Minimum-delay opportunity charging scheduling for electricbuses

Citation for the original published paper (version of record):
McCabe, D., Ban, X., Kulcsár, B. (2025). Minimum-delay opportunity charging scheduling for electricbuses. Communications in Transportation Research, 5.
http://dx.doi.org/10.1016/j.commtr.2025.100209

Contents lists available at ScienceDirect

Communications in Transportation Research

journal homepage: www.journals.elsevier.com/communications-in-transportation-research

Full Length Article

Minimum-delay opportunity charging scheduling for electricbuses

Dan McCabe[a], Xuegang (Jeff) Ban[a], Balaźs Kulcsár[b,*]

[a] Department of Civil and Environmental Engineering, University of Washington, Seattle, WA, 98195- 2700, USA
[b] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, SE-412 96, Sweden

ARTICLE INFO

Keywords:
Battery-electric bus
Combinatorial Benders decomposition
Layover charging
Opportunity charging
Heuristics

ABSTRACT

Transit agencies that operate battery-electric buses must carefully manage fast-charging infrastructure to extend daily bus range without degrading on-time performance. To support this need, we propose a mixed-integer linear programming model to schedule opportunity charging that minimizes the amount of departure delay in all trips served by electric buses. Our novel approach directly tracks queuing at chargers in order to set and propagate departure delays. Allowing but minimizing delays makes it possible to optimize performance when delays due to traffic conditions and charging needs are inevitable, in contrast with existing methods that require charging to occur during scheduled layover time. To solve the model, we develop two algorithms based on decomposition. The first is an exact solution method based on combinatorial Benders (CB) decomposition, which avoids directly enumerating the model's logic-based "big M" constraints and their inevitable computational challenges. The second, inspired by the CB approach but more efficient, is a polynomial-time heuristic based on linear programming that we call Select–Sequence–Schedule (3S). Computational experiments on both a simple notional transit network and the real bus system of King County, Washington, USA demonstrate the performance of both methods. The 3S method appears particularly promising for creating good charging schedules quickly at real-world scale.

1. Introduction

Battery-electric buses (BEBs) make up a significant and growing share of the global transit vehicle fleet (Peng et al., 2021; Tingstad Jacobsen et al., 2023; Wu et al., 2021; Zhou et al., 2025). Over 60,000 such vehicles were sold worldwide in 2022, about 5% of global bus sales (IEA, 2023). While China has been the world leader in electric bus adoption for many years, BEBs are beginning to see greater usage worldwide. In USA, the Bipartisan Infrastructure Law of 2021 allocated over US $5 billion to help agencies purchase low-emissions transit vehicles and charging infrastructure (Federal Transit Administration, 2022). Worldwide, BloombergNEF projects that 50% of buses will be battery-powered by 2032, a milestone passenger cars are not expected to reach for a further ten years (Stock, 2023).

Compared to conventional transit buses, BEBs create new management challenges for transit agencies because of their limited driving range and lengthy recharging times, which may be handled with a few different strategies. The preferred approach for many agencies is to rely primarily on low-power overnight charging at bus bases (known as depot charging), which mimics traditional bus operating patterns in which refueling is not a significant concern. However, many agencies also plan to use high-power chargers during the day to effectively extend bus ranges without utilizing a larger battery (King County Metro Transit, 2022; Los Angeles County Metropolitan Transportation Authority, 2021; Massachusetts Bay Transportation Authority, 2021). This approach is commonly referred to as opportunity charging and sometimes on-route or layover charging. Throughout this work, we focus on these agencies that plan to use depot charging as much as possible, but extend vehicle range using opportunity charging where necessary.

Fig. 1 displays how the charging scheduling problem addressed in this study fits with the overall transit planning and operations process. Network design, timetabling, vehicle scheduling, and operator scheduling are interrelated problems that are typically solved sequentially rather than jointly because of their complexity (Ceder, 2007; Desaulniers and Hickman, 2007). BEBs complicate this process by introducing decisions about vehicle fleet composition (including fuel types and battery sizes, if applicable) and charging infrastructure (such as the number, power output, and location of chargers), but also by increasing the complexity of existing decisions. For example, the BEB vehicle scheduling problem should incorporate new features, such as vehicle

[*] Corresponding author.
E-mail address: kulcsar@chalmers.se (B. Kulcsár).

https://doi.org/10.1016/j.commtr.2025.100209
Received 16 January 2025; Received in revised form 8 April 2025; Accepted 9 April 2025
Available online xxxx

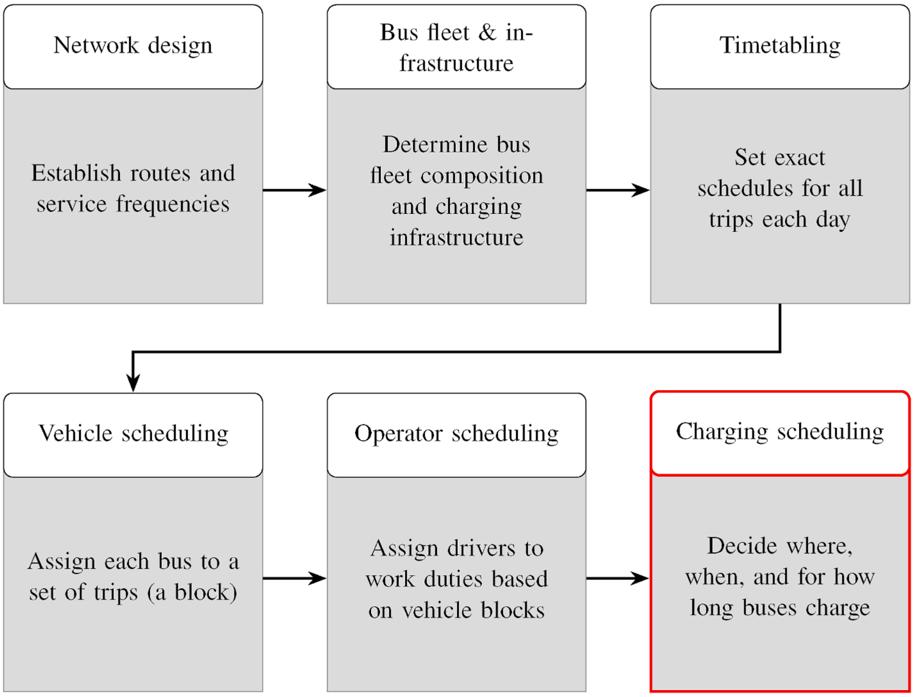

Network design
Establish routes and service frequencies
Bus fleet & infrastructure
Determine bus fleet composition and charging infrastructure
Timetabling
Set exact schedules for all trips each day
Vehicle scheduling
Assign each bus to a set of trips (a block)
Operator scheduling
Assign drivers to work duties based on vehicle blocks
Charging scheduling
Decide where, when, and for how long buses charge

Fig. 1. Relationship between charging scheduling and the complete transit planning and operations process.

range limitations and opportunity charging, that were not relevant for diesel buses (Perumal et al., 2022).

In this study, we view charging scheduling as an independent problem that should be solved each day in response to updated predictions of weather, traffic, ridership, and other factors that impact vehicle energy consumption and on-time performance. These daily variations can be predicted much more accurately on a given day of service than they can when vehicle schedules are produced (typically months in advance), and their values significantly impact the optimal decisions about when and for how long each BEB should charge during the service day. Although daily adjustments to upstream decisions such as vehicle schedules could potentially result in even better performance, these are typically fixed well in advance and cannot be revised frequently because of complex labor agreements with drivers. For example, the current labor contract between King County Metro and the Amalgamated Transit Union (Amalgamated Transit Union, Local 587, 2022) stipulates that operator duties will be selected twice per year. Because vehicle schedules (blocks) are an important input to this process, it follows that they must be determined about six months in advance.

Designing charging schedules for each day necessitates a flexible approach that can handle a wide variety of operating conditions. In the current literature on opportunity charging scheduling, a ubiquitous assumption is that trips must run according to schedule (Abdelwahed et al., 2020; Bao et al., 2023; Liu et al., 2022), so opportunity charging must take place during scheduled layover time in between passenger trips. This assumption is appropriate for long-term planning, but inconsistent with real daily operations in which delays are pervasive; for example, Washington Metropolitan Area Transit Authority (WMATA) in Washington, D.C. and Maryland Transport Authority (MTA) in Maryland both report bus on-time performance well under 80% (Maryland Transit Administration, 2024; Washington Metropolitan Area Transit Authority, 2024). Agencies that wish to employ data-driven planning based on historical operating conditions rather than published schedules will find such models inappropriate, since a solution in which even a single trip is delayed a single instant would be considered infeasible. A more flexible approach would enable planning for such conditions and other unfavorable scenarios, such as when chargers are out of service or buses consume an unusually large amount of energy.

This work proposes a novel modeling approach to opportunity charging scheduling that allows, but tracks and minimizes, departure delays across all trips. We do this by explicitly modeling the queuing process at charging stations to track exactly when each bus is ready to start each of its trips. Our model also propagates any delays across trips served by the same bus, ensuring their cascading impacts are captured accurately. Second, our focus on accurate delay tracking makes the model applicable to a wide range of operating conditions, including worst-case scenarios. Rather than constraining all trips to leave on time, our model handles scenarios where zero delay is not achievable—for example, when delays are inevitable due to traffic conditions, or when buses consume enough energy that scheduled layover time is not sufficient to keep their batteries charged. It produces actionable results regardless of the ultimate delay and extends naturally to stochastic applications where energy consumption and traffic conditions can make on-time departures impossible.

Our precise modeling approach produces some computational challenges. In particular, the mixed-integer linear programming model includes many binary variables and "big M" constraints. To mitigate this issue, we develop both an exact solution method and a polynomial-time heuristic algorithm. Both approaches decompose the complete problem into a series of easier-to-solve problems with easily interpretable structure. The exact method uses combinatorial Benders (CB) decomposition to split the problem into a master problem that contains all binary variables and a subproblem with all continuous variables, which reduces the burden of the big M constraints. This restructuring is natural for our problem: The binary variables reflect high-level decisions about the ordering of charging throughout the day, whereas the continuous variables track the exact timing of events, including charging times and trip departures. The CB approach exploits this structure in an iterative framework.

Our Select–Sequence–Schedule (3S) heuristic algorithm was inspired by the CB approach and is based on a similarly explainable decomposition. As in CB, we make binary decisions first and continuous decisions second. To expedite this process, 3S initially relaxes the queue tracking constraints that link different buses to each other, so that we can select when charging occurs with a separate linear program for each bus. We then establish the sequence in which buses visit each charger with a simple sorting operation and finally schedule the exact charger plugin times, charging durations, and delays with the same approach as in the CB subproblem. The 3S method has polynomial-time complexity in the worst case, so it can be used to solve real-world instances quickly. Our experiments show that despite its lack of a performance guarantee, 3S reliably identifies good feasible solutions (and often optimal solutions) much faster than exact approaches do.

In summary, the main contributions of this study are as follows:

1) A mixed-integer linear programming model for recharging scheduling that exactly tracks queuing behavior at chargers, propagates delays across trips completed by the same bus, and minimizes the total delay;
2) An exact solution method based on CB decomposition to solve this computationally demanding model;
3) A polynomial-time heuristic method, motivated by the exact solution method, which generates good solutions quickly for complex real-world transit networks.

The remainder of this paper is organized as follows. Section 2 provides a review of relevant literature. Section 3 describes our methodology and mixed-integer programming formulation. Section 4 describes our exact solution method based on CB decomposition, while Section 5 documents the related heuristic. Section 6 applies our methods to both a simple notional transit network and the real Seattle-area bus network. Section 7 summarizes findings and concludes the study.

2. Literature review

2.1. Recharging scheduling for BEBs

Recharging scheduling for BEBs connects to a larger transit planning process that designs routes, timetables, and vehicle and driver schedules. In traditional transit planning with diesel buses, these problems are addressed sequentially rather than simultaneously, as each step is already complex to optimize (Ceder, 2007). Replacing diesel buses with BEBs further complicates the traditional transit planning process by introducing new challenges related to capital investments in buses and chargers, location of chargers, possible schedule revisions, and charging scheduling (Perumal et al., 2022). Because these various stages of the BEB planning process have significant overlap, opportunity charging scheduling has been studied as a standalone problem as well as through its integration with other decisions.

One common context in which researchers have modeled opportunity charging scheduling is when designing blocks for BEBs, which for diesel buses is classically addressed with the vehicle scheduling problem (VSP). Although the basic single-depot VSP can be solved easily for conventional buses, extensions including multiple depots, battery range limitations, multiple vehicle types, and recharging planning make VSP modifications for BEBs quite difficult to model and solve. Because of this complexity, when charging scheduling is incorporated into a BEB VSP model, significant assumptions are usually made that limit the applicability for day-to-day scheduling. For example, Tang et al. (2019) and Wu et al. (2022) assumed charging always takes a fixed amount of time. Bie et al. (2021) took a more flexible approach that calculates and minimizes departure delays, but is focused on scheduling buses that serve only a single route, not a complete BEB system.

The comprehensive review of BEB VSP publications provided by Perumal et al. (2022) further highlights the difficulty of incorporating detailed charging scheduling models into vehicle scheduling. For example, among the 23 publications reviewed in Perumal et al. (2022), none simultaneously considered multiple vehicle types and partial recharging of batteries. Both of these attributes are essential for daily recharging scheduling—many transit agencies operate both standard-length and articulated buses, for example, and partial recharging is a natural strategy for large batteries—but present major computational challenges to incorporate in a model that also designs blocks.

