Les Cahiers du GERAD ISSN: 0711–2440

Adaptive real-time optimization for electric bus depot operations with state-of-charge tracking signal

S. Khalilpourazari, R. Barra, G. Desaulniers, J. E. Mendoza

G–2026–41

July 2026

Citation suggérée : S. Khalilpourazari, R. Barra, G. Desaulniers, J. E. Mendoza (Juillet 2026). Adaptive real-time optimization for electric bus depot operations with state-of-charge tracking signal, Rapport technique, Les Cahiers du GERAD G– 2026–41, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2026-41) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: S. Khalilpourazari, R. Barra, G. Desaulniers, J. E. Mendoza (July 2026). Adaptive real-time optimization for electric bus depot operations with state-of-charge tracking signal, Technical report, Les Cahiers du GERAD G–2026–41, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2026-41) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Adaptive real-time optimization for electric bus depot operations with state-of-charge tracking signal

Soheyl Khalilpourazari [a, b]

Romane Barra [c]

Guy Desaulniers [a, b]

Jorge E. Mendoza [b, d]

[a] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4*

[b] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[c] *École Polytechnique, Palaiseau, 91129, France*

[d] *Department of Logistics and Operations Management, HEC Montréal, Montréal (Qc), Canada, H3T 2A7*

soheyl.khalilpourazari@polymtl.ca
guy.desaulniers@gerad.ca
jorge.mendoza@hec.ca

July 2026
Les Cahiers du GERAD
G–2026–41

Abstract : Managing electric bus (EB) fleets at scale introduces operational challenges that differ fundamentally from those of conventional fuel-powered fleets. A main challenge is the need to jointly coordinate parking, recharging, and service dispatching under tight time and infrastructure constraints while accounting for uncertainty in energy consumption. We consider an indoor depot layout with parallel FIFO lanes and a limited number of simultaneous chargers (a configuration used in Canada and other Nordic countries). For this setting, we formulate the resulting real-time electric bus parking, charging, and dispatching problem as a mixed-integer linear program solved at each EB arrival within a sliding-window framework. To account for systematic errors in energy consumption predictions, we introduce a tracking signal algorithm, based on a backward cumulative sum scheme, that continuously monitors energy consumption prediction errors and dynamically adjusts arrival state-of-charge predictions for future EBs. Computational experiments on instances with up to four lanes and twelve spots per lane, covering 1,500 stochastic scenarios across five energy consumption regimes, show that the adaptive approach reduces reserve bus deployments by up to 55.6% relative to a static forecasting baseline. It also improves solution quality by up to 27.9%, without increasing computational times.

Keywords : Electric bus management; state-of-charge uncertainty; recharge scheduling; bus parking; tracking signal

Acknowledgements: We would like to thank Tom Kasprzak from École Polytechnique, Paris, France, for initiating the study of this problem as part of an internship at Polytechnique Montréal. We are also grateful to the personnel of GIRO for their collaboration throughout this project. Finally, we acknowledge the financial support of GIRO, the Natural Sciences and Engineering Research Council (NSERC) of Canada (Alliance grant: ALLRP 567169–21), Prompt Québec (PSO grant: 171_Desaulniers-Giro 2021.12), and Fonds de recherche du Québec – Nature et technologies (FRQNT) (B3X postdoctoral scholarship: 373603).

1 Introduction

Road transportation accounts for 15% of global greenhouse gas emissions (Jaramillo et al., 2022), making electrification of vehicle fleets a priority for reducing urban air pollution and meeting climate targets. Yet, widespread adoption of electric vehicles remains constrained by battery costs and inadequate charging infrastructure (International Energy Agency, 2023). Beyond infrastructure, the operational characteristics of electric vehicles introduce new planning challenges absent in conventional fleets including recharging takes hours rather than minutes, batteries require frequent recharging, and charging capacity at depots is limited. For public transit operators managing large electric bus (EB) fleets, these constraints make the coordinated scheduling of parking, recharging, and service dispatching a critical and complex operational problem.

Depot management is one of the principal obstacles in the electrification of public transit fleets. In Canada and many Nordic countries, EBs are parked in indoor depots to protect batteries and mechanical systems from extreme cold and heavy snowfall. While some outdoor spots may exist, EBs scheduled for the following day are parked indoors during winter, making the indoor depot the operational unit of interest. These facilities are typically smaller than outdoor depots due to construction and operating costs, and are organized as a set of narrow parallel lanes in which overtaking is prohibited for safety reasons. Lanes therefore operate under a first-in, first-out (FIFO) policy, whereby departure order must respect arrival order. Each parking spot is equipped with a charging outlet, but grid capacity constraints limit the number of EBs that can charge simultaneously. This combination, FIFO lane policy, constrained charging capacity, and weather-driven variability in battery state-of-charge (SoC), defines the operational environment studied in this paper.

The operational context of this paper is inspired by real-world practice at EB depots in Quebec, including the STL depot in Laval and the STM Stinson depot in Montreal, and was developed in collaboration with transit software developer GIRO Inc. While our model captures the main structural features of these depots, it does not aim to replicate every operational detail. In these settings, daily trip schedules are precomputed by solving an electric vehicle scheduling problem (eVSP) and organized into sequences of consecutive trips, hereafter called services, defined by their start time, end time, and net energy consumption. An offline assignment plan, determined in advance by solving an EB assignment problem with parking constraints (see Azéma et al., 2024), pairs each EB with an upcoming service and specifies its parking lane and charging schedule. In practice, however, this plan is routinely disrupted by varying energy consumption with weather conditions that causes EBs to arrive at the depot with SoC levels that deviate from predictions. When deviations are severe, planned EB-to-service assignments may become infeasible, forcing agencies to deploy reserve buses, often diesel-powered, to maintain service continuity at considerable fuel and environmental cost. Avoiding such deployments requires dispatchers to make rapid, integrated decisions upon each EB arrival including where to park it, whether and when to charge it, and whether to revise its planned service assignment. These decisions must simultaneously account for the actual observed SoC, the evolving state of FIFO lanes, and the risk of depot entrance congestion if arriving EBs cannot be processed quickly. This dynamic, uncertainty-driven environment calls for adaptive real-time optimization methods.

In this paper, we formulate the real-time EB parking, charging, and dispatching problem (RTeBPCDP) and develop a solution methodology for its real-time resolution. The RTeBPCDP captures the joint decisions that must be made upon each EB arrival (parking lane assignment, charging schedule, and service assignment) within a depot operating under FIFO lane policy and limited charging capacity. Decisions are made over a rolling look-ahead interval and are therefore provisional meaning that they may be revised as subsequent EBs arrive and new information becomes available. The problem is formulated as a mixed-integer linear program (MILP) minimizing a weighted combination of reserve bus deployments, lane forward moves, and service changes. To handle SoC forecast uncertainty arising from systematic, fleet-wide factors such as temperature variability, we develop a tracking signal algorithm that dynamically corrects energy consumption predictions as EBs arrive, so that forecasts and decisions co-evolve throughout the day. The combined framework is evaluated through a sliding-window simulation over 1,500 stochastic scenarios across six depot configurations, demonstrating that adaptive SoC forecasting reduces reserve bus deployments by up to 55.6% and improves overall solution quality by up to 27.9% relative to a static forecasting baseline, with no added computational overhead.

The remainder of the paper is structured as follows. Section 2 reviews the relevant literature related to the RTeBPCDP. Section 3 formally defines the problem. Section 4 presents the mathematical model, while Section 5 introduces the tracking signal algorithm. Section 6 details the sliding-window framework. Section 7 reports computational results, analyzes the performance of the adaptive approach under diverse operational conditions, and provides sensitivity analyses on key model parameters. Finally, Section 8 concludes the paper and outlines directions for future research.

2 Literature review

The RTeBPCDP sits within a broader decision making hierarchy: upstream, the eVSP determines daily EB schedules whose output feeds directly into the RTeBPCDP, while downstream, bus parking and dispatching and on-site charging scheduling constitute its two principal subproblems. The literature offers no real-time treatment of the full problem, but partial coverage exists at each level of this hierarchy. For the eVSP, comprehensive reviews are provided by Perumal et al. (2022) and Dirks et al. (2022), thus, we do not review it further here. For the two subproblems, which directly inform our model design, we provide dedicated reviews in Sections 2.1 and 2.2. We begin below with the works most directly related to the RTeBPCDP as a whole, namely, offline, deterministic variants that share its structural features but fall short of its full scope. Section 2.3 briefly discusses the literature on tracking signals.

The closest work to ours is Azéma et al. (2024), who study an indoor EB depot with FIFO lanes, limited charging capacity, and precomputed trip services from an upstream eVSP, namely, the same structural setting we adopt. However, they address an offline, deterministic feasibility problem defined over a one to three day horizon, jointly assigning EBs to services, lanes, and charging schedules without real-time decision-making, reserve-bus modeling, or SoC forecasting. By contrast, our RTeBPCDP is solved at each EB arrival over a short rolling horizon and additionally (i) allows service exchanges between EBs, (ii) explicitly models reserve-bus deployments and lane-forward moves in a weighted minimization objective, and (iii) introduces a tracking signal that adaptively corrects arrival SoC forecasts. While Azéma et al. (2024) use constraint programming, our formulation is a MILP solved by a commercial solver. In a follow-up paper, Azéma et al. (2025) develop a MILP approach for the same planning problem that outperforms their earlier constraint programming model on three-day instances. Their formulation differs substantially from ours, as their horizon spans multiple days during which each EB may visit the depot more than once.

Messaoudi and Oulamara (2019) study a related planning problem in which each incoming EB must be assigned to a parking spot, a charging schedule, and an upcoming service, subject to a limited number of chargers and discrete 15-minute recharge periods. Unlike our setting, their model assumes that each EB occupies the same parking spot throughout its depot visit, precluding FIFO lane dynamics, and minimizes a cost function combining fixed EB costs and energy costs rather than operational disruptions. They propose a three-step heuristic that sequentially assigns EBs to services, parking spots, and charging schedules.

2.1 Bus parking and dispatching

Depot planning involves positioning buses so that they can perform their assigned services, recharging beforehand if needed. Hamdouni et al. (2006) study a bus parking and dispatching problem for diesel buses that plans evening and morning depot operations. Buses arriving in the evening are assigned to spots in a parking lot under a FIFO policy, a direct consequence of the on-site safety risks posed by backing maneuvers. Because buses come in different types, each must be parked so that every next-morning service can be operated on time by a bus of the required type. To facilitate this, lanes are restricted to at most two bus types, each occupying contiguous spots. Such arrangements, called block patterns, improve the solution's robustness to changes in the buses' arrival order. The authors formulate the problem as an integer linear program minimizing the number of lanes that require rearrangement to remain feasible.

Two follow-up papers extend this framework. Hamdouni et al. (2007a) study two variants in which lane rearrangement is no longer allowed. In the first, block patterns are computed in advance but may be violated by the actual bus arrival or departure sequence, creating incompatibilities. The objective minimizes incompatibilities first, then the number of two-block lanes used, and the resulting integer program is solved exactly. The second variant generalizes the first by permitting a bus of the wrong type to be assigned to a service, a so-called mismatch, yielding a threefold objective that minimizes mismatches, incompatibilities, and two-block lanes in sequence. This richer model is solved via Benders decomposition. Building on the same framework but introducing uncertainty in bus arrival times, Hamdouni et al. (2007b) propose two robust formulations: a k-position approach that preserves feasibility against arrival shifts of up to k positions, and a stochastic approach that minimizes the expected value of a penalty increasing in the number of arrivals that would make the plan infeasible. Both are formulated as integer programs and solved with a commercial MILP solver.

