8-2024

A Position Allocation Approach to the Battery Electric Bus Charging Problem

Alexander Brown
Utah State University

A POSITION ALLOCATION APPROACH TO THE BATTERY ELECTRIC BUS CHARGING PROBLEM

by

Alexander Brown

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Greg Droge, Ph.D. Major Professor	Dr. Jacob Gunther, Ph.D. Committee Member
Dr. Burak Sarsilmaz, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

ABSTRACT

A POSITION ALLOCATION APPROACH TO THE BATTERY ELECTRIC BUS CHARGING PROBLEM

by

Alexander Brown, Master of Science

Utah State University, 2024

Major Professor: Dr. Greg Droge, Ph.D.
Department: Electrical and Computer Engineering

With an increasing adoption of Battery Electric Bus (BEB) fleets, developing a reliable charging schedule is vital to a successful migration from their fossil fuel counterparts. In this work, a BEB charging scheduling framework that considers spatiotemporal schedule constraints, fixed route schedules, fast and slow charging, and battery dynamics is modeled as a Mixed Integer Linear Program (MILP). The MILP is modeled after the Berth Allocation Problem (BAP) in a modified form known as the Position Allocation Problem (PAP). Linear battery dynamics are included to model the charging of buses while at the station. To model the BEB discharges over their respective routes, it is assumed each BEB has an average kWh charge loss while on route. The optimization coordinates BEB charging to ensure that each vehicle remains above a specified state-of-charge (SOC). The model also minimizes the total number of chargers utilized and prioritizes slow charging for battery health. The model validity is demonstrated with a set of routes and is compared to a heuristic algorithm based on charge thresholds referred to as the Qin-Modified method.

The MILP approach is then further extended via a Simulated Annealing (SA) implementation. The framework maintains the same considerations and while further developing a method to minimize the demand cost. Two generation mechanisms are implemented for the SA algorithm denoted as the “quick” and “heuristic” implementations, respectively. Similarly, the model validity is demonstrated by utilizing a set of routes sampled form the Utah Transit Authority (UTA) and comparing the results to the MILP which is utilized as a baseline as it is formulated in a way that can guarantee optimality. The Qin-Modified is again presented as another means of comparison. The results presented show that the “heuristic” approach was able to generate a solution comparable to that of the MILP approach over similar execution times. The SA PAP framework is further extended to incorporate non-linear battery dynamics to further increase the accuracy of the SOC model during a BEBs charging phase.

(109 pages)

PUBLIC ABSTRACT

A POSITION ALLOCATION APPROACH TO THE BATTERY ELECTRIC BUS CHARGING PROBLEM

Alexander Brown

With an increasing adoption of Battery Electric Bus (BEB) fleets, developing a reliable charging schedule is vital to a successful migration from their fossil fuel counterparts. In this work, a BEB charging scheduling framework that considers fixed route schedules, multiple charger types, and battery dynamics is modeled as a Mixed Integer Linear Program (MILP). The MILP is modeled after the Berth Allocation Problem (BAP) in a modified form known as the Position Allocation Problem (PAP). The optimization coordinates BEB charging to ensure that each vehicle remains above a specified charge percentage. The model also minimizes the total number of chargers utilized and prioritizes slow charging for battery health. The model validity is demonstrated with a set of routes and is compared to a heuristic algorithm based on charge thresholds referred to as the Qin-Modified method.

The MILP approach is then further extended via a Simulated Annealing (SA) implementation. The framework maintains the same considerations and while further developing a method to minimize the peak power use (demand cost). Two mechanisms are implemented for the SA algorithm denoted as the “quick” and “heuristic” implementations, respectively. The model validity is demonstrated by utilizing a set of routes sampled form the Utah Transit Authority (UTA) and comparing the results two other models: the MILP approach and the Qin-Modified. The MILP is utilized as a baseline as it is formulated in a way that can guarantee optimality. The results presented show that the “heuristic” approach was able to generate a solution comparable to that of the MILP over a similar execution times. The SA PAP framework is further extended to incorporate non-linear battery dynamics to further increase the accuracy of the SOC model during a BEBs charging phase.

To my dearly beloved fiancée and family for all your unwavering support and encouragement over the years. May this work symbolize a token of my deepest gratitude and love in return.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to Dr. Greg Droge, my major professor, for the countless hours of reflecting, reading, encouraging, and, most of all, patience throughout the years. I thank you for the freedom that you provided to conduct my research while maintaining a guiding hand. Similarly, I wish to thank those from my committee, Dr. Jacob Gunther and Dr. Burak Sarsilmaz.