Similar limitations are seen in the many publications that focus on designing opportunity charging infrastructure and incorporate a charge scheduling element. For example, Esmaeilnejad et al. (2023) proposed a stochastic optimization model to determine both charger locations and recharging schedules along a single bus route, where charging may take place at intermediate stops but is penalized based on passenger waiting costs. However, this model assumes buses will always charge to 100% state of charge between a pair of consecutive trips, which is only appropriate for buses with small battery capacities. McCabe and Ban (2023) developed a deterministic model for simultaneously locating chargers and scheduling charger usage. Charger capacity is accounted for with predetermined “conflict sets” based on posted timetables, which could result in unexpected queuing at chargers if buses are delayed. Gairola and Nezamuddin (2023) proposed a comprehensive robust model to minimize the costs of converting to BEBs by optimizing battery sizes, charging infrastructure, and charging schedules. Their model allowed recharging at terminals during scheduled layover time and handled capacity by discretizing time into unit-length intervals and ensuring the number of buses charging at a terminal did not exceed the number of chargers there, similar to McCabe and Ban (2023). However, neither of these approaches extends well to daily operations where unusually high energy demand or exogenous delays might make it impossible for buses to stay on schedule and charge only within scheduled layover time.

On the other hand, developing charging schedules for each day of operations after vehicle schedules and charging infrastructure are fixed allows for a more detailed and flexible approach. This strategy also allows transit agencies to adapt to daily variations that impact energy consumption and travel times, including weather and traffic conditions, passenger load, and unplanned disruptions such as charger downtime. However, many notable charge scheduling models from the literature still make restrictive assumptions, while others simply take an approach not appropriate for our problem setting. Many works including Abdelwahed et al. (2020), Bao et al. (2023), Liu et al. (2022), and Zeng et al. (2022) still constrained charging to occur during scheduled layover times that are treated as fixed. He et al. (2023) was even more restrictive, requiring each bus to charge every 3 trips regardless of the route served rather than tracking battery state of charge. He et al. (2020) did disregard the number of available charging stations. Authors assumed that each bus route had its own dedicated charger with sufficient plugs always available for charging.

Other charging scheduling research is tailored to different applications than our intended setting. For example, Lacombe et al. (2024) studied an optimal control problem intended to minimize both energy costs and schedule deviations, where the bus schedule is encoded as a route-specific headway rather than a timetable. They developed a decomposition strategy based on Lagrangian relaxation and local heuristics to apply the method in practice. This approach is much more flexible than restricting charging to scheduled layover time, but is only appropriate for systems where maintaining a target headway is more important than matching an advertised schedule. Finally, we note that there are various works such as Brinkel et al. (2023) and Manzolli et al. (2022) that focused exclusively on optimizing depot charging. This is a fundamentally different challenge from our setting that is focused on opportunity charging at terminals (possibly including, but not limited to, depots), so we do not review those in detail here.

2.2. CB decomposition

Most approaches to optimal charge scheduling result in a mixed-integer programming problem, making them difficult to solve. We handle this challenge using a tailored version of Benders decomposition, a classical algorithm for mixed-integer programming that was first proposed over 60 years ago (Benders, 1962). The central idea of this approach is to separate the problem into a master problem (MP) containing integer variables and a subproblem (SP) containing continuous variables. In each iteration, the MP is solved to obtain candidate values of all integer variables. These fixed values are an input to the SP, which verifies the feasibility and potential optimality of the candidate solution. If the SP determines that the current candidate solution cannot be an optimal solution, one or more Benders cuts are generated and added to the MP to exclude it (along with, ideally, many other solutions), then the MP is solved again. The procedure repeats until an optimal solution is found (Rahmaniani et al., 2017).

Benders decomposition is often used in stochastic programming applications, where after fixing a limited number of first-stage variables, the second-stage problem decomposes into several independent problems that can be solved quickly. However, in the last two decades, researchers have identified additional classes of problems where the approach can outperform standard branch-and-bound or branch-and-cut algorithms. Hooker and Ottosson (2003) introduced the idea of logic-based Benders decomposition, wherein cuts are generated not based on the linear programming dual of the subproblem but a so-called "inference dual" that generalizes LP duality. Their approach has been applied successfully to a variety of problems but appears especially effective in cases where specialized constraint programming methods can be applied to the subproblem, including some types of scheduling problems (Hooker, 2007).

Codato and Fischetti (2006) soon after developed the idea of CB cuts for a specific class of mixed-integer linear programs with logical constraints. Their approach largely follows Hooker and Ottosson (2003), but they derived a tailored method for finding cuts for their particular problem class and demonstrated its performance benefits on some example problems. A major selling point of this method is that it eliminates the computational problems caused by big M constraints; once variable values have been set by the MP, the corresponding logical constraints are either included or excluded from the SP. As such, the CB approach can avoid the usual problem of big M values giving a poor linear programming relaxation and resulting bounds.

It should be noted that a direct application of the Benders (standard or combinatorial) algorithm can have poor performance for a variety of reasons. These issues and strategies to mitigate them are reviewed thoroughly in Rahmaniani et al. (2017). The keys to a successful Benders implementation include initializing the MP with a set of strong cuts to aid in finding feasible solutions; using heuristics to generate good solutions and strong Benders cuts; and embedding Benders cuts within a branch-and-cut algorithm to reduce redundant computations. These strategies were essential in making the CB approach competitive with an off-the-shelf solver.

2.3. Summary

The worldwide growth of BEBs has produced a significant body of literature on charging scheduling. However, most of these approaches are more suited to long-range planning than day-to-day or real-time scheduling. When charging scheduling is embedded within a model primarily focused on designing a charging infrastructure network or vehicle blocks, it tends to be simplified and restrictive. Even models focused purely on charge scheduling such as Abdelwahed et al. (2020) and Liu et al. (2022) are only applicable under ideal conditions, since they assume buses run on schedule and charging can take place within scheduled layover time. There is a need for models that can provide useful output even under challenging conditions when avoiding departure delays is impossible, so that bus operators are provided with actionable instructions rather than being told a problem instance is infeasible. Such models should be accompanied with efficient algorithms so they can be run repeatedly as conditions evolve or forecasts of ridership, traffic, and weather conditions improve.

Based on this research gap, we propose one such model for charging scheduling. Our approach permits but penalizes departure delays, so that buses can recharge as much as necessary whenever they reach a charger. Like existing models, our approach is applicable to ideal situations in which buses stay on schedule, but also generalizes to more difficult real-world conditions—for example, when buses' energy demands are unusually high, traffic conditions create exogenous delays, or charging stations are out of service and queues form at available chargers. We enable this flexibility by precisely quantifying queue delays at chargers and propagating delays across trips. The optimization model and solution methods are presented in Sections 3–5.

3. Mathematical programming formulation

3.1. Problem setting, assumptions, and modeling approach

We consider a general setting in which fast chargers with predetermined power outputs have already been installed at some terminals of a BEB system. Fig. 2 shows a simple illustration of the type of transit network considered in this study. In this basic example, two bus routes ($\mathbb{A}$ and $\mathbb{B}$ where $\mathbb{A}, \mathbb{B} \in \mathbb{P}$, i.e. they belong to a set of routes $\mathbb{P}$) operate across two terminals and each terminal has a single charger installed. A bus may recharge at either of the two charging sites when out of service in between trips, as long as the bus is at the corresponding terminal and the charger is not already occupied. Note that in our approach, a single bus may serve more than one route (referred to as interlining) and/or use multiple different chargers during a day. A single charger also may serve buses on any number of different routes, so that infrastructure is shared as efficiently as possible. Bus trips may take place on a one-way route between distinct terminals (i.e., Route $\mathbb{A}$) or a loop route that starts and ends at the same terminal (i.e., Route $\mathbb{B}$).

Our approach relies on the following assumptions:

1) Buses rely on slow overnight charging at the depot in addition to fast charging during the day. Each bus has a known state of charge when it enters service at the start of the day. A minimum state of charge is also required to return to the depot at the end of the day.
2) Charger locations, trip schedules, and vehicle blocks have been determined in advance and cannot be altered. The agency's primary goal is to adhere to the posted schedule as much as possible, i.e., to minimize delays.
3) Buses may only charge at terminals when passengers are not onboard, never at intermediate stops.
4) Deadheading to chargers does not require significant time or energy. Buses may only use chargers that are sufficiently close to a trip terminal that the driving distance and time can be neglected.
5) All buses in the network have enough battery capacity to complete all trips between charges.
6) Charging behavior is linear, proportional to the maximum power output of each charger.

Assumption 1 acknowledges the different technologies, time scales, and constraints faced by transit agencies during depot versus opportunity charging. Agencies can be expected to have different priorities for charging scheduling in these different environments. When buses use opportunity chargers during the service day, maintaining schedule adherence is the greatest priority, whereas overnight charging should

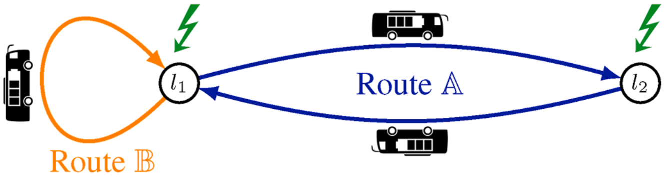


Fig. 2. Simple example of the type of BEB system considered in this study.

allow more time to optimize for other objectives such as energy costs and battery health. Bus bases are also less likely to be subject to queuing for chargers due to the much lower cost of lower-power chargers.

Assumption 2 reflects our intended problem setting as described in Section 1. Assumption 3 ensures that the current passenger level of service is maintained. Assumption 4 is common in the recharging literature and simplifies the model by making cumulative energy consumption independent of any charging decisions and their corresponding binary decision variables. Its implications for using the model in practice are discussed further in Section 7.

The last two assumptions concern battery range and charging behavior. Assumption 5 is necessary to ensure operations are always feasible with fast chargers; any such infeasible blocks can be discarded from the analysis. Assumption 6 is another common assumption made to limit computational complexity. It can provide a good enough approximation even if the real charging behavior is nonlinear, since charging is typically linear until the state of charge exceeds about 80% when power must be reduced to protect the battery (Montoya et al., 2017).

Following these assumptions, suppose we are given a set of BEBs $\mathcal{V}$, each of which is scheduled to complete a specific set of trips; that is, every vehicle has been assigned to a predetermined block. We represent each trip as a tuple $i = (b_i, n_i)$, where b_i is the bus ID and n_i is the trip number; i.e., bus A completes trips $(A, 1)$, $(A, 2)$, $(A, 3)$, etc. Let $\mathcal{T}$ be the set of all trips completed by any bus. We will use the shorthand indices i and $j \in \mathcal{T}$ throughout this formulation to limit indices, but keep in mind that each of these refer to a specific trip completed by a specific bus as part of a predetermined block. For each trip $i \in \mathcal{T}$, the timetable provides a scheduled departure time σ_i and scheduled duration τ_i (so that the scheduled end time is $\sigma_i + \tau_i$). If a trip i has an immediate predecessor in its block, that predecessor is denoted i^-; likewise, the next trip in its block is denoted i^+. Let $\theta(i)$ be the set of trips completed by the same bus prior to i (i.e., all of its predecessor trips).

Fig. 3 illustrates a simple example of how to interpret these parameters on a timeline for two buses labeled A and B. The colored segments of the timeline represent times when the buses are scheduled to be in service and unavailable to charge. Considering the trips served by bus B in this example, we have $\theta((B, 3)) = \{(B, 1), (B, 2)\}$, $\theta((B, 2)) = \{(B, 1)\}$, and $\theta((B, 1)) = \emptyset$. Likewise, $(B,3)^- = (B, 2)$ and $(B,1)^+ = (B, 2)$.

Our approach to charging scheduling focuses on accurately tracking trip departure delays, which is challenging because of the underlying queuing behavior. To address this challenge, we define multiple continuous decision variables for each trip that track the time of certain events. For each trip i, let d_i be the departure delay, so that the actual start time of the passenger trip is $d_i + \sigma_i$. Let p_i be the time that charging begins after trip i, i.e., the "plugin" time when bus b_i connects to a charger. Note that p_i may be equal to the end time of i if bus b_i can plug in immediately, or may be later if b_i must queue. By convention, if there is no charging after trip i, we set the value of p_i to be the completion time of trip i, i.e., $p_i = d_i + \sigma_i + \tau_i$. For every trip i and every charger l in the set of chargers $\mathcal{C}$, let t_i^l be the amount of time spent charging after trip i at charger l.

To summarize, each trip is scheduled to begin at σ_i, actually starts at time $d_i + \sigma_i$, ends at time $d_i + \sigma_i + \tau_i$, begins charging at time p_i, and finishes charging at time $p_i + \sum_{l\in\mathcal{C}} t_i^l$, at which point the next trip can begin.

Fig. 4 further illustrates the relationship between the problem variables and parameters, continuing the example established in Fig. 3. Suppose that, following the routes from Fig. 2, Bus A completes a block consisting of one-way trips between the terminals that host chargers l_1 and l_2. The first trip starts at l_1 and ends at l_2, the second starts at l_2 and ends at l_1, and so on. Bus B serves a loop route in which each trip starts and ends at l_1.

