} \sum_{t \in F_c(r,i)} y_{irt}^k c_{ir}\,; \forall i \in N, \forall k \in K. \quad (6.9)$$

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:

$$e_j^k \leq e_i^k - x_{ij}^k(c_i + c_{ij}) + SoE_{max}^k(1 - x_{ij}^k); \forall j \in N, \forall i \in B(j), \forall k \in K, \quad (6.10)$$

$$e_j^k \geq e_i^k - x_{ij}^k(c_i + c_{ij}) - SoE_{max}^k(1 - x_{ij}^k); \forall j \in N, \forall i \in B(j), \forall k \in K. \quad (6.11)$$

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 next trip start:

$$e_j^k \leq \varepsilon_{rt}^k + z_{rtj}^k\left((s_j - t_{rj} - s_{rt})q_r - c_{rj}\right) + SoE_{max}^k(1 - z_{rtj}^k); \forall j \in N, \forall r \in R, \forall t \in B_c(r,j), \forall k \in K. \quad (6.12)$$

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):

$$SoE_{max}^k \geq e_j^k + c_{rj} - Mq_r(1 - z_{rtj}^k); \forall r \in R, \forall t \in T^r, \forall k \in K, \forall j \in F_i(r,t). \quad (6.13)$$

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, (the M-term offers flexibility if the bus does not advance to its next charge):

$$SoE_{max}^k \geq \varepsilon_{rt+1}^k - Mq_r(1 - w_{rt}^k); \forall r \in R, \forall t \in T^r, \forall k \in K. \quad (6.14)$$

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

$$e_j^k + c_{rj} + Mq_r(1 - z_{rtj}^k) \geq SoE_{min}^k + z_{rtj}^k c_{rj};\ \forall r \in R, \forall t \in T^r, \forall k \in K, \forall j \in F_i(r,t). \quad (6.15)$$

The following two equations are ensured by the preservation of energy between the service trip and the charger, where the next charging occurs:

$$\varepsilon_{rt}^k \leq e_i^k - y_{irt}^k(c_i + c_{ir}) + SoE_{max}^k(1 - y_{irt}^k); \forall r \in R, \forall t \in T^r, \forall k \in K, \forall i \in B_i(r,t). \quad (6.16)$$

$$\varepsilon_{rt}^{k} \geq e_{i}^{k} - y_{irt}^{k}(c_i + c_{ir}) - SoE_{max}^{k}\left(1 - y_{irt}^{k}\right); \forall r \in R, \forall t \in T^{r}, \forall k \in K, \forall i \in B_i(r,t). \quad (6.17)$$

Furthermore, the following constraint delineates the energy that can be charged during an event, accounting for the time gap between consecutive charging events:

$$\varepsilon_{rt+1}^{k} \leq \varepsilon_{rt}^{k} + w_{rt}^{k}(s_{rt+1} - s_{rt})q_r + SoE_{max}^{k}\left(1 - w_{rt}^{k}\right); \forall r \in R, \forall t \in T^{r}, \forall k \in K. \quad (6.18)$$

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.

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:

$$e_{j}^{k} + c_{rj} - \varepsilon_{rt}^{k} - SoE_{max}^{k}\left(1 - z_{rtj}^{k}\right) \leq (s_{rt+1} - s_{rt})q_r; \forall r \in R, \forall t \in T^{r}, \forall k \in K, \forall j \in F_i(r,t). \quad (6.19)$$

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.

$$e_{j}^{k} + c_{rj} - \varepsilon_{rt}^{k} + SoE_{max}^{k}\left(1 - z_{rtj}^{k}\right) \geq 0; \forall r \in R, \forall t \in T^{r}, \forall k \in K, \forall j \in F_i(r,t), \quad (6.20)$$

$$\varepsilon_{rt+1}^{k} - \varepsilon_{rt}^{k} + SoE_{max}^{k}\left(1 - w_{rt}^{k}\right) \geq 0; \forall r \in R, \forall t \in T^{r}, \forall k \in K. \quad (6.21)$$

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:

$$\sum_{r \in R} \sum_{t \in T^{r}} z_{rtD_n}^{k} = 1; \ \forall k \in K. \quad (6.22)$$