Alexander Brown

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

With an ever-increasing interest in the electrification of vehicles in the push for green transportation, many organizations and companies have been looking to adopt a fleet of electric vehicles [1]. This transition also stems into the electrification of public bus transportation via battery electric buses (BEBs) [2, 3]. In particular, agencies such as the Utah Transit Authority (UTA), have directed focus in replacing their fleets with BEBs. Alongside all the benefits that are associated with BEBs come new challenges that must be addressed prior to their integration into mainstream utilization. The energy storage capacity of BEBs is typically significantly less than their combustion counterparts while also have significantly longer refueling periods [2,4]. This is further complicated due to the care that must be taken in prolonging the lifespan of the battery [5–7], and the fact that BEB refueling is no longer a fixed cost (i.e. price per gallon multiplied by tank size). Utility companies, in addition to charging for the total energy consumed over a pay period, often further introduce a demand cost. The demand cost is based on the peak power drawn during the pay period (i.e., charging multiple BEBs simultaneously), and can significantly impact the overall monetary cost of maintaining the BEBs. Due to these factors, the charging schedules for the BEBs significantly impacts the overall cost of utilizing the fleet. This work introduces a scheduling framework for a fixed-schedule fleet of BEBs that utilizes linear and non-linear battery dynamics models, accounts for multiple charger types, allows partial charging, and attempts to minimize the monetary cost by considering the total energy consumed by the schedule as well as peak power use.

BEBs have been in service for many major markets, North America, Europe, and China, for more than a decade with expected growth in the near future [8]. The Asia Pacific market is forecasted to dominate the sales and some major companies of the industry have also begun to enter the global market such as Volvo, BYD, and Proterra by 2025

[8]. Much focus has been placed on the engineering of individual BEBs such as battery type, brake regenerative charging, optimal battery charging, and battery degradation [8–11]. The problems of route scheduling, charge scheduling, and optimizing the infrastructure are problems of recent interest, and are therefore timely and an ever increasingly relevant problems [12–14].

Literature shows an interest in solving the problem of assigning BEBs to charging queues or optimizing their infrastructure [12–15]. In fact, many works attempt to solve both problems simultaneously barring some simplification such as only allowing chargers of a single type [16,17], or if multiple chargers are allowed, then one of each type is assigned at separate locations. Others have also assumed that a BEB shall recieve full charge upon each arrival [14,15,18,19]. Some works on the stochastic effects of energy consumption while on route for a BEB as well as trip times [20,21]. One other source, as far as the research for this work has shown, describes a method of producing a BEB charge schedule wih a high fidelity while accounting for multiple charger types as well as being able to minimize over the total charger count [22].

The work described in this thesis is similar to the work in [22]; however provides an alternate method of modeling the problem in what is known as the Position Allocation Problem (PAP). The PAP is modeled after the Berth Allocation Problem (BAP), which was designed to optimally schedule cargo vessels to be berthed and serviced [23–25]. The PAP utilizes this notion to develop a model of assigning EVs to positions on a charger with a predefined charge duration [26].

Because the PAP's modeling is similar to that of the BAP, literature from the BAP provides a foundation for the development of the PAP. The BAP has been studied in literature since the 1990s and provides a depth of work to derive from [27]. The work to be introduced promises much potential for further research and development in regard to scheduling BEBs. What follows is a Mixed Integer Linear Program (MILP) and Simulated Annealing (SA) implementation of the PAP for BEB charge scheduling that consider relevant constraints for BEB schedules, considers linear and non-linear battery dynamics, and minimizes monetary cost.

The work proceeds as follows: Chapter 2 provides the state of the art along with various introductory material pertinent to this work. Chapter 3 constructs the MILP for BEB scheduling, including modifications to the PAP queuing constraints and the development of a dynamic charging model. An example is presented as a demonstration of the model's utility. The results are subsequently provided and discussed. In Chapter 4, the previously derived BEB charging model is adapted for a Simulated Annealing (SA) implementation. This method maintains the same considerations from the MILP implementation, but further accounts for a peak power demand cost. An example is then provided with discussion on the results. Chapter 5 further adapts the SA approach by deriving and incorporating non-linear battery dynamics. The example from Chapter 4 is run again utilizing the non-linear battery dynamics. The results are then presented and discussed.