Buses A and B each complete their initial trips according to schedule. Bus A uses charger l_2 immediately after trip $(A, 1)$, shown in green on its timeline, charging from time $p_{A1} = \sigma_{A1} + \tau_{A1}$ until $p_{A1} + t_{A1}^{l_2}$. It then begins trip $(A, 2)$ on time. Likewise, Bus B plugs in at charger l_1 immediately after trip $(B, 1)$. After trip $(A, 2)$, bus A also uses charger l_1, but cannot begin charging until bus B is finished at time $p_{A2} = \sigma_{B2}$. Trip $(A, 3)$ starts later than scheduled because of the time spent queuing and charging, incurring the departure delay $d_{A3} = p_{A2} + t_{A2}^{l_1} - \sigma_{A3}$ shown on Bus A's timeline.

Table 1 compiles all the set, parameter, and decision variable definitions used in this work. Sections 3.2–3.7 next describe the formulation of the mixed-integer linear program we developed for recharging scheduling, including the objective function and the constraints that track queuing, delays, and battery levels.

3.2. Objective function

$$\min_{d_i, p_i, t_i^l, x_i^l, y_{ij}^l} \sum_{i\in\mathcal{T}} d_i \tag{1}$$

The objective (1) is to minimize the total amount of departure delay across all trips $i \in \mathcal{T}$ served by BEBs. While the objective is straightforward, setting the appropriate value of d_i for each trip i requires carefully formulated constraints, as described in Sections 3.3–3.6.

3.3. Queue tracking and delay propagation

In order to accurately track departure delays for each trip so that their sum can be minimized in objective (1), our model needs to correctly capture the relationships between plugin time, charging time, and delay that were illustrated in Fig. 4. Setting each of these variables' values correctly ensures that queuing effects are captured and delays for each trip are properly quantified and propagated across trips.

First, consider the plugin time p_i for each trip i. Recall our convention that if bus b_i does not visit a charger after trip i, then p_i should be equal to the time that passenger trip i is completed. Likewise, if b_i uses a charger after trip i, the earliest it can start charging is the completion time of trip i. Constraints (2) use the fact that trip i ends at time $d_i + \sigma_i + \tau_i$ to establish this lower bound on the plugin time:

$$p_i \geq d_i + \sigma_i + \tau_i \qquad \forall i \in \mathcal{T} \tag{2}$$

The plugin time p_i should exceed the trip end time $d_i + \sigma_i + \tau_i$ if bus b_i has to queue at the charger. To handle these queuing relationships, we must introduce some auxiliary binary variables and related constraints. Let $\mathcal{C}$ be the set of all chargers that are installed. We say that a charger $l \in \mathcal{C}$ serves a trip $i \in \mathcal{T}$ if b_i uses l after finishing trip i. Let y_{ij}^l be a binary

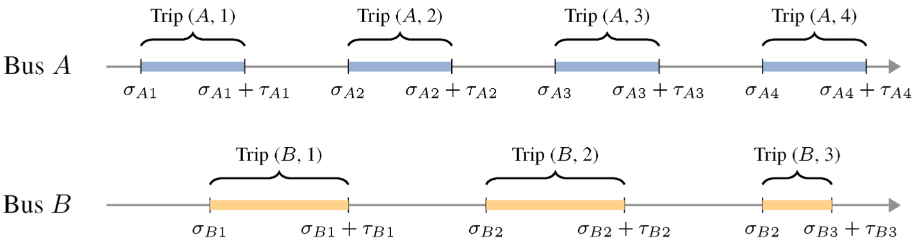


Fig. 3. Timeline of passenger service for two example buses and relationship to trip time parameters.

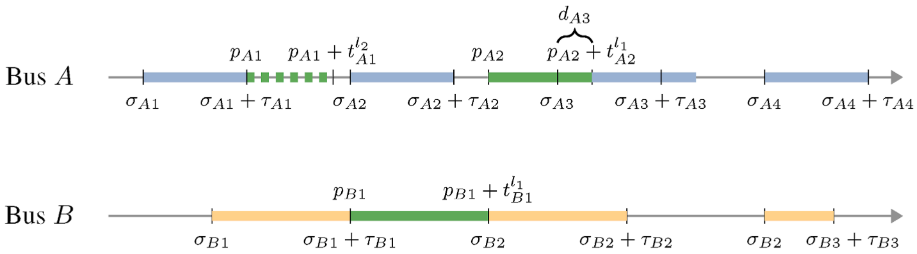


Fig. 4. Illustration of variable meanings for d_i, t_i^l, and p_i with their relation to schedule parameters σ_i and τ_i. The dashed green line segment represents charging at l_2, solid green segments indicate charging at l_1, and other segments show passenger trips.

Table 1
Notation definition.

Set	
$\mathcal{V}$	Buses
$\mathcal{T}$	Trips
$\mathcal{C}$	All chargers
$\mathcal{C}(i)$	Chargers that can be used after trip i
$\mathcal{A}$	Arcs
$\mathbb{P}$	Route
Decision variable	
x_i^l	Binary variable indicating whether the bus on trip i uses charger l afterwards
y_{ij}^l	Binary variable indicating whether charging arc (l, i, j) is used
d_i	Delay at start of trip i
p_i	Plugin time of charging after trip i
t_i^l	Charging duration at l after completion of trip i
Parameter	
b_i	ID of bus that completes trip i
n_i	Trip number of trip i
$n_i^{\max}$	Total number of trips served by bus b_i
ρ^l	Power output of charger l
$\overline{t_i^l}$	Maximum charging duration at site l after trip i
σ_i	Scheduled start time of trip i
τ_i	Scheduled duration of trip i
δ_i	Energy consumed during trip i
ϵ_i	Battery capacity of bus serving trip i
η_i^0	Initial state of charge of bus serving trip i
$\eta_i^{\min}$	Minimum feasible state of charge of bus serving trip i
$\eta_i^{\max}$	Maximum feasible state of charge of bus serving trip i
M	“Big M”, arbitrary large number
c_{ij}^l	CB master problem objective coefficient
ϕ	CB subproblem solution improvement tolerance
α_i^l	Random cost coefficient in 3S heuristic

decision variable that equals 1 if charger l serves trip j immediately after trip i and 0 otherwise. By “immediately”, we mean only that trips i and j are served successively by charger l and no other trips are served in between these two; there may be a large gap between the charging completion time for trip i and the plugin time of trip j. Finally, let $\mathcal{A}$ be the set of all “arcs” (l, i, j) connecting trips i and j that may be served sequentially by charger l. These arcs are discussed in more detail in Section 3.4.

If $y_{ij}^l = 1$ for some arc (l, i, j), then bus b_j cannot start charging until b_i has finished, since the charger can serve only a single bus at a time. This relationship can be stated with the conditional logic of constraint (3):

$$y_{ij}^l = 1 \Longrightarrow p_j \geq p_i + t_i \qquad \forall (l,i,j) \in \mathcal{A} \tag{3}$$

The logical relationship in constraint (3) can be linearized using the common big M method (where M is some large constant), yielding constraints (4):

$$p_j \geq p_i + t_i^l - M(1 - y_{ij}^l) \qquad \forall (l,i,j) \in \mathcal{A} \tag{4}$$

With the plugin time values controlled by constraints (2) and (4), it is straightforward to set the delay for each trip. After each trip i, bus b_i has finished charging and is ready to start its next trip at time $p_i + \max_{l \in \mathcal{C}} \{t_i^l\}$. Note that this is true whether or not any charging is done. The delay of the following trip i^+ is the (nonnegative) difference between this actual start time and the scheduled start time σ_{i^+}, which leads to constraints (5) and (6):

$$d_{i^+} \geq p_i + t_i^l - \sigma_{i^+} \qquad \forall l \in \mathcal{C}, i \in \mathcal{T} : n_i \leq n_i^{\max} - 1 \tag{5}$$

$$d_i \geq 0 \qquad \forall i \in \mathcal{T} \tag{6}$$

Together, constraints (2) and (4)–(6) make sure all plugin time and delay relationships are captured accurately. They also propagate delays across trips—the delay of each trip d_i impacts its plugin time p_i, which impacts the delay of the following trip, d_{i^+}.

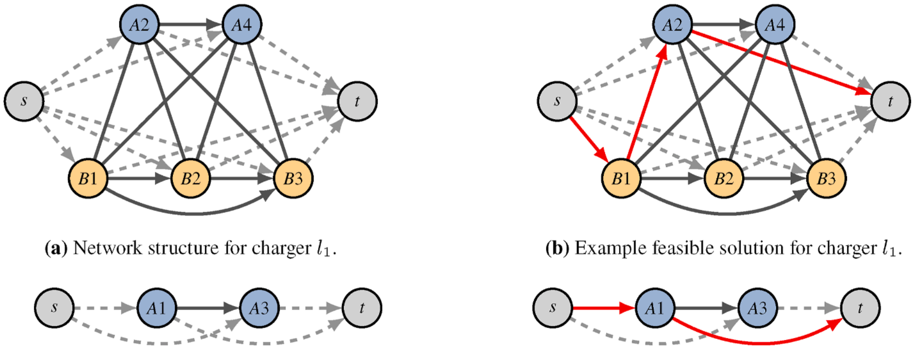
3.4. Charger sequencing

The binary variables y_{ij}^l explicitly track the sequence of trips that are served by each charger in the network. Our model therefore includes constraints to ensure that the optimal values of y_{ij}^l encode a valid sequence connecting all trips served by each charger. By a valid sequence, we mean that (1) any time a charger is used (i.e., $x_i^l = 1$), that trip appears somewhere in the sequence, and (2) every trip i served by charger l has exactly one trip before and one trip after it in the sequence.

To formulate these constraints, we model the charging sequence for each charger l as a path through a network in which each node corresponds to a trip (a charging opportunity) and arcs represent feasible charging connections. That is, if trip j can be served by the charger immediately after trip i, then arc (l, i, j) is included in the network. In general, a directed arc joins every pair of edges for the same bus (since we can’t travel back in time to charge at, say, trip 1 after trip 3). On the other hand, an undirected edge joins any pair of trips completed by different buses, because we do not know a priori which order these trips will be optimally served in. For instance, it could be optimal to use arc (l, i, j) in the solution even if trip j is scheduled to end before trip i, because trip j is delayed in the optimal solution.

We also introduce two dummy nodes to model the initial (node s) and final (node t) idle state of each charger. These nodes are necessary because the first and last trips to be served by each charger have no predecessor and successor trip, respectively, so they require slightly different constraints. Moreover, there is no way to know a priori which trips will be first and last, so we instead construct dummy nodes to handle these special cases. Now, a feasible sequence for charger l corresponds to a path through this virtual network from s to t. Note that this network model is similar to and inspired by those used in vehicle scheduling approaches (e.g., the maximum flow formulation of the VSP (Ceder, 2007)).

Fig. 5 illustrates our network model for the simple example of two buses and two chargers previously depicted in Figs. 3 and 4. Fig. 5a and c displays the network structure for each charger, including the trip and dummy nodes for each charger as well as the arc set $\mathcal{A}$. Arrows indicate directed arcs and lines indicate undirected edges. Fig. 5b and d highlight the paths corresponding to the charging decisions from Fig. 4, in which



(a) Network structure for charger l_1.

(b) Example feasible solution for charger l_1.

(c) Network structure for charger l_2.

(d) Example feasible solution for charger l_2.

Fig. 5. Network representations and feasible paths through them for example of two chargers serving two buses, as in Fig. 4.

bus A uses charger l_2 after trip $(A, 1)$, bus B uses l_1 after trip $(B, 2)$, and bus A charges at l_1 after trip $(A, 2)$. These charging decisions correspond to arc variable values of $y^{l_1}_{sB1} = y^{l_1}_{B1A2} = y^{l_1}_{A2t} = y^{l_2}_{sA1} = y^{l_2}_{A1t} = 1$, and $y^l_{ij} = 0$ for all other arcs.

Following this network representation, a feasible sequence for each charger can be enforced using the familiar network flow constraints (7)–(9):

$$\sum_{i\in\mathcal{T}} y^l_{si} = 1 \qquad \forall l \in \mathcal{C} \tag{7}$$

$$\sum_{i\in\mathcal{T}} y^l_{it} = 1 \qquad \forall l \in \mathcal{C} \tag{8}$$

$$\sum_{j:(l,j,i)\in\mathcal{A}} y^l_{ji} - \sum_{j:(l,i,j)\in\mathcal{A}} y^l_{ij} = 0 \qquad \forall l \in \mathcal{C}, \forall i \in \mathcal{T} \tag{9}$$

Constraints (7) and (8) ensure that exactly one arc leaves the source dummy node and exactly one arc arrives at the sink dummy node, respectively, for each charger. Constraint (9) ensure connectivity of the charging sequence; the number of arcs entering and leaving each trip node must be equal. Notably, the network flow constraints (7)–(9) do not preclude the existence of subtours in paths through the network. It is not necessary to include subtour elimination constraints because any solution containing a subtour cannot possibly be optimal for our problem; in fact, it would result in unbounded delay due to constraints (4) and (5).