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

$$\varepsilon_{rt}^{k} + \left(s_{rt+1}^{k} - s_{rt}^{k}\right)q_r \geq SoE_{max}^{k} - M(1 - z_{rtD_n}^{k}), \ \forall r \in R, \forall t \in T^{r}, \forall k \in K \quad (6.23)$$

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:

$$s_{rt+1}^{k} \leq s_{j}^{k} + t_{rj} + 1440 + M(1 - z_{rtD_n}^{k}), \ \forall r \in R, \forall t \in T^{r}, \forall j \in N, \forall k \in K. \quad (6.24)$$

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

6.3.4 Domain constraints

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

$$x_{ij}^{k} \in \{0,1\}; \forall k \in K, \forall i \in N \cup D_0, \forall j \in F(i) \cup D_n, \tag{6.25}$$

$$z_{rtj}^{k} \in \{0,1\}; \forall k \in K, \forall r \in R, \forall t \in T^r, \forall j \in F_i(r,t), \tag{6.26}$$

$$y_{irt}^{k} \in \{0,1\}; \forall k \in K, \forall i \in N, \forall r \in R, \forall t \in F_c(r,i), \tag{6.27}$$

$$w_{rt}^{k} \in \{0,1\}; \forall k \in K, \forall r \in R, \forall t \in T^r, \tag{6.28}$$

$$\varepsilon_{rt}^{k} \geq 0; \forall k \in K, \forall r \in R, \forall t \in T^r, \tag{6.29}$$

$$e_{i}^{k} \geq 0; \forall k \in K, \forall i \in N. \tag{6.30}$$

6.4 Genetic algorithms approach

Genetic algorithms (GA), inspired by biological evolution, offer a unique and general approach to optimization by simulating natural selection [18]. 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 that 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.

The initial population of the GA has been formed by solving a relaxed MILP problem for 14 sub-formulations. All 14 sub-formulations utilize either objective function (6.1) or (6.2), depending on the optimization phase (see Fig. 53), while adhering to the vehicle scheduling constraints (6.3)-(6.7) and the domain constraints (6.25)-(6.30). Between the remaining constraints (6.8)-(6.24), 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.

Algorithm 4: 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 soc_penalty if soc is negative
                    IF soe < 0:
                        soe_penalty += ABS(soe)
            # Adjust soe_penalty if the final soc 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 employed the same solution representation as in the case of MILP formulation, where the binary decision variables x_{ij}^k, z_{rtj}^k, y_{irt}^k, and w_{rt}^k are optimized to obtain the final solution (see Fig 54). To ensure feasibility and optimality, the GA fitness function has been carefully designed (see

Algorithm 4). It assigns a lower value to solutions that use more vehicles and correspond to more constraint violations.

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.

6.5 Optimization results

6.5.1 Scenario generation and data description

A detailed system scenario has been developed to replicate the complexity of a city bus transport system [19]. For the purpose of verifying the MILP optimization algorithm (Subsection 6.5.2), 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 (Subsection 6.5.3) 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.

6.5.2 MILP optimization

The MILP methodology depicted in Fig. 53 and elaborated in Subsection 6.3 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 (6.1) and (6.2), and the constraints (6.3), (6.4), and (6.7), 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 Subsection 6.3 is used.

Fig. 55 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 almost 550 km (Fig. 55a) 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; Fig. 55b). 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. 55a) with the deadhead distance being reduced to some extent (from 20% to 17.5%, Fig. 55b), and the Pareto frontier generally shifts to higher values of the two objectives. However, as the number of electric buses increases (to 22), the Pareto frontier approaches that of the conventional fleet. This is because the charging system is efficient enough not to disturb the bus scheduling if the bus fleet is large enough.