CHAPTER 2
BACKGROUND AND RELATED WORK

The BEB queue scheduling problem is one that has been of increasing interest and relevancy in the EV/BEB industry. Thus, it is important to address the state of the art by introducing relevant problems in BEB queue scheduling, which will be addressed first. A detailed problem description shall then be presented to provide a basis of understanding for the work described in this thesis. Relevant introductory theory will be presented throughout.

2.1 State Of The Art of Battery Electric Bus Charging

Works concerning charge planning often use a version of the vehicle scheduling problem [16, 17, 28–31], while others have based their implementation on alternative methods [22, 26]. [22] utilizes a network flow approach to model the scheduling while [26] utilizes what is known as the Position Allocation Problem (PAP). Nearly all the literature reviewed considered consumption costs [18, 20, 29, 30, 32–35], while fewer consider demand costs [16, 32–35]. Many of these works introduce simplifying assumptions for the sake of computation. For example, some approaches only consider fast chargers during planing [13, 14, 19, 20, 31]. Approaches that consider more than one charger type typically isolate the specific charger types at different locations [16, 17].

Variants of this problem address infrastructure as well as determining existing buses that should be replaced by a BEB [18, 20, 29, 30]. Other works introduce a directed graph approach to model the flow of BEBs [22, 36], where this concept was expanded to simultaneously accounting for multiple charger types, partial charging, non-linear battery charge profiles [22]. The directed graph approach provides an easy method of modeling the scheduling by discretizing the time horizon to n_Q sets of nodes. The nodes represent the chargers availability and can have a maximum of one bus at a time. The buses can flow into a node to be charged and then later can exit allowing a new bus to enter. Another method similar to the directed graph that fits the modeling of the BEB charging scenario is the PAP [26]. The PAP is derived from the BAP which takes an input of vessel arrival times and outputs the selection of the berthing quay. The PAP utilizes this model and redefines its inputs to EV arrival times and outputs queues for the EVs to be charged. While the visits remain as discrete events, the time that the BEB is on the charger is modeled in continuous time similar to [20, 25, 26]. Due to the close relationship between the BAP and PAP, BAP literature may be used for the PAP. The literature shows methods of handling multiple quays (sets of chargers) to handle general berthing scenarios [25, 37]. Heuristic procedures for quicker solve times have also been introduced [24]. Methods of defining static (full-time horizon) and dynamic (rolling-time horizon) models have been created for daily and real-time solutions, respectively, and even fuzzy set theory has been applied to allow for more flexible schedules [38].

Others have assumed that BEBs always charge to full capacity [15, 18, 28, 29], partial charging utilizing a linear battery dynamics model [14, 16, 35], or non-linear battery dynamics with partial charging [22, 28, 32–34]. Works that assumed scheduled BEBs always charge to full capacity significantly simplify the scheduling problem, but eliminates the key factor in reducing the demand cost, partial charging [17, 20, 29, 30]. The approaches that utilized non-linear charging profiles with partial charging are able to achieve a reduction in the demand cost, with the added benefit of a higher fidelity at the expense of computation [28]. Exceptions to this are [16] that utilize a piecewise-linear charging profiles. This model has the drawback of assuming that a charger is always available. A common way to model the non-linear battery dynamics is utilizing Constant Voltage (CV), Constant Current (CC), and Constant Current Constant Voltage (CCCV) [9, 10]. A novel method of modeling the non-linear behavior present in [22] proposes a discrete linear time-invariant dynamic model that results in an exponential decay non-linear charge profile.

The selected model for the battery charge dynamics, although pertinent to this work as it directly affects the quality of the produced schedule, does not impact the considerations of battery health. Battery health begins to be of concern when over-charging, undercharging, or forcing the battery to perform "deep" cycles [6, 7, 18]. Furthermore, other works have suggested that charging a battery nearly to capacity is detrimental to the health and can significantly reduce the total charge cycles a battery may undergo [6,7]. While the charge profile for batteries are inherently non-linear, some works have assumed proportional charging as linear battery dynamics remain a valid assumption when the battery SOC is below 80% [36]. Thus, while linear dynamics may lack the fidelity of accuracy above 80%, it is able to accurately estimate the SOC within a range that will guarantee battery health. This work begins with an implementation of linear battery dynamics then incorporates a non-linear model suggested by [22].