2.2 On-site EB charging scheduling

Many authors have studied EB recharging strategies. Chao and Xiaohong (2013) propose a battery-swapping model in which depleted batteries are exchanged for fully charged ones within 15 minutes, mimicking the speed of diesel refueling, though this approach has seen limited practical adoption. Sassi and Oulamara (2017) study a single-day scheduling and charging problem for a mixed diesel-EB fleet with heterogeneous charger types and soft grid capacity constraints, assuming a linear charging function and 15-minute discretized recharge periods. They formulate the problem as a MILP and develop two heuristics for large instances. Zhang et al. (2022) address a multi-depot scheduling problem with heterogeneous EB types and partial recharging, allowing buses to charge only the energy needed for their next service rather than to full capacity. They formulated the problem as a MILP minimizing fleet acquisition and operational costs and solved via an adaptive large neighborhood search with specialized destroy-repair operators and embedded local search. Nonlinear charging functions, in contrast to the linear assumption used above, are explored by van Kooten Niekerk et al. (2017) and Olsen and Kliewer (2020).

A second stream of work combines limited charging capacity with additional considerations such as battery degradation and multiday planning horizons. Zhang et al. (2021) study an EB scheduling problem incorporating battery degradation, nonlinear charging profiles, and a limited number of chargers. They formulate it as a MILP minimizing total operational costs and solve it via a branch-and-price algorithm with dual stabilization and a multi-label correcting method for the pricing subproblem. Vendé et al. (2023) address a multiday EB assignment and overnight recharge scheduling problem in which vehicle blocks are assigned to EBs and charging operations are scheduled over several days, accounting for limited chargers, piecewise linear charging functions, and calendar aging costs that favor charging closer to departure times. They formulate the problem as a MILP based on a two-commodity network flow model and develop two matheuristics that solve instances with up to 60 EBs over 5 days.

Klein and Schiffer (2023) further extend this line of work by considering a charge scheduling problem for electric commercial vehicles with pre-assigned service operations, limited charger capacity, nonlinear charging functions, battery degradation costs, and time-of-use energy tariffs, where service operations are pre-assigned, but flexible departure times within predefined windows allow charging schedules to be optimized. They solve the problem with an exact branch-and-price algorithm based on a set-covering formulation, modeling the pricing subproblem as a resource-constrained shortest path problem on a time-expanded network and solving it with a custom label-setting algorithm. This approach scales to instances with up to 68 vehicles over a 5-day horizon.

2.3 Tracking signals

Tracking signals are widely used in forecasting and process control to monitor the bias of prediction models. The classical tracking signal divides the sum of forecast errors (or residuals) by the mean absolute deviation and flags persistent over- or under-forecasting when the statistic exceeds predetermined limits (Brown, 1959). Brown (1959) also introduced cumulative-sum (CUSUM) tracking signals for inventory control, and Harrison and Davies (1964) later showed that CUSUM can detect demand-pattern changes over time, signaling when a forecasting system has become inadequate and requires re-estimation. Conventional (forward) CUSUM, however, suffers from reduced power and longer detection delays when early observations are uninformative. To address this, Harrison and Davies (1964) proposed a *backward* CUSUM that cumulates recursive residuals from the end of the sample, collecting the most informative residuals first and thereby achieving higher power and shorter detection delays, a property particularly valuable for real-time monitoring. Several variants of the backward CUSUM have since been developed and applied across diverse domains, recently improving bias detection in COVID-19 case data (Otto and Breitung, 2023), surgical learning curves (Lin et al., 2023), and manufacturing process control (Superville, 2019).

3 Problem description

Let us consider an indoor EB depot as a set L of parallel lanes, in which the EBs are parked following a FIFO policy. In each lane, the EBs enter at one end (the *back* of the lane) to reach a parking spot and, when they must leave, they exit at the other end (the *front* of the lane), without taking over any EB. We assume that there are n^P parking spots available in each lane but our model can easily be adapted to a varying number of spots per lane. We also assume that the EBs are all identical and can be charged at any parking spot. However, only a maximum number of chargers n^R can be used simultaneously due to power grid restrictions. Finally, we assume that an EB dispatching plan is available for the coming days, i.e., the list of upcoming services planned to be performed by each EB. In the RTeBPCDP, it is possible to change the planned assignment of an EB to a service but this change incurs a penalty.

The RTeBPCDP occurs when an EB $\bar{b}$ arrives at the depot after performing a service. EB $\bar{b}$ must be assigned to a parking lane in the depot, a charging schedule (if required), and an upcoming service, among a set S of services that must be performed after its arrival. To make these decisions, we consider the set B^P of EBs that are already parked at the depot, and the set B^A of EBs, including $\bar{b}$, that are scheduled to arrive within a short time interval (e.g., in the next 60 minutes). We cannot change the lane assigned to the parked EBs in B^P, but we can move them forward if spots are free at the front of their lane. We can also change their planned upcoming service and reschedule their charging operation if it has not begun or is ongoing (in which case only its duration can be modified). On the other hand, for the arriving EBs in $B^A \setminus \{\bar{b}\}$, we also plan their parking lanes, charging schedules, and upcoming service assignments but these decisions might be revised when the next EB arrives. Note that, at the subsequent arrivals, EB $\bar{b}$ will be considered a parked EB which means that its proposed upcoming service and charging schedule may also be revised. The time at which the problem is solved is called the *initial time* and the time interval during which we consider EB arrivals is called the *look-ahead interval*.

We assume that the look-ahead interval is sufficiently short such that every EB in $B^P \cup B^A$ needs to be assigned to a single upcoming service in S, i.e., it does not have enough time to perform its first assigned upcoming service and return to the depot within the look-ahead interval. Thus, the number of services to cover must be equal to the total number of EBs in $B^P \cup B^A$, that is, $|S| = |B^P \cup B^A|$.

As mentioned above, some EBs might already be parked in the depot at the initial time. Thus, some lanes may have occupied parking spots while others might be fully empty. However, as an EB always parks in a lane at the free spot that is the closest to the lane's front, the occupied spots in a lane are always consecutive, possibly leaving free spots at the back, at the front, or at both ends. Therefore, at the initial time, only a limited number of parking spots are accessible, namely, the free ones at the back of the lanes and all spots of the empty lanes. Nevertheless, as the EBs arrive at the depot, some already parked EBs may leave to perform a service, possibly emptying a full lane and giving access to all its parking spots. Given that we consider only the EBs arriving in a short look-ahead interval, we assume that any EB arriving in this interval does not leave the depot within this interval. This assumption is realistic because the look-ahead interval is short (for example, 60 minutes) whereas EBs typically remain at the depot for several hours to recharge and not for only a few minutes. Moreover, an EB with sufficient charge for its next service is dispatched to it directly and would not pass through a lane only to leave again within this interval.

When dedicated drivers are available in the depot, parked EBs in a lane can be moved forward when there are free spots at its front. This option, that we call *lane forward move*, can be applied sparingly to increase the number of accessible parking spots. A lane forward move can be executed in at most one lane within any time interval of a fixed duration (say, 15 minutes). This restriction reflects the limited personnel available for these maneuvers, since typically only one or two drivers perform them, so at most one move occurs across all lanes within each interval. Given this practical restriction, we assume that at most one such operation can be performed per lane during the look-ahead interval and that it incurs a cost.

To ensure operations feasibility, transit operators always have a sufficient number of reserve buses that are available at any time and ready to cover any service if required (e.g., these can be diesel buses). To model these reserve buses, we assume that there are special parking spots that are always available (we say that they form the reserve lane l^*). An arriving EB that is assigned to the reserve lane incurs a large penalty cost and is seen as seamlessly replaced by a reserve bus. The lanes in L are said to be the *regular* lanes.

When assigned to an upcoming service $s \in S$ with an expected energy consumption of e_s, an EB must be sufficiently charged to complete it, i.e., at the end of the service, the EB's SoC cannot be less than a minimum SoC (set to 0 without loss of generality). While arriving EBs have a forecasted arrival SoC, their actual arrival SoC might deviate from this forecast due to factors such as weather temperature variations. Therefore, decisions must be based on the actual SoC observed upon arrival rather than the forecast for the currently arriving EB $\bar{b}$. Furthermore, an EB might need to recharge its battery between its arrival time and the departure time of its assigned upcoming service. To schedule the recharges, we consider a discretized horizon into disjoint periods of duration δ. This discretization is independent of the initial time. For instance, if the initial time is 8:52 and $\delta = 15$, the periods can be [9:00, 9:15], [9:15, 9:30], [9:30, 9:45], etc. The first of these periods is called the *initial period* and starts no later than $\delta - 1$ minutes after the initial time. To limit long-term battery wear, we assume that charging operations are non-preemptive that is, a recharge covers a number of consecutive periods. Consequently, for every parked EB, we need to know its *initial charging status*, i.e., if charging is finished, ongoing, or not started. We assume that the charging function is linear (i.e., the SoC gain is proportional to the charging duration), but our model can easily be adapted to a nonlinear function. For every EB $b \in B$, we denote by $q_b \in [0, 1]$ its (real or forecasted) SoC at the beginning of the initial period. Finally, we neglect the recharging time lost when EBs are moved forward in a lane and limit to n^R the number of EBs that can recharge simultaneously.

Without loss of generality, we assume that the arrival times of the EBs in B^A are all distinct and, therefore, that we know the EBs' arrival order. We also assume that the recharge of an EB b can start at the earliest at the beginning of the recharging period following its arrival if $b \in B^A$ or at the beginning of the initial period if $b \in B^P$. Symmetrically, an EB assigned to a service can end its recharge at the latest at the end of the period preceding the start of this service. Finally, we also assume that the start times of the services in S are all different.

In summary, the RTeBPCDP can be stated as follows. Given a depot divided into lanes with parking spots, a set of already parked EBs, a set of arriving EBs, and a set of upcoming services to cover, the problem consists in assigning a lane, a charging schedule, and an upcoming service to the first arriving EB, while possibly revising the planned charging schedules and service assignments of the parked EBs, and planning the lane assignment, the charging schedule and the upcoming assignment of the other arriving EBs. These decisions must respect the following constraints including lane capacity, lanes' FIFO policy, at most one lane forward move per lane over the look-ahead interval and one per time interval across all lanes, battery capacity, charging capacity, and parked EBs' initial charging status. The objective function seeks to minimize the penalty costs incurred by using reserve buses, performing a lane forward move, and changing a planned service assignment for all considered EBs.

Figure 1 provides an illustrative example of a solution to the RTeBPCDP. For each arriving EB in the look-ahead interval, it indicates the proposed charging periods, the departure time of its upcoming service, and its parking lane/spot. Because EB 5 and EB 6 are not in the look-ahead interval, no charging periods, nor lanes/spots are yet planned for them.

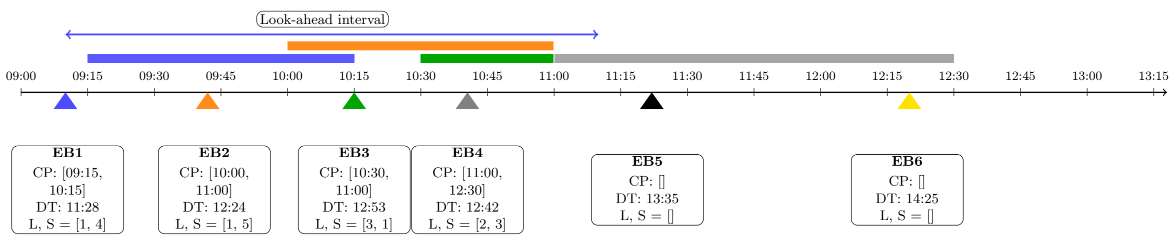

Figure 1: Arriving EBs parking, charging, and dispatching.