3.5. Charging logic

If multiple chargers are located close to the end of trip i, then each bus must be restricted to only use one charger at a time. Let $x^l_i = 1$ if charger l serves trip i and 0 otherwise. Then this condition is simple to enforce with constraint (10):

$$\sum_{l\in\mathcal{C}} x^l_i \le 1 \qquad \forall i \in \mathcal{T} \tag{10}$$

For constraint (10) to be applied as intended, the logical relationship between the x^l_i and y^l_{ij} variables also needs to be enforced. Charging after trip i corresponds to visiting its trip node in Fig. 5, meaning one arc leaves (or, equivalently, enters) that node, giving constraint (11):

$$x^l_i = \sum_{j:(l,i,j)\in\mathcal{A}} y^l_{ij} \qquad \forall l \in \mathcal{C}, i \in \mathcal{T} \tag{11}$$

3.6. State of charge management

Finally, the model needs constraints that track charging throughout the service day to make sure that each bus maintains a feasible state of charge (SOC). Let ϵ_i be the full battery capacity in kWh of bus b_i that serves trip i. Suppose we are also given lower and upper bounds $\eta_i^{\min}$ and $\eta_i^{\max}$ on b_i's battery state of charge, e.g., $\eta_i^{\min} = 0.2$ and $\eta_i^{\max} = 0.95$, which may be imposed to protect the battery or provide reserve capacity. Let η_i^0 be the initial SOC of bus b_i at the start of its first trip.

The SOC of each bus throughout the day depends on its energy consumption in service as well as recharging decisions. Let δ_i be the amount of energy consumed by bus b_i during trip i, in kWh. Let ρ^l be the power output in kW of charger l. Then, the cumulative energy consumption of bus b_i at the end of trip i is $\delta_i + \sum_{j\in\theta(i)}\delta_j$ and the cumulative amount of energy gained is $\sum_{j\in\theta(i)}\sum_{l\in\mathcal{C}}\rho^l t^l_j$. Constraint (12) then enforce that b_i is above the minimum SOC at the end of each trip i:

$$\eta_i^0\epsilon_i + \sum_{j\in\theta(i)}\left(\sum_{l\in\mathcal{C}}\rho^l t^l_j - \delta_j\right) - \delta_i \ge \eta_i^{\min}\epsilon_i \qquad \forall i \in \mathcal{T} \tag{12}$$

The left-hand side of constraint (12) gives the battery level in kWh at the time trip i is finished, while the right-hand side gives the minimum required battery level in kWh. Likewise, the similar constraint (13) ensures that the maximum battery SOC is not exceeded. Note that the left-hand side of constraint (13) gives the battery level immediately after any charging that follows trip i.

$$\eta_i^0\epsilon_i + \sum_{j\in\theta(i)}\sum_{l\in\mathcal{C}}\left(\rho^l t^l_j - \delta_j\right) - \delta_i + \sum_{l\in\mathcal{C}}\rho^l t^l_i \le \eta_i^{\max}\epsilon_i \qquad \forall i \in \mathcal{T} \tag{13}$$

The continuous charging duration variables t^l_i also must follow the correct logical connection to the binary variables x^l_i used in constraints (10) and (11): t^l_i must be forced to zero if $x^l_i = 0$; otherwise, it can take any feasible value. Let $\overline{t}^l_i$ be the maximum possible charging time at charger l after trip i. Then, the charging time t^l_i is restricted by constraint (14):

$$0 \le t^l_i \le \overline{t}^l_i x^l_i \qquad \forall l \in \mathcal{C}, i \in \mathcal{T} \tag{14}$$

The charging time upper bound $\overline{t}^l_i$ should reflect the access to charger l as well as the maximum charging duration, which could be set to a constant by a transit agency policy or limited only by the maximum time it could take to fully charge the battery. This work sets the charging time based on Eq. (15):

$$\overline{t}^l_i = \begin{cases} \epsilon_i\left(\eta_i^{\max} - \eta_i^{\min}\right)/\rho^l & \text{if trip } i \text{ ends at } l \\ 0 & \text{otherwise} \end{cases} \qquad \forall l \in \mathcal{C}, i \in \mathcal{T} \tag{15}$$

Equation (15) reflects that bus b_i can only use charger l if trip i ends at l. If l is accessible to b_i after trip i, then the maximum charging duration is given by the time it would take to recharge the battery's full feasible state of charge at the rated power of charger l.

3.7. *Complete MILP formulation*

Compiling the objective function and full set of constraints developed in Sections 3.2–3.6, the complete mixed-integer linear programming problem for opportunity charging scheduling is given below as optimization problem (P):

$$
\begin{array}{lll}
\min\limits_{d_i,p_i,t_i^l,x_i^l,y_{ij}^l} & \text{Total delay (1)} & \\
\text{s.t.} & \text{Plugin time constraints (2) and (4)} & \\
 & \text{Delay tracking constraints (5) and (6)} & \\
 & \text{Charger sequencing constraints (7)–(9)} & \\
 & \text{Charging logic constraints (10) and (11)} & \\
 & \text{State of charge management constraints (12)–(14)} & \\
 & x_i^l \in \{0,1\} \qquad \forall l \in \mathcal{C}, i \in \mathcal{T} & \\
 & y_{ij}^l \in \{0,1\} \qquad \forall (l,i,j) \in \mathcal{A} &
\end{array}
\qquad \text{(P)}
$$

4. Exact solution method: CB decomposition

The optimization problem (P) is difficult to solve for large instances, largely due to the binary variables y_{ij}^l and the corresponding big M constraint (4). Because the number of y variables and big M constraints scales with the square of the number of trips, solving the problem with an off-the-shelf MIP solver is not possible for many large instances. We develop two strategies to deal with this computational challenge. In this section, we describe an exact solution approach based on CB decomposition, which circumvents the typical issues of weak linear programming relaxations caused by big M constraints. Section 5 later describes a polynomial-time heuristic with randomization that can generate a large number of good solutions quickly. The heuristic helps accelerate the convergence of the CB algorithm on smaller problems and also shows good performance as a standalone method for difficult problems on real networks where exact algorithms are unacceptably slow.

4.1. *Overview of CB decomposition*

Fig. 6 shows a high-level overview of the CB algorithm we implemented. We begin by using a randomized heuristic (described in detail in Section 5) to generate a set of feasible charging schedules. The best solutions are used to create initial CB cuts to restrict the MP's feasible region and we further strengthen the MP by adding cuts derived from the state-of-charge constraint (12). Following this initialization step, the main CB loop begins. In each iteration, we first solve the MP to get a candidate solution in terms of the binary x and y variables only. Assuming we obtain one, we solve the subproblem, which aims to find a solution that outperforms the current incumbent. If the SP is infeasible, indicating that this solution cannot outperform the incumbent, we use the CB cut generation procedure to exclude it. If it is feasible, our current MP solution becomes the new incumbent. We update it as such, then re-solve the SP and add CB cuts.

Sections 4.2–4.4 next describe the individual steps of our CB implementation in detail, including the MP and SP formulations as well as cut generation.

4.2. *Master problem*

The CB master problem formulation includes all constraints on the

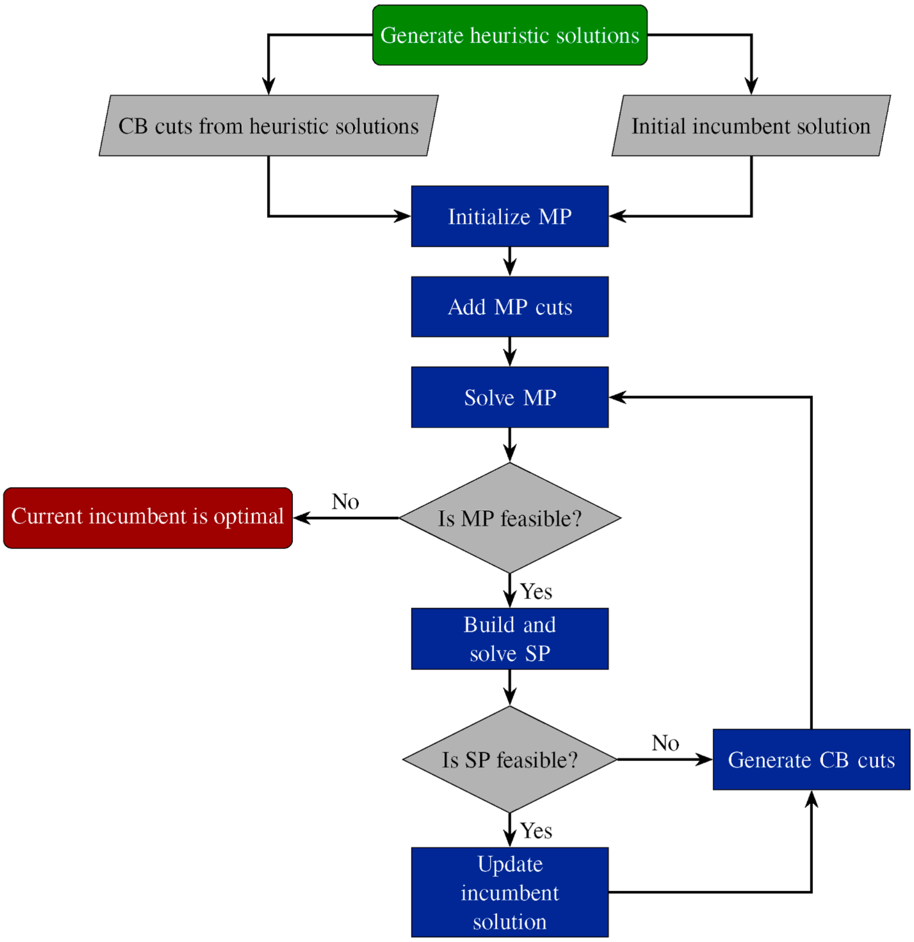

Generate heuristic solutions
CB cuts from heuristic solutions
Initial incumbent solution
Initialize MP
Add MP cuts
Solve MP
Current incumbent is optimal
No
Is MP feasible?
Yes
Build and solve SP
Is SP feasible?
No
Generate CB cuts
Yes
Update incumbent solution

Fig. 6. Overview of CB solution process.

binary variables x_i^l and y_{ij}^l from the complete formulation. Let c_{ij}^l be an objective function cost coefficient, which is discussed in more detail later in this section. Then the MP is given by

$$\min_{x_i^l, y_{ij}^l} \sum_{(l,i,j)\in\mathcal{A}} c_{ij}^l y_{ij}^l \tag{16}$$

$$\begin{aligned} \text{s.t.} \quad & \text{Charger sequencing constraints (7)–(9)} \\ & \text{Charging logic constraints (10) and (11)} \\ & \sum_{(i,j)\in\mathcal{A}(\mathcal{S})} y_{ij}^l \leq |\mathcal{S}| - 1 \qquad \forall \mathcal{S}\subset\mathcal{T} : |\mathcal{S}| \geq 2 \end{aligned} \tag{17}$$

$$\begin{aligned} & x_i^l \in \{0,1\} \quad \forall l \in \mathcal{C}, i \in \mathcal{T} \\ & y_{ij}^l \in \{0,1\} \quad \forall (l,i,j) \in \mathcal{A} \end{aligned}$$

Note that the MP introduces new constraint (17), which ensures that the MP solution does not contain any subtours. In constraint (17), $\mathcal{A}(\mathcal{S})$ denotes the set of arcs contained in the trip subset $\mathcal{S}$, i.e.,

$$\mathcal{A}(\mathcal{S}) = \{(l,i,j) \in \mathcal{A} : i \in S, j \in S\} \tag{18}$$

Subtour elimination constraints were not necessary in the complete problem formulation (P) because any subtour would produce unbounded delay. However, in the CB framework, delay calculation is delegated to the subproblem, so the master problem will frequently generate solutions with subtours if constraint (17) is omitted. We generate these constraints "lazily" within a branch-and-cut framework by checking for subtours and adding constraints only when subtours are detected, but having to handle these constraints directly in the MP is still a disadvantage of the CB method for our problem.

Because the full problem objective function (1) is not dependent on the binary variables, the MP essentially has a constant objective. In practice, this means that we may select any objective we wish, so we use a heuristic objective function to encourage the MP to generate integer solutions that are more likely to result in optimal delays as calculated by the subproblem. To do so, we define cost coefficients c_{ij}^l for all arcs $(l,i,j) \in \mathcal{A}$ as the lower bound on delay that would result from using that arc.

This lower bound is derived from the queuing and delay constraints. Recall that j^+ is the successor trip of j on the same block served by bus j. If arc (l, i, j) is used, then trip j^+ might be delayed due to bus b_j queuing while b_i charges. Specifically, $p_j \geq p_i$ by (4) and because $t_i^l \geq 0$. Accounting for constraint (2), we have $p_j \geq \sigma_i + \tau_i$, and then $d_{j^+} \geq \sigma_i + \tau_i - \sigma_{j^+}$. So, we set the MP cost for most arcs according to Eq. (19):

$$c_{ij}^l = \max\{0, \sigma_i + \tau_i - \sigma_{j^+}\} \tag{19}$$

All dummy arcs (Fig. 5) and any arcs (l, i, j) for which j has no successor trip are assigned a cost of 0. Setting the costs with Eq. (19) disincentivizes the solver from setting $y_{ij}^l = 1$ when doing so is guaranteed to delay trip j^+. In our experiments, this objective function gave better performance than a constant objective that essentially chose feasible solutions at random.

4.2.1. Master problem cuts

When initializing the MP, we also add some cuts to strengthen its formulation. Adding cuts helps to generate MP solutions that are more likely to be feasible for the SP, which decreases the total number of iterations that must be performed.