2.2 Problem Description

The work of this thesis builds upon the Position Allocation Problem (PAP) [26], a modification of the well studied Berth Allocation Problem (BAP), as a means to schedule the charging of electric vehicles (EVs) [23–25]. The goal of the PAP is to allocate incoming EVs into queues to be charged as depicted in Figure 2.1. An example of a standard PAP/BAP solution (their visual representations are interchangeable) is visualized in Figure 2.2, note that the figure utilized BAP terminology. The x and y-axis represent time and queuing space, respectively. The figure discretizes the queuing space, but it may be continuous if desired. The shaded rectangles' widths represent their respective allocated charge times, and their heights represent the physical space taken by each EV.

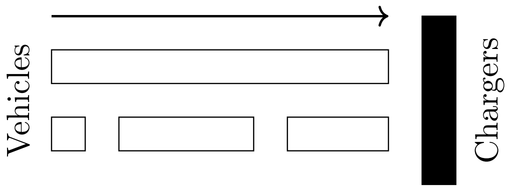


Fig. 2.1: Example of position allocation. Vehicles are placed in queues to be charged and move in the direction indicated by the arrow.

To adapt the PAP model for BEBs, consider a fleet of BEBs scheduled to perform a set

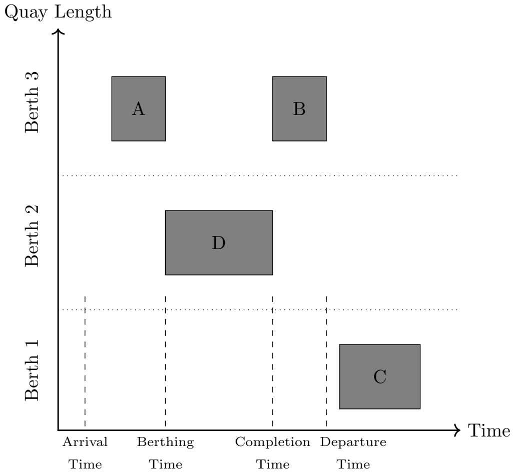


Fig. 2.2: The representation of the berth-time space. The x and y-axis represent time and space, respectively. Along the y-axis, the dashed lines represent discrete berthing locations. These locations may be chosen to be continuous. The shaded rectangles represent scheduled vessels to be serviced. The height of each shaded rectangle represents the space taken on the berth and the width being the time to service said vessel. The vertical dashed lines are associated with vessel D and represent the arrival time, berthing time, serviced completion time, and departure time. Note that the arrival time may be before the berthing time and the completion time may before the departure time.

of prescribed routes on a given day. An individual BEB from said fleet begins and completes an individual route at the same station from which it also receives its charge. During each route, the BEB's State of Charge (SOC) is depleted by a certain amount. The charge supplied during its visit must be enough to sustain the BEB's SOC at an appropriate level so that it may complete its next route. The charge may be supplied from any single charger given a set of chargers at the station. Let the term "arrival" describe the time at which a BEB reaches the station. Furthermore, let the term "visit" denote a BEB having arrived, awaited its predetermined time (whether it has received a charge or not), and departed from the station. Each BEB may have multiple visits to the station throughout their working day. This thesis describes a method to optimize the assignment of each visit to a charger given a schedule for a fleet of BEBs that follow the behavior described above. The various models presented in this work optimizes over peak power usage and energy consumption, as well as attempts to optimize the amount of chargers utilized. Both linear and non-linear battery dynamics are introduced and implemented.

2.3 Mixed Integer Linear Programming

A mixed integer linear programming (MILP) problem is a class of constrained optimization in which one seeks to find a set of continuous or integer values that maximizes or minimizes an objective function while satisfying a set of constraints [39]. Given an objective function J, decision variables (i.e. variables of optimization) $x_j \in \mathbb{R}$ and $y_k \in \mathbb{Z}^+$ (where $\mathbb{Z}^+$ denotes the set of non-negative integers), and input parameters $c_j, d_k, a_{ij}, g_{ik}, b_i \in \mathbb{R}$, a MILP has a mathematical structure [39]:

$$\text{max} \qquad J = \sum_j c_j x_j + \sum_k d_k y_k \tag{2.1a}$$

$$\text{subject to} \qquad \sum_j a_{ij} x_j + \sum_k g_{ik} y_k \leq b_i \qquad (i = 1, 2, ..., m) \tag{2.1b}$$