Note: CP: charging period. DT: departure time. L: lane. S: spot.

4 Mathematical formulation

In this section, we propose a MILP formulation for the RTeBPCDP. We first introduce our mathematical notation in Section 4.1 before defining the model's objective function in Section 4.2. Then, we walk the reader through several sets of constraints, grouped by functionality, in Section 4.3.

4.1 Notation

To model the RTeBPCDP, we use the sets, parameters and variables defined in Tables 1–3, respectively.

Table 1: Notation – Sets.

Sets	
B^A	Set of arriving EBs;
B_b^{A-}	Set of EBs arriving before EB b;
B^P	Set of EBs already parked;
B	Set of all EBs ($B = B^A \cup B^P$);
S	Set of upcoming services;
S_b^R	Set of upcoming services that can be assigned to EB $b \in B$ only if it recharges after the initial time;
L	Set of regular lanes;
L^*	Set of all lanes, including the reserve lane;
P	Set of time periods considered for recharging;
P_b	Set of possible periods for starting a recharge for EB b, taking into account the initial charging status if $b \in B^P$;
H_{bp}	Set of possible recharging durations (in periods) for EB b when the recharge starts at period p.

Table 2: Notation – Parameters.

Parameters	
l^*	Reserve lane;
l_b	Lane containing parked EB $b \in B^P$;
a_b	Arrival time at the depot of EB $b \in B$;
s_b	Planned upcoming service for EB $b \in B$;
d_s	Departure time of service $s \in S$;
q_b	Predicted SoC at the arrival of EB $b \in B^A \setminus \{\bar{b}\}$ or real SoC of EB $b \in B^P \cup \{\bar{b}\}$ at the start of the initial period ($\in [0,1]$);
e_s	Expected SoC decrease (energy consumed) along service $s \in S$ ($\in]0,1]$);
γ	Recharging rate (SoC gain per period);
n^R	Maximum number of EBs that can recharge simultaneously;
δ	Duration of a period in P;
p_b^E	Earliest recharge starting period for EB $b \in B$;
p_s^L	Latest recharge ending period of the EB assigned to service $s \in S$;
n^P	Number of parking spots in a lane;
f_l^B	Number of free parking spots at the back of lane l at the initial time ($= n^P$ if l is initially empty);
f_l^F	Number of free parking spots at the front of lane l at the initial time ($= 0$ if l is initially empty);
Δ	Duration of the time intervals during which at most one forward lane move can be performed;
w_i	Weight associated with term $i \in \{1,2,3\}$ in the objective function.

Table 3: Notation – Variables.

Variables	
x_{bs}^l	Binary variable equal to 1 if EB $b \in B$ is assigned to lane $l \in L^*$ and service $s \in S$;
y_b^{ph}	Binary variable equal to 1 if EB $b \in B$ starts to recharge at period $p \in P_b$ for h periods ($h \in H_{bp}$);
z_b^l	Binary variable equal to 1 if EB $b \in B^A$ is parked in lane $l \in L$ which has been emptied between the initial time and the arrival of b;
v_b^l	Binary variable equal to 1 if EB $b \in B^A$ is assigned to lane $l \in L$ which is subject to a lane forward move just before b's arrival;
t_b^l	Non-negative integer variable indicating the number of free parking spots gained by a lane forward move in lane $l \in L$ upon the arrival of EB $b \in B^A$.

Some variables can be fixed to 0 a priori and can, thus, be removed (though we keep them in our model for notational conciseness). They are:

- $x_{bs}^l = 0$, $\forall s \in S, b \in B^P, l \in L^* \setminus \{l_b\}$, because EB b must stay in lane l_b;
- $x_{bs}^l = 0$, $\forall s \in S, l \in L^*, b \in B^A$ such that $a_b \geq d_s$, because EB b cannot be assigned to a service that departs before its arrival;
- $x_{bs}^l = 0$, $\forall s \in S, l \in L, b \in B^A$ such that $q_b + \gamma(p_s^L - p_b^E + 1) < e_s$, because EB b cannot be assigned to regular lane l and service s if it does not have enough time to charge;
- $z_b^l = v_b^l = t_b^l = 0$, $\forall b \in B^A, l \in L$ if lane $l \in L$ is empty at the initial time.

4.2 Objective function

The objective function is three-fold:

1. Minimize the number of EBs assigned to the reserve lane;
2. Minimize the number of lane forward moves; and
3. Minimize the number of upcoming service changes;

These objective functions are aggregated into a single one using weights w_1 to w_3 as follows:

$$\min \quad w_1 \sum_{b \in B^A} \sum_{s \in S} x_{bs}^{l^*} + w_2 \sum_{b \in B^A} \sum_{l \in L} v_b^l + w_3 \sum_{b \in B} \sum_{l \in L^*} \sum_{s \in S \setminus \{s_b\}} x_{bs}^l. \tag{1}$$

4.3 Constraints

Constraints (2)–(5) describe how the parking depot operates:

$$\sum_{l\in L^*}\sum_{s\in S} x^l_{bs} = 1, \qquad \forall b \in B^A \tag{2}$$

$$\sum_{s\in S} x^{l_b}_{bs} = 1, \qquad \forall b \in B^P \tag{3}$$

$$\sum_{b\in B}\sum_{l\in L^*} x^l_{bs} = 1, \qquad \forall s \in S \tag{4}$$

$$\sum_{\substack{s'\in S:\\ d_{s'}<d_s}} x^l_{bs'} \leq 1 - \sum_{\substack{b'\in B:\\ a_{b'}<a_b}} x^l_{b's}, \qquad \forall b \in B, s\in S, l \in L. \tag{5}$$

Constraints (2) ensure that each arriving EB is assigned to a unique lane and a unique upcoming service, whereas constraints (3) assign every parked EB to a service. Constraints (4) guarantee that each upcoming service is covered by an EB or a reserve bus. Constraints (5) enforce the FIFO policy for each regular lane. More precisely, for a regular lane l, an EB b, and an upcoming service s, the constraint stipulates that, if b is assigned to l and a service s' that departs earlier than s, then it is not feasible to assign in the same lane an EB b' that would arrive earlier than b and be assigned to service s.

Constraints (6)–(9) model the charging operations:

$$\sum_{p\in P_b}\sum_{h\in H_{bp}} y^{ph}_b \leq \sum_{l\in L}\sum_{s\in S} x^l_{bs}, \qquad \forall b \in B \tag{6}$$

$$\sum_{b\in B}\sum_{\substack{p'\in P_b:\\ p'\leq p}}\sum_{\substack{h\in H_{bp'}:\\ p\leq p'+h-1}} y^{p'h}_b \leq n^R, \qquad \forall p \in P \tag{7}$$

$$\sum_{l\in L} x^l_{bs} \leq \sum_{p\in P_b}\sum_{\substack{h\in H_{bp}:\\ q_b+\gamma h\geq e_s\\ p+h-1\leq p^L_s}} y^{ph}_b, \qquad \forall b\in B, s\in S^R_b \tag{8}$$

$$\sum_{p\in P_b}\sum_{\substack{h\in H_{bp}:\\ p+h-1>p^L_s}} y^{ph}_b \leq 1 - \sum_{l\in L} x^l_{bs}, \qquad \forall b\in B, s\in S\setminus S^R_b. \tag{9}$$

Constraints (6) ensure that a maximum of one charging operation is scheduled for each EB parked in a regular lane and none for an EB assigned to the reserve lane. Constraints (7) restrict to n^R the number of EBs that can recharge simultaneously in any period p. Constraints (8)–(9) link the x and y variables. Constraints (8) impose that, if EB b is assigned to a regular lane and an upcoming service s that requires a recharge ($s \in S^R_b$), then b must also be assigned to a feasible recharge that will start sufficiently early to end before the start of service s. Constraints (9) stipulate that, if EB b is assigned to a regular lane and a service s that does not require to recharge b ($s \in S \setminus S^R_b$), then any optional recharge operation for b must finish before the start of service s, that is, y^{ph}_b can be equal to 1 only if $p \in P_b$, $h \in H_{bp}$ and $p + h - 1 \leq p^L_s$.

Constraints (10)–(11) are used to identify the lanes that are emptied before assigning them a first arriving EB:

$$z^l_b \leq \sum_{s\in S} x^l_{bs}, \qquad \forall l \in L, b\in B^A \tag{10}$$

$$z^l_b \leq 1 - \sum_{\substack{s\in S:\\ d_s>a_b}} x^l_{b's}, \qquad \forall l\in L, b\in B^A, b'\in B^P\cup B^{A-}_b. \tag{11}$$

These constraints impose upper bounds on the value of the z_b^l variables. More precisely, z_b^l is set to zero if EB b is not assigned to lane l (constraints (10)), or if at least one EB b' that is initially parked in lane l or arrives before b still occupies it when b arrives (constraints (11)). Note that, for each lane l, at most one variable z_b^l can take value 1 as we assume that each arriving EB will not leave it during the look-ahead interval.

Constraints (12)–(17) concern the lane forward moves and the spots gained as a result of these moves:

$$v_b^l \leq \sum_{s \in S} x_{bs}^l, \qquad \forall l \in L, b \in B^A \tag{12}$$

$$v_b^l \leq 1 - \sum_{\substack{b' \in B^A: \\ a_{b'} \leq a_b}} z_{b'}^l, \qquad \forall l \in L, b \in B^A \tag{13}$$

$$\sum_{b \in B^A} v_b^l \leq 1, \qquad \forall l \in L \tag{14}$$

$$\sum_{l \in L} \sum_{\substack{b' \in B_b^{A-} \cup \{b\}: \\ a_b - a_{b'} < \Delta}} v_{b'}^l \leq 1, \qquad \forall b \in B^A \tag{15}$$

$$t_b^l \leq (n^P - 1) v_b^l, \qquad \forall l \in L, b \in B^A \tag{16}$$

$$t_b^l \leq f_l^F + \sum_{b' \in B^P} \sum_{\substack{s \in S: \\ d_s \leq a_b}} x_{b's}^l, \qquad \forall l \in L, b \in B^A. \tag{17}$$

Constraints (12) ensure that variables v_b^l take value 0 if EB b is not assigned to lane l, whereas constraints (13) also set these variables to 0 if lane l has been previously emptied (in which case there are no free spots at the lane's front after assigning a first arriving EB to this lane). Constraints (14) and (15) impose a maximum of one move per lane during the look-ahead interval and a maximum of one move per time interval of duration Δ over all lanes. Next, constraints (16) set to 0 the number of accessible parking spots gained in a lane l by a lane forward move if no such move is occurring before the arrival of EB b. Alternatively, when such a move occurs, constraints (17) determine the maximum number of free spots that can be gained by summing the number of spots available at the initial time at the front of the lane (f_l^F) and the number of initially parked EBs that have pulled out of the depot. Note that, when a lane forward move is triggered (usually anticipated by a few minutes), the EBs arriving during the move and assigned to the corresponding lane are temporarily staged at the back of the lane or in a free outside spot.