We generate MP cuts based on the minimum state-of-charge constraint (12), using them to derive lower bounds on the total number of times each bus must charge. Essentially, we convert constraint (12) to restrict the MP decision variables x_i^l rather than the continuous variables t_i^l, which are now delegated to the subproblem. These cuts are based in large part on valid inequality (20):

$$\rho^l t_i^l \leq \epsilon_i\left(\eta_i^{\max} - \eta_i^{\min}\right) x_i^l \qquad \forall i \in \mathcal{T}, l \in \mathcal{C} \tag{20}$$

Inequalities (20) are based on the fact that whenever a bus b_i uses a charger, the total amount of energy gained is bounded above by its useable battery capacity (the full battery size ϵ_i, adjusted for the minimum and maximum SOC values $\eta_i^{\max}$ and $\eta_i^{\min}$), coupled with constraint (14) that establish the relationship between t_i^l and x_i^l.

Next, for any trip i, let $\mathcal{C}(i)$ denote the set of chargers that could be used by b_i after i, i.e., $\mathcal{C}(i) = \{l \in \mathcal{C} : \overline{t_i^l} > 0\}$. Note that $\mathcal{C}(i)$ may be empty if trip i does not end close to any chargers. Substituting inequalities (20) into constraints (12) and restricting the summation to feasible charging opportunities only, we obtain valid inequalities (21):

$$\sum_{j\in\theta(i)}\sum_{l\in\mathcal{C}(i)} \epsilon_i\left(\eta_i^{\max} - \eta_i^{\min}\right) x_j^l \geq \sum_{j\in\theta(i)} \delta_j + \delta_i + \epsilon_i\left(\eta_i^{\min} - \eta_i^0\right) \qquad \forall i \in \mathcal{T} \tag{21}$$

Dividing the inequalities (21) by the useable battery capacity and adding the ceiling operator to the right-hand side gives the final form of the master problem cuts:

$$\sum_{j\in\theta(i)}\sum_{l\in\mathcal{C}(i)} x_j^l \geq \left\lceil \frac{\sum_{j\in\theta(i)} \delta_j + \delta_i + \epsilon_i\left(\eta_i^{\min} - \eta_i^0\right)}{\epsilon_i\left(\eta_i^{\max} - \eta_i^{\min}\right)} \right\rceil \qquad \forall i \in \mathcal{T} \tag{22}$$

Here, the ceiling of the right-hand side can be taken because the left-hand side of (22) is a summation of binary variables and therefore always integer. Inequalities (22) yields one MP cut for every trip in the problem instance. In practice, we do not need to add all of these to the MP. Rather, we create these cuts by proceeding through each block in order of trip sequence and calculating the value of the right-hand side. Each additional trip adds variables $\sum_{l\in\mathcal{C}(i)} x_i^l$ to the left-hand side and may or may not increase the right-hand side, since the total energy consumption is nondecreasing as we progress through a block. We therefore add a new cut to the MP each time the right-hand side increases to a new integer value; if it does not increase, then the cut for trip i is dominated by the cut for i^- and does not improve the MP formulation.

4.3. Subproblem

Solving the CB master problem provides a candidate solution in terms of the x and y variable values. The subproblem's role is then to check whether it could be an optimal solution to the complete problem by verifying both that it is a feasible choice of integer variable values and that it can minimize delay. This optimality criterion is evaluated by tracking an incumbent solution and its total delay value, denoted z^*, as described in Codato and Fischetti (2006).

For a given MP iteration k, let $\mathcal{A}_1^{(k)}$ be the set of arcs used in the solution, i.e., $(l,i,j) \in \mathcal{A}_1^{(k)}$ if $y_{ij}^{l(k)} = 1$, where the variable superscript (k) denotes the optimal value in the kth iteration. Likewise, let $\mathcal{T}_0^{(k)}$ be the set of trip-charger pairs for which charging is skipped, i.e., $(l,i) \in \mathcal{T}^{(k)}$ if $x_i^{l(k)} = 0$. Then the SP for iteration k is

$$\begin{aligned} \min_{d_i, p_i, t_i^l} \quad & \text{Total delay (1)} \\ \text{s.t.} \quad & \sum_{i\in\mathcal{T}} d_i \leq z^* - \phi \end{aligned} \tag{23}$$

$$p_j \geq p_i + t_i^l \qquad \forall (l,i,j) \in \mathcal{A}_1^{(k)} \tag{24}$$

$$t_i^l = 0 \qquad \forall (l,i) \in \mathcal{T}_0^{(k)} \tag{25}$$

Plugin time lower bound (2)
Delay tracking constraints (5) and (6)
State of charge management constraints (12)–(14)

The CB subproblem aims to minimize the delay based on the binary decisions from the MP. Constraint (23) requires that the subproblem solution must improve the current incumbent objective value z^* by some small tolerance ϕ, which ensures that when the MP is found to be infeasible, the current incumbent is optimal. It should be a small positive number that can be thought of as an optimality gap, since the subproblem checks whether the incumbent objective can be improved by at least ϕ. In our experiments, ϕ is set to 1 min, which should be an insignificant amount of total delay across the full transit network. A larger value of ϕ could be appropriate if a large amount of total delay is anticipated, which would reduce computational time. Constraints (24) and (25) replace the corresponding big M constraints from the full problem, enforcing bounds on plugin time p_i and charging time t_i^l only as needed based on the current MP solution. The remaining SP constraints are identical to those from the complete formulation in Section 3.7.

4.4. CB cuts

As shown in Fig. 6, when the SP is infeasible, we add CB cuts to exclude the incumbent MP solution. The key to creating these cuts is to identify a minimal infeasible subsystem (MIS), also called an irreducible infeasible subsystem (IIS) of the infeasible SP instance. An MIS is an inclusion-minimal set of rows of the SP constraint matrix, where “inclusion-minimal” means that if any single constraint is removed, the resulting subsystem of constraints admits a feasible solution. Codato and Fischetti (2006) described an algorithm to generate multiple such MISs for any MP solution that yields an infeasible SP.

An MIS of the subproblem identifies a set of MP variables—corresponding to the instances of the conditional constraints (24) and (25) included in the MIS—that have forced the problem to be infeasible. To make sure that the MP does not produce a solution with this same set of variables again, we add a CB cut to exclude it. For this purpose, we can represent the MIS as two sets of indices: one set $\mathcal{M}_x$ corresponding to constraints (24) and another $\mathcal{M}_y$ corresponding to constraints (25). That is, $\mathcal{M}_x = \left\{(l,i) \in \mathcal{M} : x_i^l = 0\right\}$ and $\mathcal{M}_y = \left\{\left(l,i,j\right) \in \mathcal{M} : y_{ij}^l = 1\right\}$. Given such an MIS, a CB cut is formed by the simple inequality (26):

$$\sum_{(l,i)\in\mathcal{M}_x} x_i^l + \sum_{(l,i,j)\in\mathcal{M}_y} (1 - y_{ij}^l) \geq 0 \tag{26}$$

Equation (26) enforce that at least one of the binary variables x_i^l and y_{ij}^l included in the MIS must change its value in order to obtain a feasible solution.

4.5. Implementation details

We implemented the CB algorithm using Python and the Gurobi solver via the `gurobipy` package. Following Fig. 6, we begin by initializing the master problem and adding CB cuts for heuristic solutions. In our experiments, we generate cuts for any heuristic solution with an objective value within 50% of the best identified objective. Once the MP is initialized, we run the CB algorithm within a branch-and-cut framework using Gurobi's callback capabilities. When a new optimal solution to the MP is detected, we first check if it contains any subtours. If it does, we cut off any such subtours by adding constraint (17) as lazy constraints. If it does not, we progress to solving the subproblem and generating CB cuts. To generate an MIS each time that the SP is infeasible, we use Gurobi's `compute IIS()` function, which identifies a single MIS out of many possibilities. Identifying multiple MISs (and consequently multiple cuts) can help the algorithm converge faster, so each time an MIS is found, we remove one constraint from that MIS at random, verify that the relaxed SP model is still infeasible, and run `computeIIS()` again to find a new unique MIS. We repeat this procedure until the relaxed SP model is feasible, which typically results in finding several MISs per CB iteration.

5. Heuristic solution method: 3S

Deriving the CB algorithm for our charge scheduling problem (P) inspired a heuristic algorithm that follows a similar pattern, but with much faster convergence. The heuristic design is based on a few key insights presented by the decomposition approach. First, most of the problem's complexity comes from the two interconnected decisions of selecting which trips include charging (the values of x_i^l) and the sequence in which these trips are served by each charger (the values of y_{ij}^l). Once these values are set, it is easy to determine the optimal charging durations and complete schedule of plugin/departure/delay times with the subproblem, which is just an LP. Additionally, the problem of selecting the order in which to serve trips may not be too difficult in practice once the charging trips have been selected. We can expect that the optimal charging order is unlikely to differ too much from simple first-in, first-out priority.

Based on this logic, we devised a heuristic algorithm based on relaxing and separating the MIP formulation (P). We relax the complicating queue constraint (4) and restructure the problem into three phases we call selection, sequencing, and scheduling, together forming what we call the 3S algorithm.

Fig. 7 gives an overview of the 3S heuristic algorithm. The first step is to initialize some random cost parameters α_i^l, which help the algorithm explore a wider range of feasible solutions. Then, in the selection phase, we solve a separate LP for each bus to select a set of trips when it will charge. This is equivalent to setting the values of x_i^l or populating the set $\mathcal{T}_0^{(k)}$ in the CB approach. In the sequencing phase, we perform a simple sorting operation for each charger to set the order in which it serves trips. The sequencing phase is analogous to setting the y_{ij}^l variable values or the set $\mathcal{A}_1^{(k)}$ from the CB master problem. In the final scheduling phase, we solve one more LP to optimize charging durations and the resulting delay given the selection and sequence decisions. This final LP is identical to the CB subproblem with the incumbent bound constraint (23) removed. The final output of any run of the 3S heuristic is a feasible solution to problem (P) and its corresponding objective value. Sections 5.1–5.3 next describe each of the three phases in detail.

5.1. Phase 1: Trip selection

The purpose of Phase 1 is to set feasible values of the binary charging decisions x_i^l for all trips i and chargers l. To do this, we relax constraint

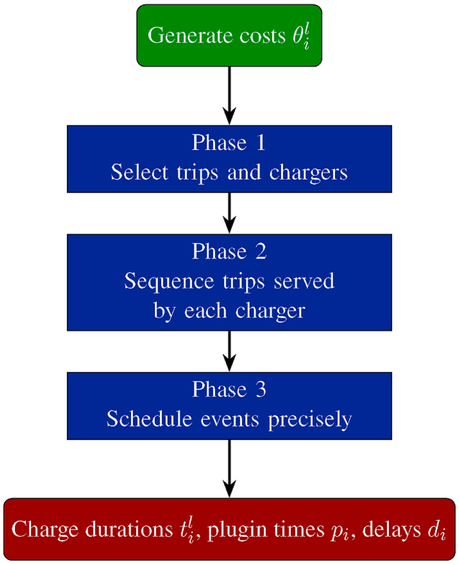

Generate costs θ_i^l
Phase 1
Select trips and chargers
Phase 2
Sequence trips served
by each charger
Phase 3
Schedule events precisely
Charge durations t_i^l, plugin times p_i, delays d_i

Fig. 7. Flowchart of 3S heuristic solution procedure.

(4), which is used to accurately calculate the plugin times and delays. Note, the relaxation does not influence feasibility with respect to state of charge. As a result, the plugin time variables p_i are controlled only by constraint (2), so that $p_i = d_i + \sigma_i + \tau_i$ for all trips i, and p_i can be removed from the formulation. The binary variables y^l_{ij} are also no longer needed when constraint (4) are removed from the MIP formulation (P), since they have no impact on the objective function.

One more simplification is needed to eliminate the binary variables x^l_i and make the Phase 1 problem a linear program, noted for brevity as Phase 1 LP in the sequel. Rather than including constraint (10) that rely on x^l_i and ensure each bus uses only one charger after each trip, we select the charger to be used at random from the set $\mathcal{C}(i)$ of accessible chargers for each trip i. Let $\widehat{l_i}$ be the randomly chosen charger for trip i. Then we ensure only one charger is used by defining a new upper bound on the charging time with Eq. (27):

$$\widehat{t^l_i} = \begin{cases} \overline{t^l_i} & \text{if} \quad l = \widehat{l_i} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

With the single-charger limit handled as such, the binary variables x^l_i can be excluded, yielding the Phase 1 LP (28)–(30):

$$\min_{d_i, t^l_i} \quad \sum_{i \in \mathcal{T}} \left[d_i + \sum_{l \in \mathcal{C}} \alpha^l_i t^l_i \right] \tag{28}$$

$$\text{s.t.} \quad d_i \geq \sigma_{i^-} + d_{i^-} + \tau_{i^-} + t_{i^-} - \sigma_i \qquad \forall i \in \mathcal{T} : n_i \geq 2 \tag{29}$$

$$\begin{aligned} &\text{Delay nonnegativity (6)} \\ &\text{State of charge constraints (12) and (13)} \\ &0 \leq t^l_i \leq \widehat{t^l_i} \qquad \forall i \in \mathcal{T}, l \in \mathcal{C} \end{aligned} \tag{30}$$

The Phase 1 LP, the objective function (28) includes a randomization term $\sum_{i \in \mathcal{T}, l \in \mathcal{C}} \alpha^l_i t^l_i$, where α^l_i is randomly sampled from a suitable distribution. This term helps to generate a variety of feasible solutions through repeated runs of the 3S algorithm. Since there will often be many different charging patterns that result in the same optimal delay in Phase 1 (especially if the minimum total delay is zero), adding this random coefficient to the charging times helps 3S to consider a greater variety of potentially good first-stage solutions. A distribution should be chosen so that α^l_i takes on a variety of both positive and negative values. Negative values of α^l_i are useful to produce results in which a bus distributes its charging over a greater number of trips, which may help to reduce queuing. It is also important that the values of α^l_i are not so large that the randomization term dominates the delay term in the objective function, which could result in poor performance. In our experiments, we sample the values of α^l_i from a normal distribution with mean 0 and standard deviation 0.5.