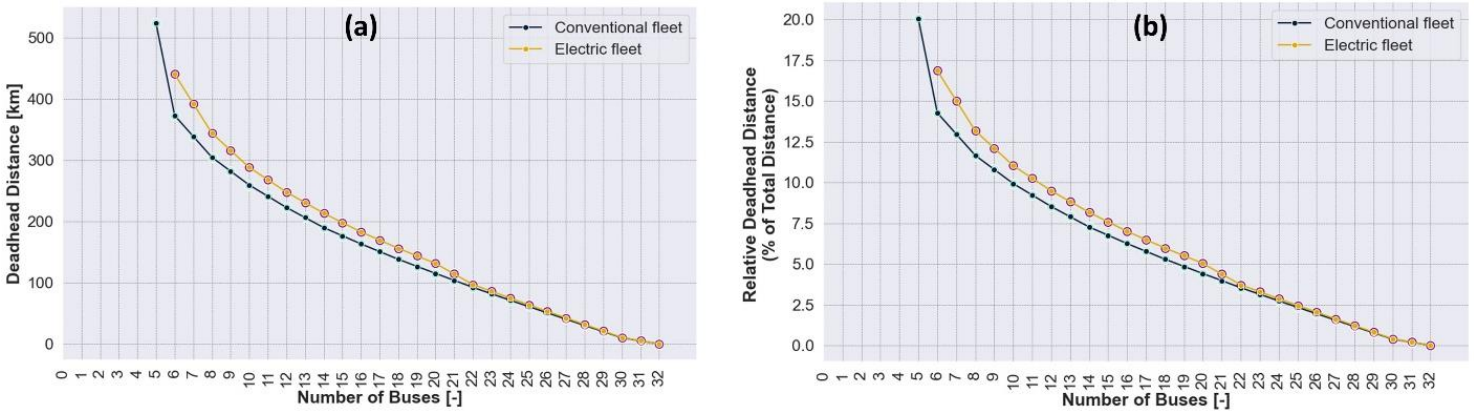


Fig. 55 Comparative Pareto frontiers obtained by MILP approach in the case of conventional and electric bus scheduling optimization

6.5.3 Comparative analysis of MILP and GA optimization results

A comparative analysis of the MILP and GA optimization results is presented in Fig. 56 for the case of minimizing only the total number of buses criterion (6.1). 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. 56).

Fig. 56 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 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. 56. 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. 56), 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 the large-size conventional fleet. Hence, the GA is deemed to be a suitable choice for large-scale e-bus transport systems.