Constraints (18) ensure that the capacity of each lane is never exceeded:

$$\sum_{\substack{b' \in B^A: \\ a_{b'} \leq a_b}} \sum_{s \in S} x_{b's}^l \leq f_l^B + \sum_{\substack{b' \in B^A: \\ a_{b'} \leq a_b}} t_{b'}^l + (n^P - f_l^B) \sum_{\substack{b' \in B^A: \\ a_{b'} \leq a_b}} z_{b'}^l, \qquad \forall l \in L, b \in B^A. \tag{18}$$

For a lane l and an arriving EB b, the left-hand side computes the number of arriving EBs assigned to a lane l up until a_b, the arrival time of b. The right-hand side is equal to the total number of spots in lane l that are or have been available until a_b. This number always includes the number of spots available at the back of lane l at the initial time (f_l^B). It is augmented by the number of spots gained ($t_{b'}^l$) by a lane forward move occurring at time a_b or before, and the number of spots gained ($n^P - f_l^B$) by emptying the lane before assigning it a first EB ($z_{b'}^l = 1$).

Finally, the domains of the variables are:

$$x_{bs}^l \in \{0,1\}, \qquad \forall l \in L^*, b \in B, s \in S \tag{19}$$

$$y_b^{ph} \in \{0,1\}, \qquad \forall b \in B, p \in P_b, h \in H_{bp} \tag{20}$$

$$z_b^l \in \{0,1\}, \qquad \forall l \in L, b \in B^A \tag{21}$$

$$v_b^l \in \{0,1\}, \qquad \forall l \in L, b \in B^A \tag{22}$$

$$t_b^l \geq 0, \qquad \forall l \in L, b \in B^A. \tag{23}$$

Note that, for a parked EB $b \in B^P$, its lane is fixed to l_b, so the only variables x_{bs}^l that are not fixed to zero are those with $l = l_b$. The t_b^l variables are declared continuous, as their integrality is redundant given the binary requirements on the other variables. Consequently, model (1)–(23) is a mixed-integer linear program (MILP) that can be solved by a branch-and-cut algorithm, namely, one provided by a commercial MILP solver.

5 Tracking signal algorithm for dynamic energy consumption prediction adjustment

Model (1)–(23) takes as input the predicted SoC q_b of each EB $b \in B^A \setminus \{\bar{b}\}$ arriving after $\bar{b}$ within the look-ahead interval. When solving the first RTeBPCDPs of the day, q_b is typically derived from the expected energy consumption of the service that b performs before reaching the depot. However, external factors such as weather variability can introduce systematic, correlated errors into these energy consumption predictions across all EBs. When such biases are detected, the predictions can be corrected for the RTeBPCDPs solved at subsequent arrivals. To this end, we integrate a tracking signal algorithm, based on a modified backward CUSUM framework (Harrison and Davies, 1964; Gardner, 1983, 1985; Krishnamurthy, 2006), that monitors the discrepancy between predicted and realized energy consumption upon each EB arrival and dynamically adjusts the predicted energy consumption and, consequently, the predicted arrival SoC of future EBs.

Let $\hat{B}^A$ be the set of EBs that are planned to arrive at the depot during a certain time horizon that is at least as long as the look-ahead interval. For each EB $b \in \hat{B}^A$, we denote by $e_{s_b^-}$ the expected energy consumption of the service s_b^- assigned to b before its arrival, by $e_{s_b^-}^P$ the adjusted energy consumption prediction (initialized to $e_{s_b^-}$ at the start of the day), and by $e_{s_b^-}^O$ the energy consumption observed upon arrival. Note that the adjusted prediction $e_{s_b^-}^P$ is updated by the tracking signal algorithm. Upon the arrival of EB $\bar{b}$, we compute the relative prediction error of the energy consumption as follows:

$$\varepsilon_{\bar{b}} = \frac{e_{s_{\bar{b}}^-}^O - e_{s_{\bar{b}}^-}^P}{e_{s_{\bar{b}}^-}^P}, \tag{24}$$

assuming that $e_{s_{\bar{b}}^-}^P \neq 0$.

We set $\sigma = 0.05$ as the nominal standard deviation of the relative prediction error ε, representing the inherent variability in energy consumption predictions. The initial control limit is set as $L_0 = \sigma\, \omega\, \eta$, where $\omega = 0.1$ and $\eta = 12.0$ are tunable parameters. These values are chosen based on a parameter tuning process that is explained in Section 7.1. We initialize the backward CUSUM moving statistics D^+ and D^- as $D^+ = L_0$ and $D^- = -L_0$, and update them iteratively for each arriving EB $\bar{b} \in \hat{B}^A$ as follows:

$$D^+ \leftarrow \min\{D^+, L_0\} + \sigma\omega - \varepsilon_{\bar{b}}, \tag{25}$$

$$D^- \leftarrow \max\{D^-, -L_0\} - \sigma\omega - \varepsilon_{\bar{b}}. \tag{26}$$

When EB $\bar{b}$ arrives and the real energy consumption $e_{s_{\bar{b}}^-}^O$ of its associated service is observed, the algorithm can detect three states based on the CUSUM statistics. A positive bias occurs when energy consumption predictions are consistently too low, indicated by $D^+ < 0$. Conversely, a negative bias occurs when energy consumption predictions are consistently too high, indicated by $D^- > 0$. Alternatively, the system is considered in control when $D^+ \geq 0$ and $D^- \leq 0$.

Upon detecting a bias, an adjustment factor α is computed as:

$$\alpha = \begin{cases} \min\left\{-\frac{D^+}{L_0} \cdot \beta,\, \nu\right\}, & \text{if } D^+ < 0 \text{ (positive bias)} \\ -\min\left\{\frac{D^-}{L_0} \cdot \beta,\, \nu\right\}, & \text{if } D^- > 0 \text{ (negative bias),} \end{cases} \tag{27}$$

where $\beta = 0.04$ controls adjustment sensitivity and $\nu = 0.1$ is the maximum adjustment allowed (again, see the parameter tuning process in Section 7.1). Then, based on the bias type, the algorithm adjusts the predicted energy consumptions of the services associated with the future arriving EBs b, i.e., those in $\hat{B}^A$ such that $a_b > a_{\bar{b}}$. For a negative bias, a conservative half-adjustment is applied as follows:

$$e^P_{s_b^-} \leftarrow e^P_{s_b^-}(1 + 0.5\alpha), \tag{28}$$

while for a positive bias, the update is as follows:

$$e^P_{s_b^-} \leftarrow e^P_{s_b^-}(1 + \alpha). \tag{29}$$

This adjustment difference is proposed because when an EB consumes less energy than predicted, it usually does not increase the risk of a reserve bus deployment as much as for the case of an EB consuming more energy.

Subsequently, we update the predicted SoC values q_b for future arrivals to account for the adjusted energy consumption. For each such EB b, the adjustment for a negative bias is:

$$q_b \leftarrow q_b - 0.5\alpha\, e^P_{s_b^-}, \tag{30}$$

and that for a positive bias is:

$$q_b \leftarrow q_b - \alpha\, e^P_{s_b^-}, \tag{31}$$

where $e^P_{s_b^-}$ in (30) and (31) is the predicted energy consumption value before the update in (28) and (29). Following each bias detection and adjustment, the CUSUM statistics are reset to their initial values: $D^+ = L_0$ and $D^- = -L_0$. All updated predicted SoC values are projected onto the feasible range $[0, 1]$. This adaptive algorithm enables persistent bias correction and robust energy consumption forecasting, maintaining computational efficiency and supporting real-time decision-making under uncertainty.

6 Sliding-window framework for real-time operations

Model (1)–(23) is solved to make four critical decisions including the parking lane assigned to the first arriving EB, whether a lane forward move is performed in that lane, the service assignment of every EB scheduled to depart before the second arriving EB, and the charging activities planned up to whichever is later, the start of the initial period or the end of the period containing the second arriving EB's arrival time. All other decisions may be revised once the second EB arrives, that is, when the next RTeBPCDP is solved. These critical decisions account for both the parked EBs and those arriving within the look-ahead interval, whose width we denote by W. A small W limits the number of arriving EBs considered, which may compromise decision quality through a lack of foresight. On the other hand, a large W considers more arriving EBs but can make model (1)–(23) harder to solve in real time. The value of W must therefore strike a balance between decision quality and computational tractability. To examine this trade-off and assess the performance of the tracking signal algorithm introduced in Section 5, we simulate depot operations, EB arrivals and departures, using a sliding-window framework, described next.

Let $T > W$ be a horizon (say, of a few hours), during which a set $\hat{B}^A$ of distinct EBs is planned to arrive at the depot. Thus, $a_b \in T$, $\forall b \in \hat{B}^A$. Without loss of generality, we consider that $\hat{B}^A =$

$\{b_1, b_2, \ldots, b_m\}$ is an ordered set of m EBs such that $a_{b_j} < a_{b_{j+1}}$, $\forall j \in \{1, 2, \ldots, m-1\}$. At the beginning of T, there is a set B^P of EBs parked in the depot and a set S of upcoming services to be assigned to the EBs in $B = B^P \cup \hat{B}^A$. Set S contains one planned service s_b for every EB $b \in B$. Service $s \in S$ may start after the end of T.

The proposed sliding window procedure consists in solving a RTeBPCDP for each arriving EB $b_j \in \hat{B}^A$ (in increasing order of arrival time) restricted to the arriving EBs in the interval $[a_{b_j}, a_{b_j} + W]$. After solving each problem, the critical decisions are fixed according to the computed solution and the next RTeBPCDP (for the next arriving EB) is built with updated input data. Let us detail how this update is performed assuming that the set of recharging periods P remains fixed from one RTeBPCDP to the next, e.g., the periods always start at the minutes 0, 15, 30, and 45 of every hour if $\delta = 15$.

Let RTeBPCDP$_j$ be the problem associated with EB b_j, $j \in \{1, 2, \ldots, m-1\}$. We denote by p_j^1 its initial period, i.e., the earliest period in P after a_{b_j}. We also denote by B_j^A, B_j^P, and S_j the subsets of arriving EBs, parked EBs, and upcoming services considered in RTeBPCDP$_j$. For RTeBPCDP$_{j+1}$, the subsets B_{j+1}^A, B_{j+1}^P, and S_{j+1}, and the updated input data are determined as follows.

Subset $B_{j+1}^A = \{b \in B \mid a_b \in [a_{b_{j+1}}, a_{b_{j+1}} + W]\}$ contains the EBs arriving in the interval $[a_{b_{j+1}}, a_{b_{j+1}} + W]$. For each EB $b \in B_{j+1}^A$, s_b does not change because s_b is the upcoming service that was planned for b considering the operations over a few days, not only over a few hours like in the previous optimization. Furthermore, $q_{b_{j+1}}$ is set to the observed arrival SoC of EB b_{j+1}, whereas the q_b values for the other arriving EBs $b \in B_{j+1}^A \setminus \{b_{j+1}\}$ are updated using formula (28) or (29) if the tracking signal procedure triggers an update of the predicted energy consumptions upon the arrival of bus b_{j+1}, and remain unchanged otherwise (no tracking signal is used or no update is triggered).

Denoting by $s_b^j \in S_j$ the service assigned to EB $b \in B_j^P$ in the solution of RTeBPCDP$_j$ and by $B_j^- = \{b \in B_j^P \mid d_{s_b^j} \leq a_{b_{j+1}}\}$ the set of EBs leaving the depot in the interval $]a_{b_j}, a_{b_{j+1}}]$, subset $B_{j+1}^P = \left(B_j^P \cup \{b_j\}\right) \setminus B_j^-$ if b_j was assigned to a regular lane and $B_{j+1}^P = B_j^P \setminus B_j^-$ if it was assigned to the reserve lane. For each parked EB $b \in B_{j+1}^P \cap B_j^P$, l_b and s_b do not change. Furthermore, if $p_{j+1}^1 = p_j^1$ (same initial period), neither the initial charging status of b, nor q_b change because they are set according to the beginning of the initial period. Otherwise, they must be updated according to the solution of RTeBPCDP$_j$. In particular, q_b increases by γ for each recharging period between the beginnings of periods p_j^1 and p_{j+1}^1. Furthermore, if $b_j \in B_{j+1}^P$, its planned upcoming service s_{b_j} does not change and its assigned lane l_{b_j} comes from the previous solution. Also, if $p_{j+1}^1 = p_j^1$, the initial charging status of b_j is set to “not started” and q_{b_j} does not change. Otherwise, these two information are deduced from the solution of RTeBPCDP$_j$.