$$x_j \geq 0 \qquad (j = 1, 2, ..., n) \tag{2.1c}$$

$$y_k \in \mathbb{Z}^+ \qquad (k = 1, 2, ..., n). \tag{2.1d}$$

The objective function in Equation 2.1a comprises two parts, the continuous part, $\sum_j c_j x_j$, and integer part, $\sum_k d_k y_k$. The decision variable of the first part, x_j, is continuous whereas the decision variable of the second, y_j, is integer. Their respective input parameters may be integer or continuous, in the case of this example they are modeled as continuous. The objective function's utility is to provide a numerical score to a system (provided that a set of decision variables and input parameters are defined). While an individual score may not have any intrinsic meaning, it provides a method of ranking different solutions of the same model. The constraint equations (Equation 2.1b - Equation 2.1d) must all be satisfied for the output of an objective function to have any meaning. Thus, the constraint equations limit the solution space of the decision variables. Equation 2.1b states that the summation of the products of the respective continuous and integer decision variables and input parameters must be less than or equal to some value b_i. Equation 2.1c and Equation 2.1d state that the decision variables x_j and y_k, must be greater than or equal to

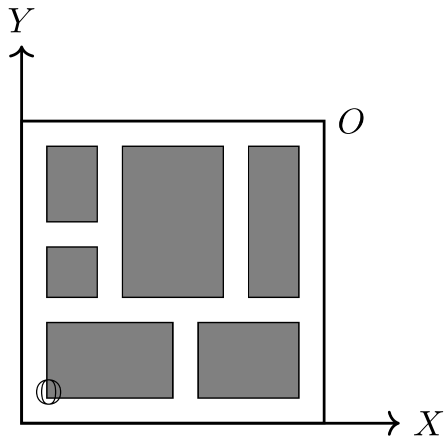


Fig. 2.3: Example of rectangle packing problem. The large square represented by O indicates the canstrained area that the set of shaded rectangles $\mathbb{O}$ must be placed within.

0, respectively.

2.4 Overview of the BAP

The BAP is a rectangle packing problem where a set of rectangles, $\mathbb{O}$, are attempted to be optimally placed in a larger rectangle, O, as shown in Figure 2.3. The rectangle packing problem is an NP-hard problem that can be used to describe many real-life problems [40,41]. In some of these problems, the dimensions of $\mathbb{O}$ are held constant such as in the problem of packing modules on a chip, where the widths and height of the rectangles represent the physical width and heights of the modules [41]. Other problems, such as the one presented in this work, allow either the horizontal or vertical edge of each rectangle in $\mathbb{O}$ to vary. As an example, suppose the vessel lengths are predefined (vertical edges are static), but the service time is allowed to vary (horizontal edges are dynamic). [23].

The BAP solves the problem of optimally assigning incoming vessels to berth positions to be serviced as shown in Figure 2.4. To relate to the rectangle packing problem, the width and height of O represent the time horizon T seconds and the berth length L meters, respectively. Similarly, the widths and heights of each element in $\mathbb{O}$ represent the time spent to service vessel i and the space taken by docking vessel i, respectively. In the BAP, the vessel characteristics (length of the vessel, arrival time, handling time, desired departure time) are assumed to be known for all vessels. A representation of a BAP solution is shown in Figure 2.2. The x and y-axis represent time horizon and berthing space, respectively. The gray squares, labeled A, B, C, and D, represent berthed vessels. The width of the boxes represents the time spent being serviced, and the height represents the amount of space the vessel requires on the berth. The vertical line adjacent to "Arrival Time" represents the actual time that the vessel arrives and is available to be berthed. "Berthing Time" is the time the vessel is berthed and begins being serviced. "Completion time" represents the time at which the berthing space becomes available again.

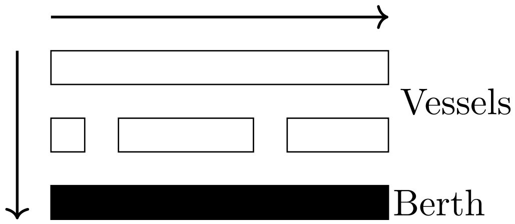


Fig. 2.4: Example of berth allocation. Vessels are docked in berth locations (horizontal) and are queued over time (vertical). The vertical arrow represents the movement direction of queued vessels and the horizontal arrow represents the direction of departure.