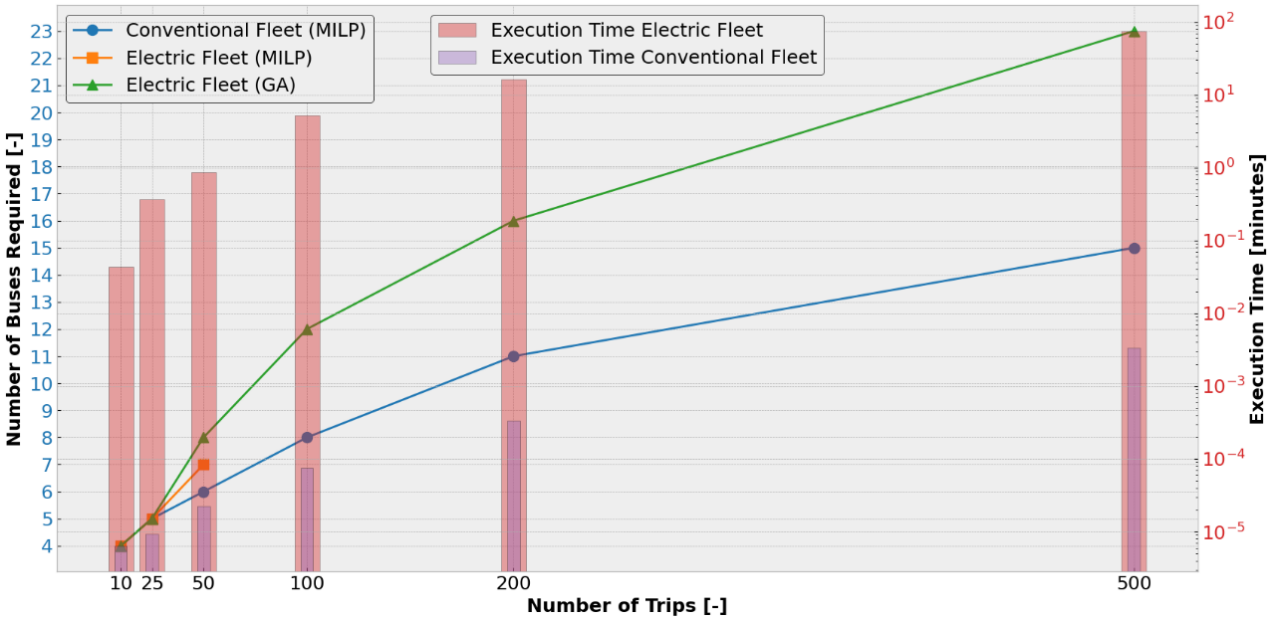


Fig. 56 Comparison of minimal numbers of buses obtained by using MILP and GA approaches for various sizes of city bus transport system

6.6 Short conclusion

A multi-objective e-bus scheduling optimization approach has been proposed based on the Mixed Integer Linear Programming (MILP) problem formulation. The strategy seeks to minimize not only the e-bus fleet size, but also the deadhead distance, thus offering a spectrum of Pareto optimal solutions as a trade-off between investment and operating costs. The problem formulation accounts for charge sustaining condition, integrated charging at depot, and inherent variability of buses and charger parameters.

Using a MILP solver offers the optimal solution, but its computational inefficiency restricts it to small-size transport systems. The results indicate that, when compared to the conventional e-bus transport system, the e-bus system faces performance reductions due to vehicle range and charging

restrictions. This gap eventually narrows with increasing the fleet size, because the range and charging restrictions become less relevant for the expanded fleet size.

To address scalability, the problem has also been solved by using a genetic algorithm (GA). The GA solution matches the MILP solution for small-scale systems, and at the same time offers feasible near-optimal solutions for large-scale systems, thus proving its suitability for large e-bus fleets' scheduling scenarios.

7 Conclusion and outlook

The previously developed SOLEZ project software application for city bus transport electrification planning has been modified and extended through this deliverable to expand its applicability (including airport landside and e-hub systems), provide automated optimization features, and increase its accuracy. The newly developed software solutions are organized around four characteristic modules:

1) **A trip-based data-driven e-bus model** has been developed to substantially reduce the virtual transport system simulation execution time and the transport data demand. The former makes the mid/large-scale virtual simulation and optimization studies feasible, while the latter allows application to typical situations where only low-resolution recorded driving cycle data are available either through bus transport system tracking or planning. The data-driven e-bus model has been parameterized and tested for a wide range of driving cycles and scenarios, and it has been demonstrated that its accuracy approaches that of the experimentally validated physical e-bus model.
2) **A charging configuration optimization framework** has been established and solved by a multi-objective genetic algorithm (GA). The optimization results in a Pareto frontier in three objectives being minimized: the number of charging terminals, the total number of chargers, and the total (cumulative) trip service delay caused by e-bus range and charging constraints. This module provides an automated and optimal design of the charging system, thus minimizing the investment costs while penalizing for the service delay. The designer simply selects a convenient point from the Pareto optimal frontier based on his/her expert knowledge, rather than nominating multiple (and still limited) charging configurations and running virtual simulations in a manual way.
3) **A model predictive control** (MPC) **strategy for optimal online charging management** has been developed along with an offline tool for globally optimal offline charging management optimization. A custom-made dynamic programming (DP) algorithm has been used in both tools to provide globally optimal solution for a general (nonlinear) problem formulation. The

main objective is to minimize the charging electricity cost, partly by boosting the share of electricity produced from local renewable sources. The MPC strategy is run on the simplified aggregate battery level to facilitate its application to large-scale systems such as e-hubs. The MPC-commanded aggregate charging power its then distributed to individual vehicles by using a heuristic algorithm based on charging priorities of individual vehicles. It has been demonstrated through a realistic case study that the hierarchical MPC strategy can approach the offline optimization benchmark to a narrow margin. The strategy can be used to make the e-hub planning studies more realistic, as well as in on-line charging management applications.