Let $S_j^- = \{s \in S_j \mid d_s \leq a_{b_{j+1}}\}$ be the services in S_j that have started before or at the arrival time of EB b_{j+1} and $S_{j+1}^+ = \{s_b \in S \mid b \in B_{j+1}^A \setminus B_j^A\}$ the planned services for the additional arriving EBs considered in B_{j+1}^A. Then subset $S_{j+1} = \left(S_j \cup S_{j+1}^+\right) \setminus S_j^-$ if b_j was assigned to a regular lane and $S_{j+1} = \left(S_j \cup S_{j+1}^+\right) \setminus \left(S_j^- \cup \{s_{b_j}^j\}\right)$ if it was assigned to the reserve lane. These updates ensure that $|B_{j+1}^P \cup B_{j+1}^A| = |S_{j+1}|$.

The following parking data must also be updated to properly define RTeBPCDP$_{j+1}$: for each lane $l \in L$, the number of free spots at its front f_l^F and at its back f_l^B. The values are updated according to the parked EBs in B_j^P that have left the depot by time $a_{b_{j+1}}$, the lane to which b_j was assigned, a possible lane forward move in that lane, and the lanes that have become empty. Finally, because there can be at most one lane forward move per interval of Δ minutes, it is not possible to perform such a move when EB $b \in B_{j+1}^A$ arrives if the last move occurred in the interval $]a_b - \Delta, a_b[$. This constraint is imposed in RTeBPCDP$_{j+1}$ by setting $v_b^l = 0$ for all $l \in L$ and all such EBs $b \in B_{j+1}^A$.

Once the sliding window procedure is completed, we build the implemented solution $\theta(W)$ from all critical decisions made in each iteration. The objective value $\zeta(W)$ of this solution can then be computed a posteriori taken into account the three components of the weighted objective function (1), namely, the total number of reserve buses used, the total number of lane forward moves, and the total number of service assignment changes with respect to the initial planned services. The full procedure is summarized in Algorithm 1, including the tracking signal prediction error monitoring in Steps 5–10.

Algorithm 1 Sliding-window framework for simulating depot operations.

Require: Ordered arrivals $\hat{B}^A = \{b_1, \ldots, b_m\}$, initial sets B_1^A, B_1^P, S_1, look-ahead width W, control limit $L_0 = \sigma\omega\eta$.
1: $D^+ \leftarrow L_0, \quad D^- \leftarrow -L_0$
2: **for** $j \leftarrow 1$ **to** m **do**
3: Observe the realized consumption of b_j's preceding service and compute ε_{b_j}
4: **if** $j < m$ **then**
5: Update D^+ and D^-
6: **if** $D^+ < 0$ **or** $D^- > 0$ **then**
7: Compute the adjustment factor α
8: **for** $b \in \{b_{j+1}, \ldots, b_m\}$ **do**
9: Adjust the predicted consumption $e^P_{s_b^-}$
10: Update the predicted arrival SoC q_b
11: Reset $D^+ \leftarrow L_0, \quad D^- \leftarrow -L_0$
12: Formulate RTeBPCDP$_j$ and solve model (1)–(23) for B_j^A, B_j^P, and S_j
13: From the obtained solution, identify the critical decisions
14: **if** $j < m$ **then**
15: Compute B_{j+1}^A, B_{j+1}^P, and S_{j+1}
16: Construct implemented solution $\theta(W)$ from all critical decisions
17: Compute objective value $\zeta(W)$

To conclude this section, we present an example in Figure 2 that illustrates the sliding-window framework for six EB arrivals and two available chargers. In this example, the charging rate is set to $\gamma = 0.08$ per 15-minute period, and the SoC values in the figure are rounded to two decimals, so minor rounding deviations may appear. Because recharging is organized in fixed 15-minute periods, an EB arriving within a period could start charging at the next period boundary, while EBs already charging during that period continue until its end. Each RTeBPCDP solved is represented by a time line. The information associated with each EB is provided in a card, topped by a colored triangle to indicate its arrival time and associated with a colored thick horizontal bar to specify its charging schedule. EB1 arrives at 09:10 with an observed SoC of 28% (highlighted in cyan), below the predicted one due to a relative error of $\varepsilon_1 = 3.5\%$ on the energy consumed along its previous service. The optimization assigns EB1 to Lane 1, Spot 4 with a charging period [09:15, 10:15] to reach its next departure at 11:28. When making these decisions, the model considers EBs within a 120-minute look-ahead interval ($W = 120$), namely, EB2, EB3, and EB4, planning their lanes, spots, service assignments, and charging periods, although these decisions remain subject to possible revision.

On the second time line, EB2 arrives at 09:42 with an observed SoC of 45% and relative energy consumption error of $\varepsilon_2 = 3.0\%$. The re-optimization now considers EB3, EB4, and EB5 within the updated look-ahead interval [09:42, 11:42]. EB1's assignment (in the shaded rectangle) remains fixed though its SoC has increased to 44% (in yellow) because it charges for two periods up until 09:45, the start time of the initial period for the corresponding RTeBPCDP. EB2 is assigned to Lane 1, Spot 5 with charging period [10:00, 11:15], extended from the initially planned [10:00, 11:00] to compensate for its lower arrival SoC ensuring sufficient energy for its 12:24 departure. The changes with respect to the previous decisions are highlighted in red, indicating that the anticipated lane/spot assignment for EB4 has been revised by the optimizer for an unknown reason and that EB5 is now assigned a charging period and a lane/spot.

EB3 arrives at 10:15 with a lower than expected SoC (55% instead of 58%), yielded by an observed energy consumption error of $\varepsilon_3 = 8.5\%$. As the successive positive errors $(3.5\%, 3.0\%, 8.5\%)$ accumulate, their cumulative effect becomes large enough to trigger the tracking signal upon the arrival of EB3, which detects a systematic positive bias, indicating that the EBs consistently consume more energy and arrive with less charge than predicted. Consequently, the system adjusts all future EBs' predicted SoCs downward (changes highlighted in brown): EB4's prediction decreases from 38% to 36%, EB5's from 45% to 43%, and EB6's from 41% to 39%. EB3, originally planned for Lane 3, Spot 1

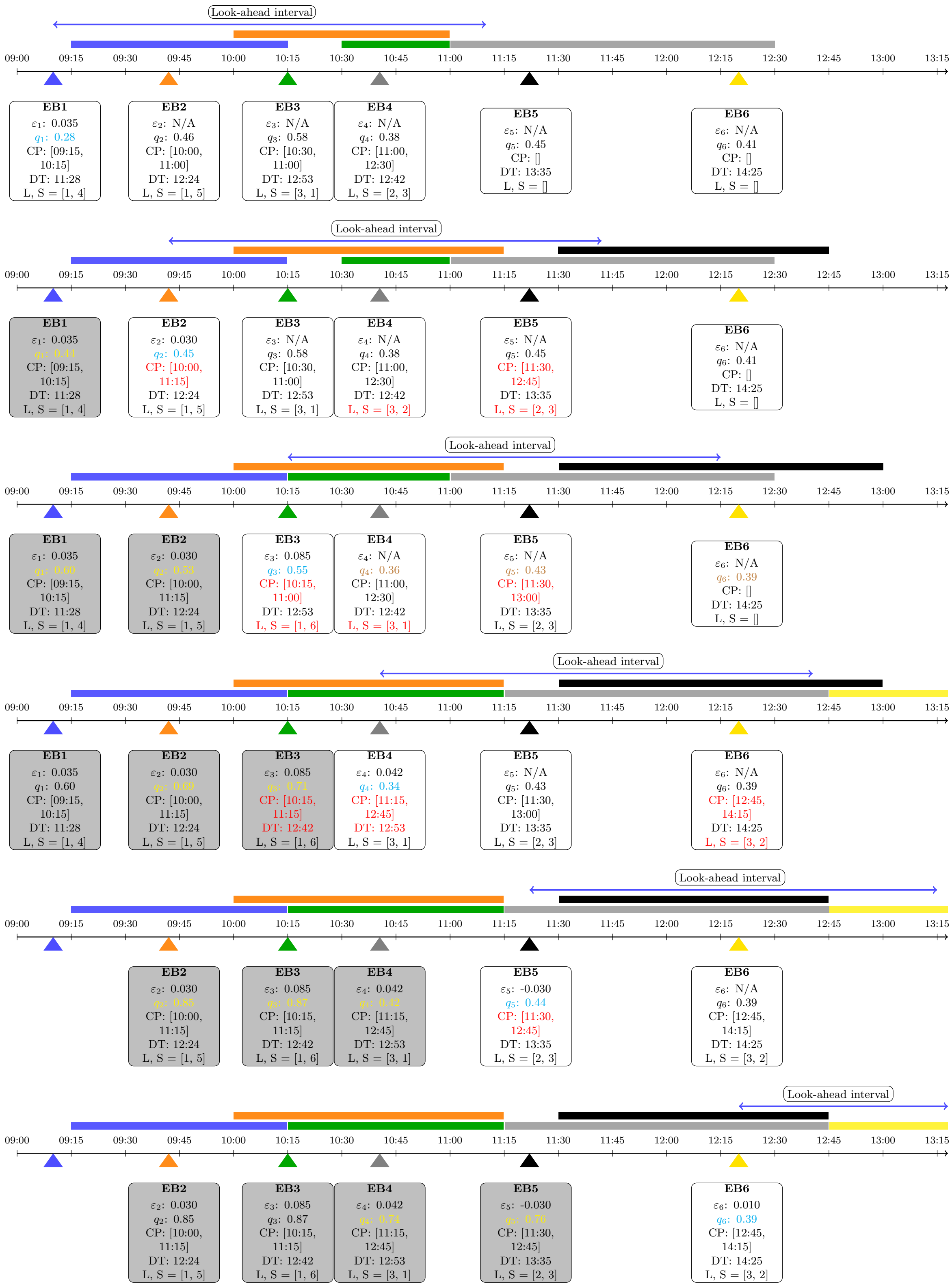

Figure 2: Sliding-window framework for six EB arrivals.

Note: CP: charging period. DT: departure time. L: lane. S: spot.

with charging [10:30, 11:00], is reassigned to Lane 1, Spot 6 with an extended charging [10:15, 11:00] to accommodate its lower arrival charge. EB4 is then re-assigned to Spot 1 in Lane 3. Furthermore, EB5's planned charging session is extended by one period.

On the fourth time line, EB4 arrives at 10:39 with a SoC that is slightly below the adjusted predicted SoC (34% compared to 36%) due to a relative error of $\varepsilon_4 = 4.2\%$ on energy consumption, confirming that the tracking signal procedure helps correcting systematic bias. EB4 is assigned to Lane 3, Spot 1 but, because it arrives with a SoC that is slightly less than the predicted one, it cannot be assigned to the 12:42 departure anymore, as planned in the previous optimization. Its next service is swapped with that of EB3 and the charging periods of both EBs are revised according to their new upcoming services. This demonstrates the framework's ability to revise decisions for parked EBs whose charging is in progress. Note that EB6 appears in the look-ahead interval and, thus, is tentatively assigned a charging period and a lane/spot.