The Phase 1 LP is separable across all buses in the system, since the conditional plugin time constraint (4) that linked trips on different blocks have been removed. Phase 1 therefore consists of N_b independent linear programs, where N_b is the number of buses serving the trips $\mathcal{T}$. Note that since the number of variables only scales with the number of trips served by a single vehicle, each LP is quite small.

Solving the Phase 1 LP yields first-stage charging durations t^l_i and delays d_i for all trips i and chargers l. We convert the charging durations to their equivalent binary values x^l_i, i.e., $x^l_i = 1$ if $t^l_i > 0$ and $x^l_i = 0$ otherwise. These binary decisions also correspond to the set of non-charging trips $\mathcal{T}^{(k)}_0$ from the CB master problem. Each trip i for which $\sum_{l \in \mathcal{C}} x^l_i = 1$ is also assigned a first-stage plugin time $p_i = d_i + \sigma_i + \tau_i$ when it is expected to start charging. These form the inputs to Phase 2.

5.2. Phase 2: Charger sequencing

In Phase 2, we process the charging decisions from Phase 1 into a sequence for every charger that is used. For each charger l, we simply sort all trips i for which $x^l_i = 1$ by their first-stage plugin times p_i. Sorting these trips gives us a simple mapping to a feasible set of charging arcs $\mathcal{A}^{(k)}_1$: For any consecutive pair of trips i and j, we add arc (l, i, j) to $\mathcal{A}^{(k)}_1$. For each charger we also add arcs connecting to the initial and final dummy nodes, respectively. That is, if i is the first and j is the last trip served by charger l, we add the arcs (l, s, i) and (l, j, t).

5.3. Phase 3: Event scheduling

Phase 3 takes the results obtained from Phases 1 and 2—essentially, all the binary decisions of the complete MIP formulation—and uses them to define an exact charging schedule to minimize delays. Since the first two phases populated the sets $\mathcal{T}^{(k)}_0$ and $\mathcal{A}^{(k)}_1$, we have all the inputs necessary to construct an instance of the subproblem, though we need to either set the incumbent delay value z^* to an arbitrarily high value or remove constraint (23) entirely. Since the SP is a linear program whose number of variables and constraints scales roughly with the number of trips in an instance, it can generally be solved quite quickly. Solving the SP yields final values of charging duration and delay for each trip that account for queuing. Note that in Phase 3 we allow for changing the charging durations originally output by Phase 1; The purpose of Phase 1 is to make the binary decisions of when and where to charge, but we relax the problem to an LP for efficiency. Also note that including the charging time constraint (30) in the Phase 1 LP for each bus ensures that Phase 3 is always feasible; the purpose of Phase 3 is to optimize charging durations to exactly determine the minimum possible delay for the given set of charging trips and sequence for each charger.

6. Case studies

We constructed a variety of instances in order to analyze the performance of the CB method and 3S heuristic. These instances were based on two different transportation networks. The first, described in Section 6.1, is a small notional network consisting of two bus routes served by a single shared charger. The small network allows us to evaluate the performance (in terms of solution time and optimality gap) of both our methods in comparison to directly solving the model with Gurobi. The second network, described in Section 6.2, is based on the actual transit system operated by King County Metro in the greater Seattle area. The instances on this larger network are too difficult to be solved to optimality by either the CB method or an off-the-shelf solver, but show how the 3S heuristic can support transit operations at a real-world scale.

6.1. Simple notional network

To test the CB method and assess the performance of the 3S heuristic, we used a small test network originally presented in Appendix B of McCabe and Ban (2023). That original network was further simplified to include only two of the routes ($\mathbb{A}$ and $\mathbb{C}$) and a single charger at their shared terminal, as sketched in Fig. 8. Route $\mathbb{A}$'s headway was decreased from 20 to 30 min to limit the number of vehicles in the case study, but all other parameter values remained the same as reported in McCabe and Ban (2023). All buses have 500 kWh of battery capacity (ϵ_i), lower and upper state-of-charge bounds $\eta^{\min}_i = 0.1$ and $\eta^{\max}_i = 0.9$, and enter service with batteries at the maximum state of charge (i.e., $\eta^0_i = 0.9$). The energy consumption δ_i for each trip is calculated based on a rate of 3 kWh/mi (1.86 kWh/km) and we test a variety of charger power levels ρ^l from 300 to 500 kW. The CB algorithm improvement tolerance ϕ is set to 1 min and the 3S objective function parameters α^l_i are sampled randomly in each iteration from a normal distribution with mean 0 and standard deviation 0.5.

Table 2 documents the schedule and distance parameters of each route. With such a schedule, the network consists of 8 buses that complete 84 total trips. All 8 buses must use opportunity charging in order to

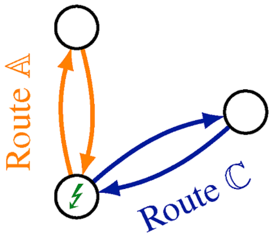


Fig. 8. Simple case study network.

maintain sufficiently charged batteries throughout the day.

6.2. King county metro network

The second test network is based on the transit system of South King County, WA, USA. We use this network to study the performance of the 3S heuristic, as it is too large to be solved to optimality by either Gurobi or our CB method. The case study includes some of the busiest routes planned for electrification in the near future: the RapidRide lines F and H as well as routes 131, 132, 150, 153, 161, and 165. We collected relevant data including trip schedules (corresponding to σ_i and τ_i values) and distances as well as block sequences from the Metro GTFS feed (King County Metro, 2024b) that was posted in March 2024. We used Wednesday, April 3, 2024 as a test date.

6.2.1. Data collection

To filter down the GTFS data to the scope of our case study, we first identified all `service_id` values active on the case study date and all blocks and trips active for these service IDs. We then filtered down these blocks to only include those that serve the specified routes exclusively. We assumed each bus had a battery capacity ϵ of 525 kWh, consistent with the current King County Metro fleet (King County Metro, 2024a), the SOC of each bus must be kept between $\eta_i^{\min} = 15\%$ and $\eta_i^{\max} = 90\%$, and each bus enters service with $\eta_i^0 = 0.9$. The energy consumption of each trip δ_i was set based on its distance as calculated from GTFS, multiplied by an average energy consumption rate of 3.19 kWh/mi (1.98 kWh/km), which was the reported average for Metro's 60-foot BEBs in March 2024 in McCabe (2024). The first and last trips of each block also had increased energy demand due to pull-out and pull-in trips from/to the depot. We assumed all buses were kept overnight at Metro's South Base and calculated pull-out/pull-in distances based on driving directions between the base and all relevant terminal locations, as calculated by the Openrouteservice API (Openrouteservice, 2024). To set the upper bound on charging time $\overline{t}_i^l$ for all trips and chargers, we identified the final stop coordinates of each trip and calculated the driving distance to each charger (again using Openrouteservice) and assumed buses were able to use a charger if the distance to it was less than 0.25 mi (0.4 km). As in the simple case study, the CB algorithm improvement tolerance ϕ is set to 1 min, and the 3S objective function parameters α_i^l are sampled randomly in each iteration from a normal distribution with mean 0 and standard deviation 0.5.

The case study includes all blocks that serve trips on the RapidRide F Line and H Line as well as routes 131, 132, 150, 153, 161, and 165. Buses on these routes are served by two chargers at the Burien Transit Center (which serves the F Line, H Line, 131, 132, 161, and 165) as well as one charger each at Renton Landing (the eastern terminal of the F Line) and Kent Station (the southern terminal of routes 150, 153, and 161). We assumed all chargers had an identical power output of 220 kW in each test case, based on the maximum accepted power of King County Metro's BEBs (King County Metro, 2024a). Fig. 9 maps all trips included in this scope as well as the three charger locations considered.

In total, the King County network we analyze includes 34 buses that need to use opportunity chargers at some point during the day. A further 73 blocks that serve these routes are excluded because they can be completed with depot charging alone. The 34 fast-charging buses complete a total of 342 trips on the test date.

Table 2
Basic parameters for notional routes used in simple case study.

Route	𝔸	ℂ
Distance (m)	15	15
Time (min)	40	45
Layover time (min)	20	15
Headway (min)	30	60
One-way distance	15 mi (24 km)	15 mi (24 km)

6.2.2. Case study scenarios

The King County case study includes two scenarios to highlight the flexibility of our modeling approach and the 3S algorithm. In Scenario A, a charging schedule is designed for the baseline instance described above, with two chargers located at the Burien Transit Center and one each at the other two sites. In Scenario B, we suppose that one of the chargers at the Burien Transit Center is broken and out of service for the day, making charging operations much more constrained. In such a scenario, King County Metro would need to determine whether this reduced capacity was sufficient to meet all opportunity charging needs for the day and, if so, what level of performance could be expected. The results in Section 6.3 show that in the scenario considered, a moderate amount of delay is still present when only one charger is available in Burien, but our model can effectively plan charging to limit delays so that service is still acceptable.

6.3. Results

6.3.1. Simple network

We solved the model for the simple network of Section 6.1 at a variety of different charger power levels from 300 to 500 kW. The two exact methods were subject to a time limit of 1 h for each instance; the 3S heuristic was run 500 times for each case, though it was terminated as soon as a solution with zero delay was found, if applicable. The results are summarized in Table 3. For each method, we report the objective value of the best solution obtained, the time to find the first solution with that best objective value, and the total solution time. A dash is used to indicate that the algorithm timed out.

In Table 3, we see that optimal delay decreases and eventually reaches zero as the power level increases. For these small instances involving only 8 buses, Gurobi solved the problem to optimality in 4 out of 5 cases. The comparative performance of the heuristic and CB methods varied depending on the instance. At the two lower power levels which represent more difficult instances, the Benders algorithm failed to prove optimality within the time limit, though in the 350 kW case it did successfully identify an optimal solution with 71.4 min of delay. With 400 kW chargers, the CB method outperforms the direct solution via Gurobi. As intended, the 3S algorithm quickly identifies an optimal solution with 5 min of total delay and the CB procedure proves its optimality in 47 s, whereas directly solving the problem took 66 s. For the 450 and 500 kW cases, the direct solution approach struggles to find a good solution, whereas 3S identifies a zero-delay solution very quickly and the entire CB procedure can be skipped because total delay can never be negative.

To illustrate the results in more detail, Fig. 10 shows the complete timeline of activities for all buses in the 400 kW instance. Gray blocks on the timeline indicate that a bus is completing a passenger service trip that departed on time; delayed trips are shown in orange. A blue block in a bus's timeline indicates that it is plugged in at the charger. Looking at block CF1 for example, which completes 7 trips on Route ℂ, we can see that it charges after trips 1, 3, and 5, and all trips are on time. Blocks AR1 and AR2, which both complete more trips and require more charging, each have one trip that departs a few minutes late. We can see that this

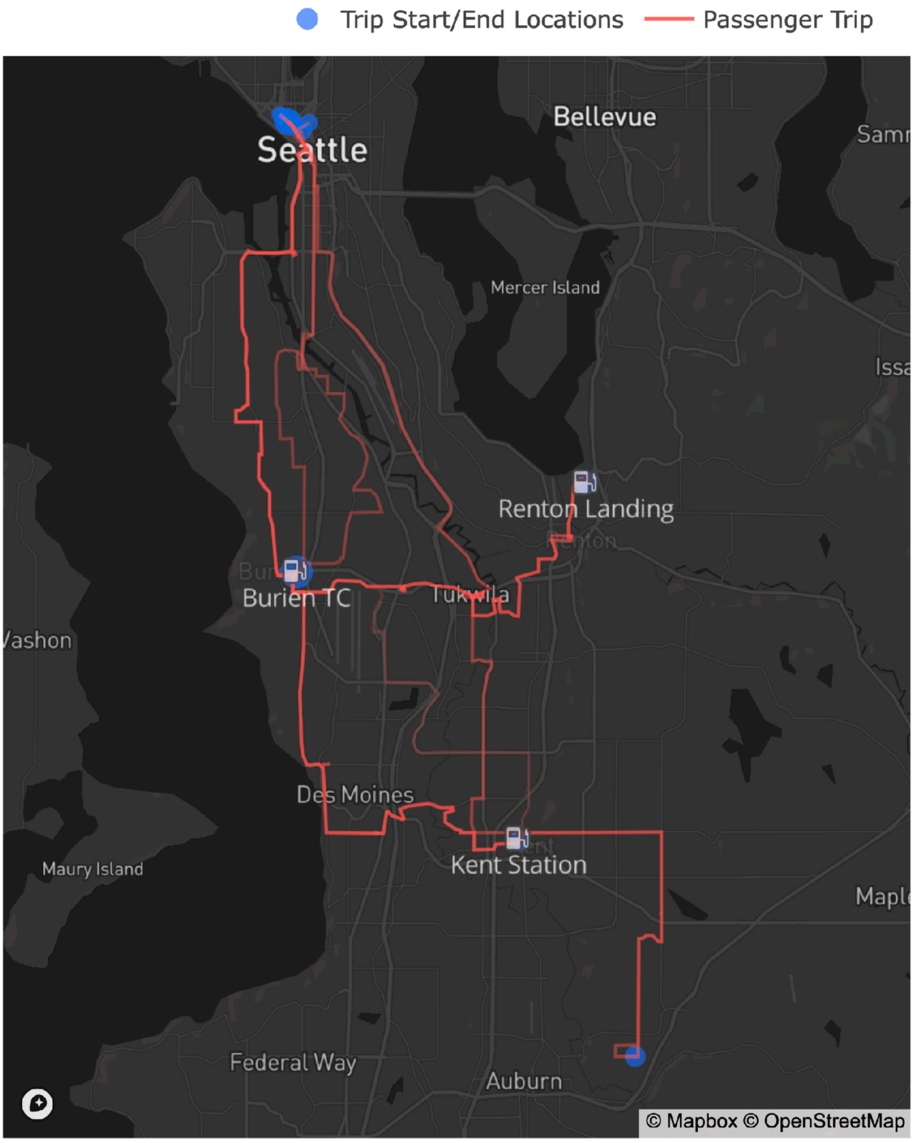


Fig. 9. Map showing trips, terminals, and charger locations for the King County case.