4) **An e-bus scheduling optimization algorithm** has been designed based on the mixed integer linear problem (MILP) formulation and MILP and GA solver alternatives. The objectives are to minimize the e-bus fleet size (i.e., the total number of buses) and the service deadhead distance (i.e., the extra miles to switch between lines), while satisfying the predetermined e-bus lines, timetables, energy demand, and charging constraints. This results in a Pareto frontier in the two objectives, from which the designer can readily choose a point which satisfies the transport system practical constraints at best. The main benefit of the developed scheduling solution includes exploiting the e-bus re-scheduling opportunity in a strictly optimal way to mitigate the e-bus range- and charging-related restrictions and, thus, minimize the fleet investment cost, while leveraging the exploitation cost in terms of additional mileage and, correspondingly, energy consumption.

In the remaining course of WP2.1 to be resulted in D2.2 by M54, the developed software solutions will be exploited to optimally design the airport-city (and intra-airport) e-bus transport system and the airport e-hub system based on real (recorded) transport and energy system data related to Zagreb airport and Paris-Charles de Gaulle airport.

8 References

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

[2] L. Guzzella, A. Sciaretta, Vehicle Propulsion Systems, 2nd ed., Springer Verlag, Berlin, 2007.

[3] J. Soldo, B. Škugor, and J. Deur, “Synthesis of Optimal Battery State-of-Charge Trajectory for Blended Regime of Plug-in Hybrid Electric Vehicles in the Presence of Low-Emission Zones and Varying Road Grades”, Energies, Vol. 12, No. 22, pp. 21, 2019.

[4] ..., EVO ELECTRIC LTD CATALOGUE, AF-230 Motor/Generator. Available online: http://www.fordmax.in.ua/wp-content/uploads/2014/12/AF-230-Spec-Sheet-V1.pdf (accessed on 6 October 2019).

[5] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth, D. U. Sauer, "Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. II: Modelling", Journal of Power Sources, Vol. 196, No. 12, pp. 5334-5341, 2011.

[6] ..., AC136 AE HP (All-Electric / Heat Pump): Eberspächer Available online: https://www.eberspaecher.com/produkte/air-conditioning/thermomanagement-bus/produkte/busse-ueber-12-m-klimaanlage-eberspaecher/ac136-ae-hp-all-electricheat-pump (accessed on 6 October 2019).

[7] Z. Dabčević, B. Škugor, J. Topić, J. Deur, "Synthesis of Driving Cycles Based on Low-Sampling-Rate Vehicle-Tracking Data and Markov Chain Methodology", Energies, Vol. 15, No. 11, pp. 21, 2022.

[8] D. Freedman, R. Pisani, R. Purves, "Statistics", 3rd edition, W.W. Norton, New York, 1998.

[9] U. Stańczyk, L.C. Jain, "Feature Selection for Data and Pattern Recognition", Springer, Berlin Heidelberg, 2014.

[10] Y. Dodge, "Analysis of Residuals", Springer, New York, NY, 2008.

[11] V. V. Vazirani, "Approximation algorithms", Springer, New York, Ny, 2001.

[12] B. Škugor, J. Deur, “A Novel Model of Electric Vehicle Fleet Aggregate Battery for Energy Planning Studies", Energy, Vol. 92, pp. 444-455, 2015.

[13] B. Škugor, J. Deur, “Dynamic Programming-based Optimization of Charging an Electric Vehicle Fleet System Represented by an Aggregate Battery Model”, Energy, Vol. 92, pp. 456-466, 2015.

[14] B. Škugor, J. Deur, “Delivery vehicle fleet data collection, analysis and naturalistic driving cycles synthesis”, Int. J. Innovation and Sustainable Development, Vol. 10, No. 1, pp. 19-39, 2016.

[15] B. Škugor, J. Deur, “A bi-level optimisation framework for electric vehicle fleet charging management”, Applied Energy, Vol. 184, pp. 1332-1342, 2016.

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

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

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

[19] 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.

[20] B. Škugor, J. Deur, “Analysis of Techno-Economic Aspects of an Energy System Including Delivery Electric Vehicle Fleet and Renewable Energy Sources”, 10th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES), Dubrovnik, Croatia, 2015.