Next, EB5 arrives at 11:22 with an observed SoC of 44% versus adjusted prediction of 43% due to an error of $\varepsilon_5 = -3.0\%$, the first negative error indicating slight over-correction by the tracking signal. EB5 is assigned to Lane 2, Spot 3 with charging [11:30, 12:45], a shorter period sufficient given its higher-than-predicted charge, for its 13:35 departure. Note that EB1 is not represented anymore because it leaves the parking at 11:28 before 11:30, the start time of the initial period.

Finally, EB6 arrives at 12:20 with an actual SoC of 39%, matching the adjusted prediction of 39% ($\varepsilon_6 = 1.0\%$), demonstrating successful calibration with prediction errors varying in the interval $[3.0\%, 8.5\%]$ initially and in the interval $[-3.0\%, 4.2\%]$ after the first three arrivals. EB6 is assigned to Lane 3, Spot 2 with charging [12:45, 14:15] for its 14:25 departure. Throughout this process, the sliding-window framework continuously advances the look-ahead interval while the tracking signal monitors prediction accuracy. The adaptive adjustments successfully compensate for systematic environmental factors affecting battery performance, enabling effective real-time depot management.

7 Computational results

To assess the proposed methodology, we conducted a series of computational experiments on randomly generated test instances. To solve the MILP model (1)–(23), we used *IBM ILOG CPLEX 22.1.1.0* via a C++ implementation and chose the default parameter values, except for the tolerance on the MILP optimality gap which was set to 0.5%. In the branching, we prioritized branching on the reserve bus variables. The goal of this section is to present and analyze the obtained computational results.

This section is organized as follows. Section 7.1 specifies how the test instances were generated, and Section 7.2 defines the solution quality metrics. Section 7.3 presents results for comparing two sliding-window simulation approaches, with and without the tracking signal algorithm. Finally, Section 7.4 evaluates the effect of changing the look-ahead interval length and the number of EBs initially parked (Section 7.4.1), and the number of available chargers (Section 7.4.2) on the objective function value and the computational time.

7.1 Instance generation

For the sliding-window simulation framework, a test instance is composed of a dataset providing the information known before the first EB arrival and an energy consumption realization vector. These two components are generated as follows.

The dataset generator takes as input the configuration $|L| \times n^P$ of a depot, the total number of EBs $|B|$ to consider, a scenario indicating the percentage of initially parked EBs, and a two-hour horizon, discretized in minutes, for the EB arrivals ($T = 120$ minutes). It provides a set of services S, described by their departure time d_s and expected energy consumption e_s, and a set of EBs B that will perform these services. For each parked EB b, it specifies its lane l_b and its location, its planned upcoming service s_b, its SoC q_b at the beginning of the initial period, and its initial charging status (i.e., charged, charging or not yet charged). For each incoming EB b, it indicates its arrival time a_b, its planned upcoming service s_b, its expected SoC q_b upon arrival, and the expected energy consumption $e_{s_b^-}$ along its assigned service s_b^- before arrival.

We have conducted experiments using six depot configurations: $|L| \times n^P \in \{3 \times 8, 3 \times 10, 3 \times 12, 4 \times 6, 4 \times 8, 4 \times 10\}$. To tackle the most challenging instances, we have decided to set $|B|$ equal to 95% of the total number of parking spots in the depot, rounded to the nearest integer, i.e., $|B| = round(0.95 \cdot |L| \cdot n^P)$. To partition the EBs into parked and incoming EBs, we consider three *initial parking scenarios*: $|B^P| = \lfloor 0.75 \cdot |B| \rfloor$, labeled as *parking*, where most EBs are initially in the parking, $|B^P| = \lfloor 0.25 \cdot |B| \rfloor$ (*incoming*) where most of the EBs are incoming in the next two hours, and $|B^P| = \lfloor 0.5 \cdot |B| \rfloor$ (*equal*) where half of the EBs are already parked. The first two scenarios model the depot operations at the start and the end of the day, respectively, while the equal scenario may represent middle-of-the-day operations.

We create a set of services S with $|S| = |B|$. The distribution of the service departure times follows a uniform distribution in the interval $[30, 330]$, assuming that the planning horizon corresponds to the interval $[0, 120]$. For each service s, we determine its energy consumption e_s by drawing a random number uniformly from the interval [0.5, 0.75].

We then place the EBs from B^P in the parking lot such that the parking layout allows the EBs from B^P to handle the $|B^P|$ services with the earliest departure times. We ensure that there is a feasible solution by leaving a sufficiently large number of spots available at the back of the lanes. We guarantee that the EBs scheduled to depart before minute 90 are already sufficiently charged to complete their service. Then, we process the EBs in B^P in the chronological order of their assigned services. For an EB in B^P, if there is still a free charger, we assume that its charging status is ongoing and we randomly generate its initial SoC, ensuring that it can be sufficiently charged to perform its planned upcoming service. If all chargers are used, then the EB is considered not charged and its expected initial SoC is randomly generated to be less than the charge required for its upcoming service but large enough to complete this service after charging for a few periods. Finally, we determine the data for the incoming EBs in B^A. We first assign from S a random service (not yet assigned) to each of these EBs. Then, for each EB, we draw an arrival time in the interval [0, 120] such that it arrives before the departure time of its assigned service. We also randomly select the expected arrival SoC of each EB making sure that it is sufficiently low to require charging. The expected energy consumption $e^P_{s_b^-}$ of the service performed by each arriving EB b before its arrival (in percentage of the battery capacity) is drawn uniformly from the interval $[0.30, \min\{0.75, 1-q_b\}]$, ensuring consistency between the energy consumed along the preceding service and the expected arrival SoC. To capture the stochastic nature of battery consumption under varying environmental conditions, we generate 50 realization vectors per dataset, equally split across five temperature regimes: cold, very cold, normal, hot, and very hot. For each arriving EB b with base expected energy consumption $e^P_{s_b^-}$, the realized energy consumption is drawn as $e^O_{s_b^-} \sim \mathcal{N}(e^P_{s_b^-}(1+\mu),\ \big(0.025\, e^P_{s_b^-}(1+\mu)\big)^2)$, where the mean shift $\mu \in \{0.15, 0.075, 0, -0.075, -0.15\}$ corresponds to the very cold, cold, normal, hot, and very hot regimes, respectively. The realized arrival SoC is then computed as $q^O_b = q_b - (e^O_{s_b^-} - e^P_{s_b^-})$, projected onto $[0, 1]$.

For each dataset, we consider charging periods of $\delta = 15$ minutes, starting at minutes 0, 15, 30, and 45 of every hour, as well as a recharging rate per period of $\gamma = 0.04167$. The maximum number of chargers n^R that can be used simultaneously is equal to $\lceil 0.4 \cdot |B| \rceil$.

7.2 Performance metrics

Based on the objective function (1) considered in the RTeBPCDP model, we report the following metrics to evaluate the quality of the solutions produced through the sliding-window framework:

Reserve Bus Usage (RBU) indicates the number of reserve buses (e.g. diesel buses) deployed when the EB fleet cannot meet service requirements.

Lane Movements (LM) captures the total number of lane forward moves throughout the operational horizon T.

Service Changes (SC) computes the total number of EBs assigned to services different from their initial plan.

Objective Function Value (OFV) synthesizes the above metrics through a weighted combination that reflects real-world priorities. The weights are set to $w_1 = 100$ for the reserve buses used, $w_2 = 10$ for the lane moves, and $w_3 = 1.5$ for the service changes. These weights, calibrated through discussion with our industrial partner GIRO and a sensitivity analysis, ensure that critical failures (reserve bus usage) are prioritized over efficiency improvements (reduced movements) and stability concerns (service changes).

Beyond solution quality metrics, we assess computational performance through multiple dimensions. We consider average computational time per optimization cycle (for each EB arrival) that hints on the applicability of the method in practice. We also track worst-case computational times to assess reliability under all conditions.

The values of the parameters of the tracking signal algorithm, namely, ω, η, β, and ν (introduced in Section 5), were selected through a dedicated parameter tuning procedure conducted prior to the main computational experiments. The tuning was carried out on ten instances of the 3×12 depot configuration under the equal initial parking scenario. For each instance, 1,000 energy consumption realization vectors were generated, with 200 realizations drawn from each of the five temperature regimes (very cold, cold, normal, hot, and very hot), yielding a total of 10,000 instance-realization pairs. We tested all combinations arising from the grids $\omega \in \{0.04, 0.06, 0.08, 0.10, 0.12\}$, $\eta \in \{5, 7, 10, 12, 14\}$, $\beta \in \{0.04, 0.06, 0.08, 0.10, 0.12\}$, and $\nu \in \{0.04, 0.06, 0.08, 0.10, 0.12\}$, for a total of 625 parameter combinations. For each combination, we ran the tracking signal algorithm in simulation, allowing it to observe the realized energy consumption of each arriving EB and to adjust the predicted energy consumptions and arrival SoCs of future EBs accordingly. We recorded the mean absolute error between the adjusted predicted arrival SoC and the realized arrival SoC, averaged over all EBs and all 10,000 instance-realization pairs. The parameter combination minimizing this criterion was selected, yielding $\omega = 0.1$, $\eta = 12.0$, $\beta = 0.04$, and $\nu = 0.1$, which are the values used throughout all computational experiments reported in this paper.

7.3 Tracking signal performance analyses

The first series of computational experiments is designed to evaluate the practical value of applying an adaptive technique to forecast energy consumptions and arrival SoCs in EB operations. The experimental framework encompasses two primary solution approaches: the baseline sliding-window re-optimization approach (ReOpt), which employs static energy consumption and SoC forecasts throughout the planning horizon; and our proposed sliding-window re-optimization method with tracking signal (ReOpt-TS), which dynamically adjusts SoC predictions through adaptive energy consumption forecast. In these experiments, we only consider the equal initial parking scenario where 50% of the EBs are initially parked in the depot and a look-ahead interval width W of 120 minutes, i.e., the interval always contains all forthcoming EBs in the planning horizon. For each of the six depot configurations, we generated 5 datasets and 50 realization vectors per dataset, for a total of 250 test instances for each configuration.

To compare the quality of the solutions produced by ReOpt and ReOpt-TS, we record for each instance, the best solution value obtained by either of these solution methods, and report for each method $m \in \{\text{ReOpt}, \text{ReOpt-TS}\}$ the following average gap:

$$\text{Gap}_m = \frac{\overline{z}_m - \overline{z}_{best}}{\overline{z}_{best}},$$

where $\overline{z}_m$ is the average solution value obtained by method m over the 250 instances and $\overline{z}_{best}$ is the average of the best solution value. We also provide the average OFV gain yielded by ReOpt-TS over ReOpt.

Table 4 presents the average comparative results across all depot configurations. They clearly demonstrate substantial and consistent gains when the tracking signal approach is applied. The most significant improvement appears in reserve bus utilization (the most important objective), with reductions ranging from 12.5% in the 3×12 configuration where average RBU decreases from 0.24 to 0.21, to 55.6% in the 4×10 configuration where average RBU decreases from 0.09 to 0.04. The 4×8 configuration also exhibits a substantial 50% reduction from 0.14 to 0.07. Statistics on average lane move reductions indicate minimal variation (within $\pm 2.6\%$) between the solutions produced by both solution approaches, except 4×10, where ReOpt-TS reduces lane moves by 6.2%. Considering average service changes, we observe that the solutions produced by ReOpt-TS show reductions for most configurations: 5.1% for 3×8, 6.1% for 3×10, 5.8% for 3×12, 4.9% for 4×8, and 9.9% for 4×10, with only 4×6 showing a slight increase of 1.6%.