Table 3
Summary of results on simple case study instances. BO: best objective value (min of total delay). T-BO: time to find solution with best objective (s), T-T: total solution time (s). A dash indicates that the algorithm did not terminate.

	Direct solve			3S Heuristic			Benders		
Power	BO	T-BO	T-T	BO	T-BO	T-T	BO	T-BO	T-T
300 kW	704.0	133.0	–	759.0	0.70	12.0	759	N/A	–
350 kW	71.4	15	57	73.7	0.05	9.0	71.4	1463.0	–
400 kW	5.0	56	66	5.0	0.70	10.0	5	0.7	47
450 kW	0.0	55	55	0.0	0.10	0.1	N/A	N/A	N/A
500 kW	0.0	138	138	0.0	0.40	0.4	N/A	N/A	N/A

optimal solution involves frequent short charges—in fact, all eight buses plug in whenever they are able to (every other trip).

These results highlight both the benefits and limitations of our methods. First, we note that the 3S heuristic identifies a near-optimal (often optimal) solution very quickly in all cases. Its worst performance was for the 300 kW case, when the best solution it identified had only 7.8% more delay than the best one found by Gurobi. One key advantage of the 3S method is that its solution time does not depend significantly on the difficulty of the instance being solved; while neither of the exact methods converged within the time limit for the 300 kW instance, the heuristic produced good feasible solutions just as quickly as it did for other instances.

CB produced mixed results on our instances. We see that for the 400 kW instance, the CB method converges more quickly than solving directly, but it is much slower than Gurobi for the two harder instances. In the 350 kW case, it is notable that CB is able to improve the incumbent solution supplied by the heuristic, but it still does not terminate before the time limit. With a 450 or 500 kW charger, integrating Benders with the 3S approach means that we can avoid running the Benders process entirely, since 0 is the best possible feasible delay.

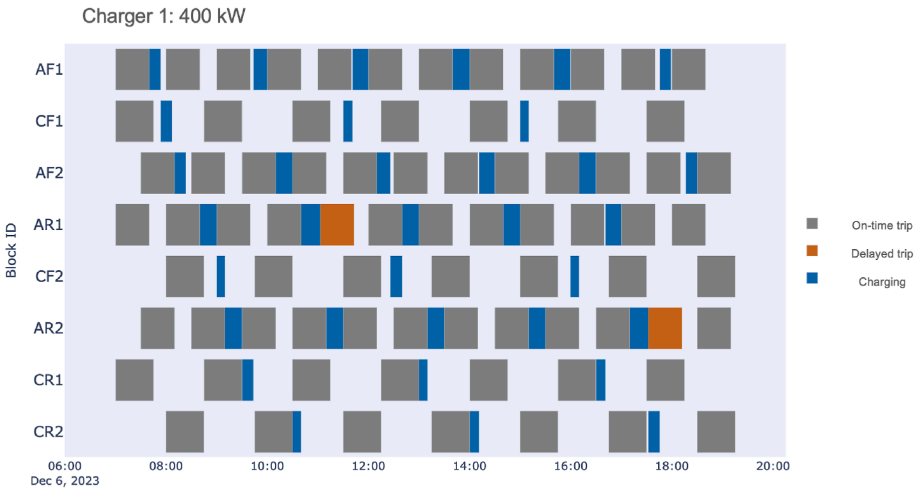


Fig. 10. Optimized timeline of operations for the simple network with 400 kW chargers, showing all charging activities, passenger trips, and on-time status.

There are a few likely reasons for the limited benefits of the CB approach for our charge scheduling model. First, delegating all scheduling and delay calculation to the subproblem limits the information that is available to the master problem, which limits Gurobi's ability to speed up the solution process via presolve methods and cuts. Based on our results, it seems that the cuts generated by the CB approach are not particularly strong compared to the cuts Gurobi generates on its own when provided with the complete formulation. Second, the CB method is primarily used for problems in which the objective depends only on the integer variables. While we followed the recommendations from Codato and Fischetti (2006) on adapting the method to a problem where the objective depends instead on continuous variables, we found that the approach is not as well suited to this type of problem. For one, convergence is inevitably slower compared to the typical CB application because it usually requires a large number of CB cuts to determine that the MP is infeasible and terminate the algorithm. Additionally, because the MP objective has no real meaning for the complete problem, we do not get any information on the optimality gap. Whereas in the typical CB application the Benders cuts are embedded within a branch-and-cut procedure and the search process continually improves upper and lower bounds on the optimal objective value, in our case we have an upper bound from the incumbent solution, but do not obtain any lower bound until the MP is infeasible and optimality is proven.

6.3.2. *King County Metro network*

Table 4 documents the key results from applying the 3S heuristic to the King County network for the two scenarios. In Scenario A, the total delay is only about 6.5 min, split across two trips. Evidently, there are two blocks where the scheduled layover time is not sufficient to accommodate all daily charging needs with a power output of 220 kW. Although no zero-delay solution can be found, the 3S algorithm identifies its best solution to the problem in about half a second, while running 500 iterations with different randomized objective coefficients takes just over 80 s.

In Scenario B, when only one charger at the Burien Transit Center is available, the total delay is unsurprisingly greater, totaling just under 4 h spread across 18 of the 342 total trips. It takes just under 1 min to run all 500 iterations of the 3S algorithm for this instance and the best solution is identified close to the end of that time. However, we note that a comparable solution with objective value 234 was identified after 2.4 s, so good solutions were still generated quickly. That said, running the algorithm for longer with various values of the randomization parameters α_i^l did yield some improvements.

Further analysis reveals the impact that reducing the number of available chargers in Burien has on the solution. Table 5 records the total amount of time all buses spend connected to each charger in the two scenarios. In Scenario A, the two identical chargers located at the Burien Transit Center are used about equally, both more heavily than the chargers at the other two locations. When only one of these Burien chargers is available in Scenario B, it serves almost all of the charging demand originally met by the two chargers in Scenario A; however, about 100 min of charging time is shifted away from Burien to Renton, which is less heavily utilized. Some buses visit both of these trip terminals, and in Scenario B it becomes beneficial for them to charge in Renton rather than Burien as much as possible. Note that the total amount of time spent charging—which is dictated by the energy consumption, battery sizes, and charger power level—is exactly the same in both scenarios.

Fig. 11 plots a timeline of charger utilization for Scenario B in the same style as Fig. 10, showing only the charger at the Burien Transit Center because it is the most heavily used and the source of most delays. As before, gray blocks on the timeline represent on-time service trips, orange blocks represent delayed trips, and blue blocks indicate charging in Burien. Because only one charger is available, it appears that some delays are inevitable on this day, and our model attempts to fit in necessary charging activities in a way that minimizes the resulting delays. Fig. 11 shows that 3S effectively exploits layover time in the schedule to limit delay propagation. Block number 7157064 (just below the middle of Fig. 11), which charges four times in Burien, provides a useful example. Though two of its trips are delayed due to charging, both of these trips have a comparatively large amount of layover time prior to the next trip, allowing the bus to catch back up to schedule for the trips that follow. In the two other cases, the bus serving this block charges for a short enough duration between trips that no delays are incurred. Fig. 11 also shows that our approach accurately propagates delays across trips—for example, the final trip on block 7157058 is delayed about 6 min because the scheduled layover time prior to it is not sufficient to fully eliminate the 38-min delay of its penultimate trip.

One more notable trend in Fig. 11 is that delays are concentrated towards the end of the service day. It is unsurprising that this is an effective way to limit total delays—if there are not many trips remaining after one that is delayed, then there are not many opportunities for

Table 4
Summary of results on King County Metro network. BO: best objective value (min of total delay). T-BO: time to find solution with best objective (s). T-T: total solution time (s). ND: number of delayed trips.

Scenario	BO	T-BO	T-T	ND
A	6.6	0.5	81.7	2
B	230.1	54.2	59.4	18

Table 5
Total amount of time (min) each charger was used across the two real-world case study scenarios.

Scenario	Burien 1	Burien 2	Renton	Kent
A	536	668	174	92
B	1104	0	274	92

delays to propagate forward. However, it should be noted that this approach results in a lower quality of service late at night compared to the rest of the day, which might not be desired by transit agencies. If it were a significant concern, this effect could be mitigated by adjusting the objective function, for example, by adding a larger coefficient to the delay of these trips later in the day. Of course, doing so could have the unintended consequence of significantly increasing total delay in favor of more consistent service.

Finally, note that there are several charger visits seen in Fig. 11 with short durations (i.e., only a few minutes in length). These events are possible because the formulation (P) does not include a lower bound on charging behavior. If a minimum charging time must be enforced, constraints (14) can be updated to include a lower bound, e.g., $5x_i^l \leq t_i^l \leq \overline{t}_i^l x_i^l \quad \forall l \in \mathcal{C}, i \in \mathcal{T}$ to enforce a minimum charge duration of 5 min. Incorporating this new lower bound into the 3S algorithm introduces some new complications, but simply adding the lower bound constraint to the Phase 3 LP should work in most cases, despite a slight risk of feasibility issues (e.g., when charging for 5 min would cause the battery capacity to be exceeded).

7. Conclusions

This study developed a novel model for scheduling within-day electric bus fleet recharging as well as exact and heuristic methods for its solution. Our method takes a unique approach to opportunity charging scheduling for BEBs: we precisely track queuing at chargers to set the start and end times of each bus's daily activities, including passenger trips and charger visits. Rather than constraining charging to always occur during scheduled layover time, we carefully track the departure delay of each trip and propagate these delays across trips completed by the same vehicle. The resulting model is a mixed-integer linear program that seeks to minimize total delay for the full BEB network.

Because this flexible scheduling model allows but minimizes delays, it still produces actionable results when operating conditions such as traffic delays, high energy consumption, or charger malfunctions make it impossible to meet charging needs without delaying any trips. Our approach stands out compared to prior methods from the literature, which typically constrain charging to take place during scheduled layover time and are not naturally extendable to scenarios where insufficient time is available. For the same reason, our approach should also be a good fit for stochastic extensions that recognize real-world variations in travel time—rather than labeling a problem instance infeasible because it results in delays, a solution to our model calculates how good or bad the transit network's performance is in terms of the complete system delay.

Our exact solution method based on CB decomposition outperforms a state-of-the-art commercial solver on some instances. However, our results show that this approach is still not efficient enough to be suitable for large real-world transit networks under demanding conditions. Although this method can provide a proof of optimality for some instances that provides theoretical value, it has considerable limitations. In particular, our usage of CB for a problem in which the objective value is set solely by the subproblem is not ideal. Further improvements to the CB approach may be possible by, for example, using a more tailored procedure to find a minimal infeasible subsystem as described in Codato and Fischetti (2006). However, given the algorithm's mixed performance and the inherent scalability limits of any exact method for a mixed-integer program, the benefits may still be limited.

However, our 3S heuristic algorithm that was inspired by the CB approach shows excellent results on our test instances. On our smaller test network instances in Section 6.1, the 3S method found good feasible solutions orders of magnitude faster than exact methods and its optimality gap was always below 8%. The two test scenarios on the King County Metro network from Section 6.2 demonstrated how the heuristic could quickly find a solution with nearly zero delay under favorable conditions, as well as its ability to respond to unfavorable conditions and produce a solution that keeps most trips running on time just as quickly.

The scheduling model and solution algorithms carry some notable limitations, primarily due to some of the key assumptions. For instance, we assume that chargers can only be used if they are located close to the terminals of trips, so that an insignificant amount of deadheading (in terms of both time and energy consumption) is required to access them. Consequently, our model is not a direct fit for agencies that plan to locate opportunity chargers away from trip terminals. However, if opportunity chargers are located close enough to trip terminals that the time and energy required to access them is within a reasonable margin of error, our approach can still be useful. Additionally, it would be straightforward to extend the energy consumption constraints (12) and (13) and plugin time constraints (2) to include deadhead energy and time concerns by multiplying the charging decision variables x_i^l by appropriate parameters. Assumption 6, which states that battery charging is linear with respect to time, also has significant implications for our model's applicability. If buses recharge close to their full state of charge during the day, when the charging power must be reduced to protect the battery, then this linear approximation may not be sufficiently accurate. For relatively short charging durations with a moderate state of charge, as can be expected for opportunity charging, assuming a linear process should be sufficiently accurate (Montoya et al., 2017). Our numerical results also show that short charging durations are typical in real-world operations (see Fig. 11), further justifying this assumption.