By examining the correlation with RBU reductions, we observe that in some cases, ReOpt-TS strategically accepts minor increases in LM and SC to achieve significant reductions in the far more costly reserve bus deployments. The average gap with respect to the best solution value reveals great improvements in solution quality when applying the tracking signal procedure, for all depot configurations. The average gap for ReOpt ranges from 34.86% to 62.09%, while ReOpt-TS achieves gaps between 7.96% and 40.90%, yielding average OFV improvements from 7.16% (3×12) to 27.89% (4×8). These improvements demonstrate that more accurate energy consumption predictions enable the system to make better real-time decisions.

Table 4: Average comparative results between Re-Opt and Re-Opt TS.

							OFV	
$\lvert L\rvert \times n^P$	$\lvert B\rvert$	n^R	**Approach**	**RBU**	**LM**	**SC**	**Gap (%)**	**Gain (%)**
3×8	23	10	ReOpt	0.14	0.98	5.53	38.49	–
			ReOpt-TS	0.09	0.97	5.25	15.45	16.64
3×10	28	12	ReOpt	0.26	0.59	5.56	53.70	–
			ReOpt-TS	0.17	0.60	5.22	17.09	23.82
3×12	34	14	ReOpt	0.24	0.81	8.57	51.76	–
			ReOpt-TS	0.21	0.81	8.07	40.90	7.16
4×6	23	10	ReOpt	0.13	0.78	3.86	34.86	–
			ReOpt-TS	0.08	0.80	3.92	7.96	19.94
4×8	30	12	ReOpt	0.14	0.60	4.46	62.09	–
			ReOpt-TS	0.07	0.60	4.24	16.89	27.89
4×10	38	16	ReOpt	0.09	0.65	7.67	48.16	–
			ReOpt-TS	0.04	0.61	6.91	11.62	24.67

Note: Gap compared to best OFV; Gain of OFV compared to ReOpt.

For each depot configuration and solution method, Table 5 reports various computational times: average time per EB (Avg.); worst average time per EB for an instance (Worst Avg.); average over all instances of the worst time per EB (Avg. Worst); worst time per EB (Worst Worst). For real-time operations, the most important figure is probably the Worst Worst time because the Avg. time is biased by the fact that the planning horizon is limited to T minutes and, therefore, the RTeBPCDPs solved for the last arriving EBs become trivial given that they consider a small number of arriving EBs.

Table 5: Computational time results.

$\lvert L\rvert \times n^P$	$\lvert B\rvert$	n^R	Approach	Time (seconds) Avg.	Worst Avg.	Avg. Worst	Worst Worst
3×8	23	10	ReOpt	0.44	1.01	2.40	7.47
			ReOpt-TS	0.43	0.95	2.30	7.40
3×10	28	12	ReOpt	0.50	1.35	3.05	14.42
			ReOpt-TS	0.52	1.34	3.25	14.42
3×12	34	14	ReOpt	1.02	2.27	9.47	30.57
			ReOpt-TS	1.06	3.26	10.08	47.86
4×6	23	10	ReOpt	0.39	0.86	1.63	5.82
			ReOpt-TS	0.39	0.87	1.68	5.88
4×8	30	12	ReOpt	0.54	1.34	2.64	9.17
			ReOpt-TS	0.55	1.24	2.69	10.15
4×10	38	16	ReOpt	1.29	2.88	12.87	34.17
			ReOpt-TS	1.31	3.55	13.49	40.43

Our observations are as follows. First, these results demonstrate that ReOpt-TS introduces no computational overhead despite incorporating tracking signal calculations. This performance parity, combined with the significant solution quality improvements discussed above, indicates that ReOpt-TS outperforms ReOpt for real-time EB depot optimization. Second, the Worst Avg. times are not much larger than the Avg. times, showing that the computational times do not vary much from one instance to another. Third, with Avg. Worst times reaching up to 13.49 seconds for ReOpt-TS (on the 4×10 configuration), this approach seems applicable in real time for the depot sizes and initial parking scenario tested. This claim is reinforced by the Worst Worst times which remain mostly under 15 seconds, with the exception of the 3×12 configuration reaching 47.86 seconds and 4×10 configuration reaching 40.43 seconds. On these two configurations, ReOpt-TS is mildly slower than ReOpt, whose corresponding Worst Worst times are 30.57 and 34.17 seconds. This difference does not originate from the tracking signal computation, which is fast, but from the adjusted predictions occasionally yielding a more difficult MILP instance. Further computational time results for the other two scenarios are provided in the following section.

7.4 Sensitivity analyses

In this section, we focus on evaluating the effect on the ReOpt-TS computational time of changing the look-ahead interval length and the number of initially parked EBs (Section 7.4.1) as well as the number of available chargers (Section 7.4.2).

7.4.1 Varying the look-ahead interval length and initial parking scenario

The goal of this section is to analyze the sensitivity of ReOpt-TS with respect to the initial parking scenario and to the length of the look-ahead interval. To perform this analysis, we have generated 250 test instances (5 datasets and 50 SoC realization vectors per dataset) for each pair of depot configuration and initial parking scenario. Each instance was then solved three times, using three different look-ahead interval lengths: $W = 30, 60, 120$. Since the planning horizon spans $T = 120$ minutes, the longest look-ahead interval encompasses all arriving EBs. However, shorter intervals implement myopic decision-making with limited future visibility.

First, we assess the quality of the solutions computed by ReOpt-TS using each interval length $W \in \{30, 60, 120\}$. To do so, we use, as in the previous section, the average OFV gap with respect to the value of the best solution obtained by these three settings. The average results for each initial parking scenario and for all of them together (column all) are reported in Table 6. The averages over all depot configurations are also provided in the last three rows. These results reveal compelling evidence of the superiority of considering extended look-ahead intervals. Indeed, the average OFV gap obtained with $W = 120$ over all scenarios is 26.47% compared to 75.56% for $W = 60$ and 155.12% for $W = 30$, which is definitely too short to anticipate a sufficient number of forthcoming EBs for yielding good real-time decisions. Observe that the variance of the average OFV is, in general, relatively small across all scenarios (last three rows) and across all depot configurations (last column). Finally, note that the large gaps are typically yielded by a relatively modest increase in the primary objective function term, namely, RBU. For instance, an average gap of around 100% may come from a change in the average RBU from 0.2 to 0.4.

Table 6: Average OFV gap for different look-ahead intervals and initial parking scenarios.

				Average OFV gap (%)			
$\vert L\vert \times n^P$	$\vert B\vert$	n^R	W	**parking**	**equal**	**incoming**	**all**
3×8	23	10	30	92.69	132.48	247.08	157.42
			60	54.70	54.92	70.35	59.99
			120	53.38	22.82	19.29	31.83
3×10	28	12	30	23.78	232.47	216.15	157.47
			60	20.11	98.99	84.90	68.00
			120	32.98	45.77	30.43	36.39
3×12	34	14	30	98.81	224.11	149.58	157.5
			60	82.95	137.51	62.80	94.42
			120	44.43	43.75	14.13	34.10
4×6	23	10	30	37.19	136.61	195.28	123.03
			60	17.00	103.12	72.73	64.28
			120	0.62	36.06	16.95	17.88
4×8	30	12	30	59.54	142.21	335.14	178.96
			60	47.57	64.04	110.24	73.95
			120	17.48	18.55	7.15	14.39
4×10	38	16	30	165.61	134.51	168.85	156.32
			60	80.59	105.83	91.77	92.73
			120	44.23	12.07	16.35	24.22
Average			30	79.60	167.06	218.68	155.12
			60	50.49	94.07	82.13	75.56
			120	32.19	29.84	17.38	26.47

Table 7 presents the computational time results for the same instances (see the description of Table 5 for the meaning of each time statistic). For all statistics, we observe that the time increases with the interval length W and with the number of arriving EBs, that is, from the parking to the incoming scenario, as the number of initially parked EBs decreases, and that this increase is more pronounced for the largest depot configurations. For instance, for the incoming scenario, using $W = 120$ instead of $W = 30$ increases the Worst Worst time by a factor 1.97 for the 4×8 depot configuration, while it increases it by a factor 28.6 for the 3×8 setting. This time increase is simply due to a larger size of the MILP model to solve at each iteration of the sliding-window procedure. Indeed, considering a longer look-ahead interval or a smaller number of initially parked EBs yields a larger number of arriving EBs and, thus, larger numbers of variables and constraints in the model.

While the Avg. Worst times (averages over all instances of the worst time per EB) may suggest feasibility for real-time deployments for the parking and equal initial parking scenarios when using $W = 120$, the extreme cases for the incoming scenario (Worst Worst times which may exceed several hours for the largest configurations) reveal potential failures that could compromise real-time operations. It is important to note that in these experiments, the MILP solver was given a generous time limit of 10 hours and a tight optimality tolerance of 0.5%, as the primary goal was to assess the quality of the solutions produced by our framework rather than to enforce real-time computational constraints. In a real-time deployment, a much shorter time limit (e.g., 60 seconds) would be imposed, and the solver would return the best feasible solution found within that limit. These extreme outliers are, moreover, statistically rare as they occur in a few percent of realizations based on our standard deviation analysis.

Table 7: Computational times for different look-ahead intervals and initial parking scenarios.

$\|L\| \times n^P$	$\|B\|$	n^R	Scenario	W	Time (seconds) Avg.	Worst Avg.	Avg. Worst	Worst Worst
3 × 8	23	10	parking	30	0.21	0.50	0.28	1.73
				60	0.21	0.34	0.28	0.74
				120	0.22	0.36	0.31	0.59
			equal	30	0.23	0.34	0.39	0.87
				60	0.30	0.68	0.98	4.40
				120	0.44	0.93	2.28	8.22
			incoming	30	0.30	0.91	0.76	4.87
				60	0.48	4.97	2.09	63.98
				120	1.11	8.35	11.8	139.33
3 × 10	28	12	parking	30	0.22	0.36	0.31	0.55
				60	0.26	0.46	0.37	1.22
				120	0.33	0.82	0.87	3.99
			equal	30	0.25	0.43	0.44	2.17
				60	0.31	0.49	0.73	2.58
				120	0.55	1.76	3.45	18.88
			incoming	30	0.35	3.90	1.34	62.10
				60	0.87	17.41	8.72	317.58
				120	0.87	6.10	8.12	64.37
3 × 12	34	14	parking	30	0.24	0.38	0.36	0.64
				60	0.26	0.45	0.40	1.09
				120	0.33	0.47	0.75	1.46
			equal	30	0.35	1.19	0.98	14.31
				60	0.55	7.15	2.57	105.93
				120	1.09	3.23	10.46	40.64
			incoming	30	1.00	22.57	11.42	568.64
				60	18.05	1593.56	345.61	28972.54
				120	11.00	422.97	238.51	10986.20
4 × 6	23	10	parking	30	0.18	0.35	0.24	0.68
				60	0.20	0.36	0.26	1.14
				120	0.21	0.41	0.32	1.12
			equal	30	0.19	0.27	0.30	0.61
				60	0.30	0.83	0.74	3.51
				120	0.37	0.84	1.61	6.34
			incoming	30	0.30	1.15	1.00	11.99
				60	0.70	18.92	5.34	312.28
				120	39.67	3264.59	588.51	36003.40
4 × 8	30	12	parking	30	0.23	0.68	0.33	4.31
				60	0.26	0.67	0.43	1.15
				120	0.30	0.73	0.68	2.41
			equal	30	0.25	0.71	0.50	2.47
				60	0.40	1.15	1.79	11.48
				120	0.55	1.69	3.08	14.97
			incoming	30	0.65	15.18	5.08	186.30
				60	7.25	943.68	134.11	17372.13
				120	1.82	30.99	18.48	367.79
4 × 10	38	16	parking	30	0.27	0.40	0.40	0.66
				60	0.32	0.52	0.49	0.86
				120	0.52	1.80	2.05	14.27
			equal	30	0.36	0.77	0.98	8.27
				60	0.49	0.93	1.56	5.60
				120	1.25	2.68	12.74	36.08
			incoming	30	1.41	34.48	23.51	951.17
				60	22.02	1353.51	523.16	30812.85
				120	14.35	557.69	334.10	14263.50

In practice, depot operators can select the length of the look-ahead interval based on their specific operational requirements. Based on the above computational performance analysis, a 120-minute interval is appropriate for the parking and equal scenarios, where computational times remain uniformly low while solution quality is highest. For the incoming scenario on the larger depot configurations, no look-ahead width consistently controls the worst-case time, namely, a 60-minute interval is not always faster than a 120-minute interval and is in fact considerably slower in several configurations (for instance, on the 4×8 configuration, the Worst Worst time is 17,372.13 seconds for $W = 60$ against 367.79 seconds for $W = 120$). In these cases, the worst-case time is best controlled by imposing a solver time limit, as discussed above, while retaining $W = 120$ for its superior solution quality. Alternatively, instead of defining the look-ahead interval by a duration, it could be defined using a fixed number of incoming EBs to consider in this interval of varied length.

7.4.2 Varying the charging infrastructure capacity

In this section, we evaluate the ReOpt-TS computational performance for different charging infrastructure levels to determine how the charging capacity affects the real-time optimization process. We conducted a final series of computational experiments on a new set of instances that was solved using ReOpt-TS with $W = 120$. For each of the same six-depot configuration, we generated 250 instances (5 datasets and 50 SoC realization vectors per dataset) for the equal initial parking scenario and the number of available chargers previously used ($n^R = \lceil 0.4 \cdot |B| \rceil$). Then, we create two other instance sets by duplicating twice the initial 250 instances, but considering $n^R = \lceil 0.3 \cdot |B| \rceil$ and $n^R = \lceil 0.5 \cdot |B| \rceil$.

The results (the same time statistics as above) presented in Table 8 show a clear trend across all depot configurations. Indeed, for all configurations, increasing n^R from $\lceil 0.4 \cdot |B| \rceil$ to $\lceil 0.5 \cdot |B| \rceil$ has a minor impact on the reported times, whereas reducing it to $\lceil 0.3 \cdot |B| \rceil$ causes the computational times to increase significantly by orders of magnitude. Looking at the Avg. Worst times, we can notice that for $n^R = \lceil 0.3 \cdot |B| \rceil$, they range from 13 to 99 seconds, while for $n^R \geq \lceil 0.4 \cdot |B| \rceil$, they remain no more than 10.1 seconds across all configurations. A similar observation can be made from the Worst Worst times, where the $n^R = \lceil 0.3 \cdot |B| \rceil$ instances reach values of several thousand seconds for most configurations (up to 12,507.67 seconds for 3×8), the only exception being the 4×10 configuration, which remains at 53.49 seconds. Consequently, when charging capacity is already tight, reducing the

Table 8: Computational times for varying charging infrastructure capacity.

			Time (seconds)			
$\|L\| \times n^P$	$\|B\|$	n^R	**Avg.**	**Worst Avg.**	**Avg. Worst**	**Worst Worst**
3×8	23	7	11.55	1602.85	98.79	12507.67
		10	0.48	1.37	2.00	5.85
		12	0.46	1.48	1.85	5.24
3×10	28	9	6.81	859.50	85.17	11997.82
		12	0.47	1.29	2.52	11.67
		14	0.46	1.38	2.47	11.07
3×12	34	11	3.77	224.03	45.79	3689.24
		14	0.86	2.24	8.01	32.29
		17	0.81	2.57	7.19	37.57
4×6	23	7	4.20	150.38	39.39	1760.35
		10	0.34	0.71	1.32	4.57
		12	0.33	0.67	1.22	3.87
4×8	30	9	5.01	468.00	63.10	6996.01
		12	0.44	1.04	2.12	7.69
		15	0.43	0.94	2.04	8.25
4×10	38	12	1.19	3.19	13.34	53.49
		16	1.00	2.52	10.10	30.39
		19	0.96	2.16	9.47	28.45

number of chargers increases complexity of solving the problem and the computational times. In fact, in this case, more reserve buses are typically required, inducing larger optimality gaps and larger branch-and-bound trees. On the other hand, when it is not tight, the same reduction has a negligible impact on the computational time.

8 Conclusion and future research

In this paper, we have addressed the real-time EB parking, charging, and dispatching problem (RTeBPCDP) under FIFO parking constraints, energy consumption and SoC forecast uncertainty, and limited charging infrastructure. We formulated the problem as a MILP that explicitly minimizes operational costs arising from reserve bus usage, lane forward moves, and service changes. Besides this MILP, our main contribution is the development of a tracking signal algorithm, based on cumulative sum control charts, that dynamically adjusts energy consumption predictions, and consequently arrival SoC forecasts, to compensate for environmental uncertainties.

We implemented a sliding-window framework in which critical decisions are made upon each EB arrival, using only short-term information about future arrivals. Computational experiments conducted across six representative depot configurations and 1,500 stochastic realizations showed that our adaptive approach reduces reserve bus deployments by up to 55.6% and improves overall solution quality by up to 27.9% relative to static forecasting, with comparable average computational times. Sensitivity analyses further showed that solution quality improves with longer look-ahead intervals, while computational tractability is most sensitive to the look-ahead interval length and the availability of charging infrastructure, both of which depot operators can tune to their specific operational requirements. Together, these results confirm the practicality of our approach for real-time deployment under realistic operating conditions.

There are several promising directions for future research. First, extending the model to heterogeneous fleets, with diverse EB dimensions, charging profiles, and lane compatibilities, would enhance its applicability to modern transit systems. Second, integrating trip scheduling decisions with depot operations would further strengthen the model's relevance, but would potentially introduce substantial mathematical and computational challenges. Third, developing more efficient solution algorithms is crucial for tackling larger instances where current methods may become computationally intractable. Finally, embedding more sophisticated stochastic representations of arrival times, battery degradation, and charging dynamics (nonlinear charging functions) within the optimization process could further improve the resilience and reliability of EB depot management. By bridging adaptive prediction with real-time optimization, this study lays a foundation for more efficient, robust, and sustainable electric transit operations.

References

M. Azéma, G. Desaulniers, J. Mendoza, and G. Pesant. A constraint programming model for the electric bus assignment problem with parking constraints. In 21th International Conference on the Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming (CPAIOR 2024), Lecture Notes in Computer Science, volume 14742, pages 17–33, 2024.

M. Azéma, G. Desaulniers, J. Mendoza, and G. Pesant. A mixed-integer linear programming model for the electric bus assignment problem with parking constraints. 2025.

R. G. Brown. Statistical Forecasting for Inventory Control. McGraw–Hill, 1959.

Z. Chao and C. Xiaohong. Optimizing battery electric bus transit vehicle scheduling with battery exchanging: Model and case study. Procedia - Social and Behavioral Sciences, 96:2725–2736, 2013.

N. Dirks, M. Schiffer, and G. Walther. On the integration of battery electric buses into urban bus networks. Transportation Research Part C, 139:103628, 2022.

E. S. Gardner. Automatic monitoring of forecast errors. Journal of Forecasting, 2(1):1–21, 1983.

E. S. Gardner. CUSUM vs smoothed-error forecast monitoring schemes: Some simulation results. Journal of the Operational Research Society, 36(1):43–47, 1985.

M. Hamdouni, G. Desaulniers, O. Marcotte, F. Soumis, and M. Putten. Dispatching buses in a depot using block patterns. Transportation Science, 40(3):364–377, 2006.

M. Hamdouni, G. Desaulniers, and F. Soumis. Parking buses in a depot using block patterns: A Benders decomposition approach for minimizing type mismatches. Computers & Operations Research, 34(11):3362–3379, 2007a.

M. Hamdouni, F. Soumis, and G. Desaulniers. Parking buses in a depot with stochastic arrival times. European Journal of Operational Research, 183(2):502–515, 2007b.

P. Harrison and O. Davies. The use of cumulative sum (CUSUM) techniques for the control of routine forecasts of product demand. Operations Research, 12(2):325–333, 1964.

International Energy Agency. Electric Vehicles. https://www.iea.org/energy-system/transport/electric-vehicles, 2023. Accessed: 21 August 2025.

P. Jaramillo, S. Kahn Ribeiro, P. Newman, S. Dhar, O. E. Diemuodeke, T. Kajino, D. S. Lee, S. B. Nugroho, X. Ou, A. Hammer Strømman, and J. Whitehead. Transport. In R. Shukla, P. J. Skea, R. Slade, A. Al Khourdajie, R. van Diemen, D. McCollum, M. Pathak, S. Some, P. Vyas, R. Fradera, M. Belkacemi, A. Hasija, G. Lisboa, S. Luz, and J. Malley, editors, Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022. doi: 10.1017/9781009157926.012.

P. S. Klein and M. Schiffer. Electric vehicle charge scheduling with flexible service operations. Transportation Science, 57(6):1605–1626, 2023.

B. Krishnamurthy. A comparison of the relative efficiency of tracking signals in forecast control. Masters' thesis, West Virginia University, 2006.

P.-L. Lin, F. Zheng, M. Shin, X. Liu, D. Oh, and D. D'Attilio. CUSUM learning curves: what they can and can't tell us. Surgical Endoscopy, 37(10):7991–7999, 2023.

B. Messaoudi and A. Oulamara. Electric bus scheduling and optimal charging. In C. Paternina-Arboleda and S. Voß, editors, Computational Logistics, Lecture Notes in Computer Science, volume 11756, pages 233–247. Springer, Cham, 2019.

N. Olsen and N. Kliewer. Scheduling electric buses in public transport: Modeling of the charging process and analysis of assumptions. Logistics Research, 13:4, 2020.

S. Otto and J. Breitung. Backward CUSUM for testing and monitoring structural change with an application to COVID-19 pandemic data. Econometric Theory, 39(4):659–692, 2023.

S. Perumal, R. Lusby, and J. Larsen. Electric bus planning and scheduling: A review of related problems and methodologies. European Journal of Operational Research, 301(6):395–413, 2022.

O. Sassi and A. Oulamara. Electric vehicle scheduling and optimal charging problem: Complexity, exact and heuristic approaches. International Journal of Production Research, 55(2):519–535, 2017.

C. Superville. Tracking signal performance in monitoring manufacturing processes. Journal of Business and Management, 21:23–28, 2019.

M. van Kooten Niekerk, J. van den Akker, and J. Hoogeveen. Scheduling electric vehicles. Public Transport, 9:155–176, 2017.

P. Vendé, G. Desaulniers, Y. Kergosien, and J. E. Mendoza. Matheuristics for a multi-day electric bus assignment and overnight recharge scheduling problem. Transportation Research Part C: Emerging Technologies, 156:104360, 2023.

A. Zhang, T. Li, Y. Zheng, X. Li, M. Abdullah, and C. Dong. Mixed electric bus fleet scheduling problem with partial mixed-route and partial recharging. International Journal of Sustainable Transportation, 16(1): 73–83, 2022.

L. Zhang, S. Wang, and X. Qu. Optimal electric bus fleet scheduling considering battery degradation and non-linear charging profile. Transportation Research Part E, 154:102445, 2021.