Although our model is intended to be realistic, agencies would still likely face some challenges in implementing our proposed opportunity charging schedules. Communicating detailed daily charging schedules to bus operators could be a challenge, especially if opportunity charging plans were revised at all during the service day. Labor regulations would also likely play a role—for example, drivers might not be considered to be on a break when responsible for charging buses, which could impact the permissibility of their work schedules. Similar challenges are likely to arise with any opportunity charging scheme.

This work also suggests various directions for further research. These future research ideas are centered on improvements to and extensions of the 3S algorithm. First, further improvements to 3S might be possible—for example, by incorporating more feedback between the three phases, taking advantage of opportunities for parallelism, and investigating theoretical bounds on the optimality gap. Additionally, given the 3S algorithm's already strong performance in finding good feasible solutions, future work could explore a dynamic application in which charging schedules are continually re-optimized throughout the day based on updated observations and predictions of energy consumption and on-time performance. With this approach, buses' planned charging amounts could be increased or decreased in accordance with their cumulative energy consumption and the timing of charger visits could be better aligned with projected schedule adherence. Another promising research direction, as mentioned earlier in this section, is to develop a stochastic version of our charging scheduling model. Such a model could, for example, minimize the expected delays across a wide range of realistic scenarios, rather than treating energy consumption and travel times as deterministic. The 3S method's ability to generate good solutions quickly could help enable this extension.

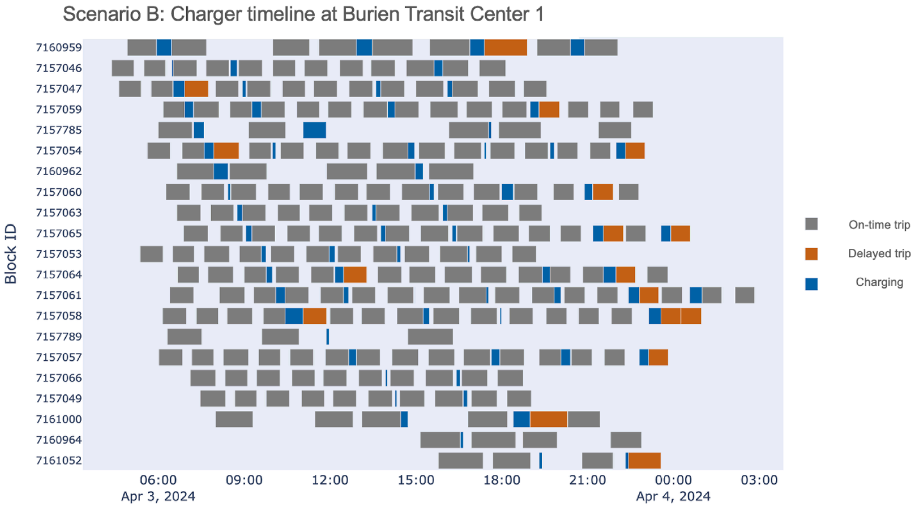


Fig. 11. Timeline showing charging activities and passenger trips (both on-time and delayed) at the Burien Transit Center in Scenario B.

Replication and data sharing

The complete source code and data required to run our analysis and replicate the two case studies is publicly available as a GitHub repository at https://github.com/dan-mccabe/min-delay-charging.

CRediT authorship contribution statement

Dan McCabe: Writing – original draft, Validation, Software, Methodology. **Xuegang (Jeff) Ban:** Validation, Funding acquisition, Conceptualization, Writing – review & editing. **Balaźs Kulcsár:** Writing – review & editing, Methodology, Conceptualization, Resources.

Declaration of competing interest

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

Acknowledgements

The work of the first author was supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under (No. DGE-1762114). This research was also partially supported by a grant from the Pacific Northwest Transportation Consortium (PacTrans) funded by US Department of Transportation (USDOT). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF, USDOT, or PacTrans. This work was in part supported by the Transport Area of Advance within Chalmers University of Technology and by the European Commission (EC) (No. F-DUT-2022-0078) co-funded project ERGODIC: Combined passenger and goods transportation in suburb traffic.

References

Abdelwahed, A., van den Berg, P.L., Brandt, T., Collins, J., Ketter, W., 2020. Evaluating and optimizing opportunity fast-charging schedules in transit battery electric bus networks. Transp. Sci. 54, 1601–1615.

Amalgamated Transit Union, 2022. Agreement between amalgamated transit union, local 587 and king county metro transit. Local 587. https://cdn.kingcounty.gov/-/media/king-county/depts/executive/documents/labor-relations/documents/agreements/410C0123.ashx.

Bao, Z., Li, J., Bai, X., Xie, C., Chen, Z., Xu, M., et al., 2023. An optimal charging scheduling model and algorithm for electric buses. Appl. Energy 332, 120512.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252.

Bie, Y., Ji, J., Wang, X., Qu, X., 2021. Optimization of electric bus scheduling considering stochastic volatilities in trip travel time and energy consumption. Comput. Aided Civ. Infrastruct. Eng. 36, 1530–1548.

Brinkel, N., Zijlstra, M., van Bezu, R., van Twuijver, T., Lampropoulos, I., van Sark, W., 2023. A comparative analysis of charging strategies for battery electric buses in wholesale electricity and ancillary services markets. Transp. Res. Part E Logist Transp Rev 172, 103085.

Ceder, A., 2007. Public Transit Planning and Operation. CRC Press, Boca Raton, USA.

Codato, G., Fischetti, M., 2006. Combinatorial Benders' cuts for mixed-integer linear programming. Oper. Res. 54, 756–766.

Desaulniers, G., Hickman, M.D., 2007. Public transit. In: Barnhart, C., Laporte, G. (Eds.), Handbooks in Operations Research and Management Science: Transportation. Elsevier, pp. 69–127.

Esmaeilnejad, S., Kattan, L., Wirasinghe, S.C., 2023. Optimal charging station locations and durations for a transit route with battery-electric buses: a two-stage stochastic programming approach with consideration of weather conditions. Transport. Res. C Emerg. Technol. 156, 104327.

Federal Transit Administration, 2022. Bipartisan infrastructure law. https://www.transit.dot.gov/BIL.

Gairola, P., Nezamuddin, N., 2023. Optimization framework for integrated battery electric bus planning and charging scheduling. Transp Res D Transp Environ 118, 103697.

He, J., Yan, N., Zhang, J., Wang, T., Chen, Y.-Y., Tang, T.-Q., 2023. Battery electricity bus charging schedule considering bus journey's energy consumption estimation. Transp Res D Transp Environ 115, 103587.

He, Y., Liu, Z., Song, Z., 2020. Optimal charging scheduling and management for a fast-charging battery electric bus system. Transp. Res. Part E Logist Transp Rev 142, 102056.

Hooker, J.N., 2007. Planning and scheduling by logic-based Benders decomposition. Oper. Res. 55, 588–602.

Hooker, J.N., Ottosson, G., 2003. Logic-based Benders decomposition. Math. Program. 96, 33–60.

IEA, 2023. Global EV outlook 2023. https://www.iea.org/reports/global-ev-outlook-2023/trends-in-electric-heavy-duty-vehicles.

King County Metro, 2024a. King county metro electric bus summary. https://assets-global.website-files.com/65dd1596eb8f2dc1fec8c36d/65fa1434f284724e556fdc99_King County Metro BEBs Summary.pdf.

King County Metro, 2024b. King county metro GTFS feed. https://metro.kingcounty.gov/GTFS/.

King County Metro Transit, 2022. Moving to a zero-emission bus fleet: transition plan. King County Metro Transit. https://kingcounty.gov//media/depts/metro/accountability/reports/2022/zero-emission-bus-fleet-transition-plan-may-2022.

Lacombe, R., Murgovski, N., Gros, S., Kulcsár, B., 2024. Integrated charging scheduling and operational control for an electric bus network. Transp. Res. Part E Logist Transp Rev 186, 103549.

Liu, K., Gao, H., Wang, Y., Feng, T., Li, C., 2022. Robust charging strategies for electric bus fleets under energy consumption uncertainty. Transp Res D Transp Environ 104, 103215.

Los Angeles County Metropolitan Transportation Authority, 2021. Final rollout plan. Los Angeles county metropolitan transportation authority. https://ww2.arb.ca.gov/sites/default/files/2021-09/LAMetroRolloutPlanADA.pdf.

Manzolli, J.A., Trovao, J.P.F., Antunes, C.H., 2022. Electric bus coordinated charging strategy considering V2G and battery degradation. Energy 254, 124252.

Maryland Transit Administration, 2024. MTA performance improvement. https://www.mta.maryland.gov/performance-improvement.

Massachusetts Bay Transportation Authority, 2021. MBTA approach to overcoming winter range challenges with battery electric buses. Massachusetts Bay Transport. Author. https://cdn.mbta.com/sites/default/files/2021-09/2021-09-30-overcoming-winter-range-with-bebs-accessible.pdf.

McCabe, D., 2024. Computational Tools for Battery-Electric Bus Systems: from Infrastructure Planning to Daily Operations. University of Washington, United States – Washington. Ph.D. Dissertation.

McCabe, D., Ban, X.J., 2023. Optimal locations and sizes of layover charging stations for electric buses. Transport. Res. C Emerg. Technol. 152, 104157.

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transp. Res. Part B Methodol. 103, 87–110.

Openrouteservice, 2024. Openrouteservice. https://openrouteservice.org/.

Peng, B., Keskin, M.F., Kulcsár, B., Wymeersch, H., 2021. Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning. Commun. Transp. Res. 1, 100017.

Perumal, S.S., Lusby, R.M., Larsen, J., 2022. Electric bus planning & scheduling: a review of related problems and methodologies. Eur. J. Oper. Res. 301, 395–413.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The Benders decomposition algorithm: a literature review. Eur. J. Oper. Res. 259, 801–817.

Stock, K., 2023. In the race to electrification, the humble bus is in the lead. https://www.bloomberg.com/news/articles/2023-06-09/buses-are-going-electric-faster-than-passenger-cars.

Tang, X., Lin, X., He, F., 2019. Robust scheduling strategies of electric buses under stochastic traffic conditions. Transport. Res. C Emerg. Technol. 105, 163–182.

Tingstad Jacobsen, S.E., Lindman, A., Kulcsár, B., 2023. A predictive chance constraint rebalancing approach to mobility-on-demand services. Commun. Transp. Res. 3, 100097.

Washington Metropolitan Area Transit Authority, 2024. Metro performance scorecard. https://www.wmata.com/about/records/scorecard/index.cfm.

Wu, J., Kulcsár, B., Selpi, Qu X., 2021. A modular, adaptive, and autonomous transit system (MAATS): an in-motion transfer strategy and performance evaluation in urban grid transit networks. Transp. Res. Part A Policy Pract 151, 81–98.

Wu, W., Lin, Y., Liu, R., Jin, W., 2022. The multi-depot electric vehicle scheduling problem with power grid characteristics. Transp. Res. Part B Methodol. 155, 322–347.

Zeng, Z., Wang, S., Qu, X., 2022. On the role of battery degradation in en-route charge scheduling for an electric bus system. Transp. Res. Part E Logist Transp Rev 161, 102727.

Zhou, F., Lischka, A., Kulcsár, B., Wu, J., Haghir Chehreghani, M., Laporte, G., 2025. Learning for routing: a guided review of recent developments and future directions. Transport. Res. E Logist. Transport. Rev. 202, 104278.

Dan McCabe received M.S and Ph.D. degrees from the University of Washington in Seattle, Washington, USA, in 2021 and 2024, respectively. In 2023, he was a visiting Ph.D student at Chalmers University of Technology in Gothenburg, Sweden. His research interests include optimization, simulation, and machine learning methods for public transit and freight transportation. He is currently a Postdoctoral Researcher at the National Renewable Energy Laboratory.

Xuegang (Jeff) Ban is the William and Marilyn Conner Endowed Professor with the Department of Civil and Environmental Engineering of the University of Washington. He received the B.S. and M.S. degrees in automotive engineering from Tsinghua University, and the M.S. degree in computer sciences and Ph.D. degree in civil engineering (transportation) from the University of Wisconsin at Madison. His research interests are in transportation network system modeling and simulation, and urban traffic modeling and control. His recent work focuses on applying optimization, control, and ML/AI methods to the understanding and modeling the system impacts of emerging technologies/systems in transportation such as connected and automated vehicles, new mobility services, and transportation electrification. He is an Associate Editor of *Transportation Research Part C: Emerging Technologies*, *IEEE Transactions on Intelligent Transportation Systems*, and *Intelligent Transportation Infrastructure*, and serves on the editorial board of *Transportation Research Part B: Methodological*.

Balázs Kulcsár received the M.S. degree in traffic engineering and the Ph.D. degree from the Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 1999 and 2006, respectively. He has been a Researcher/Post-Doctor with the Department of Control for Transportation and Vehicle Systems, BUTE, the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA, and the Delft Center for Systems and Control, Delft University of Technology, Delft, the Netherlands. He is currently a Professor with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. His main research interest focuses on traffic flow modeling and control.