2.5 Overview of the PAP

The BAP forms the basis of the PAP; however, there are some differences in the way the variables are interpreted. The starting service time, u_i seconds, is viewed as the initial charge time, and the service time, total elapsed time spent on the charger. Similarly, for the spatial term, $v_i \in [0, L]$, the berth location is instead interpreted as the initial position on the charger. There are also a few clarifying concepts about how the system is modeled. The PAP models the set of chargers as one continuous line; that is, the natural behavior of the PAP model is to allow vehicles to be queued anywhere along $[0, L]$. Similarly, the charge times are continuous and can be placed anywhere on the time horizon, $[0, T]$, as long as the allocated times do not interfere with other scheduled charge times.

The PAP formulation's parameters can be divided into two categories: input parameters and decision variables. Each type will now be introduced in turn. The following parameters are assumed to be known inputs for the MILP. L defines the length of the charger in meters. As stated previously, it is modeled as a continuous bar meaning that a vehicle can be placed anywhere in the range $[0, L]$. It is assumed that the time horizon, T seconds, is known so that vehicles may be placed temporarily in the range $[0, T]$. The total number of visits to the station over the time horizon is represented by n_V. The arrival time for each visit is represented by a_i seconds, and the required charge time is represented by s_i seconds. The width of vehicle i is represented by l_i meters.

The decision variables provide the means by which the solver may optimize the problem. The initial and final charge times for vehicle i are u_i and d_i seconds, respectively. The starting position on the charger is denoted as $v_i \in [0, L]$ meters. The temporal ordering of vehicles i and j is determined by $\sigma_{ij} \in \{0, 1\}$, where $\sigma_{ij} = 1 \implies i$ arrives before j for all $1 \leq i, j \leq n_V$. Similarly, $\psi_{ij} \in \{0, 1\}$ determines the relative position of vehicles i and j on the charger: $\psi_{ij} = 1 \implies v_i < v_j$ for all $1 \leq i, j \leq n_V$. A value of zero conveys no information about the relative ordering. As an example, suppose $\psi_{ij} = 1$ and $\sigma_{ij} = 0$, from this it is known that EV i in a queue of a lesser index than j; however, $\sigma_{ij} = 0$ does not state whether visit i is scheduled before, during, or after visit j temporally. Similarly, suppose $\psi_{ij} = 0$ and $\sigma_{ij} = 1$, then it is known that visit i is scheduled before visit j, but it is unknown as to which queue visit i is scheduled relative to visit j.

To determine the values for each of these decision variables, a MILP was formulated in [26]. The formulation is shown in its entirety for completeness. The problem to be solved is

$$\min \sum_{i=1}^{n_v} (d_i - a_i) \tag{2.2}$$

Subject to:

$$u_j - u_i - s_i - (\sigma_{ij} - 1)T \geq 0 \tag{2.3a}$$

$$v_j - v_i - l_i - (\psi_{ij} - 1)L \geq 0 \tag{2.3b}$$

$$\sigma_{ij} + \sigma_{ji} + \psi_{ij} + \psi_{ji} \geq 1 \tag{2.3c}$$

$$\sigma_{ij} + \sigma_{ji} \leq 1 \tag{2.3d}$$

$$\psi_{ij} + \psi_{ji} \leq 1 \tag{2.3e}$$

$$s_i + u_i = d_i \tag{2.3f}$$

$$a_i \leq u_i \leq (T - s_i) \tag{2.3g}$$

$$\sigma_{ij} \in \{0, 1\},\ \psi_{ij} \in \{0, 1\} \tag{2.3h}$$

$$v_i \in [0, L] \tag{2.3i}$$

$$i, j = 1..n_V;\ i \neq j \tag{2.3j}$$

The objective function, Equation 2.2, minimizes the idle and service time by summing over the differences between the departure time, d_i, and arrival time, a_i for all visits. In other words, the objective function is searching for the schedule that removes each vehicle from the service queue as quickly as possible.

Equation 2.3a-Equation 2.3e are used to ensure that individual rectangles do not overlap. In terms of the PAP, this implies that there are no conflicts in the schedule spatially or temporally. Equation 2.3a establishes temporal ordering when active ($\sigma_{ij} = 1$) in the manner described previously by utilizing big-M notation. Similarly, Equation 2.3b establishes spatial ordering when active ($\psi_{ij} = 1$). Constraints Equation 2.3c-Equation 2.3e enforce spatial and temporal ordering between each queue/vehicle pair. Constraint Equation 2.3c ensures that there exists at least one spatial or temporal ordering between EVs i and j. Constraints Equation 2.3d and Equation 2.3e enforce validity of the assignments. For example, if Equation 2.3d resulted in a value of two, that would imply both vehicle i and j are scheduled before and after each other temporally, which is impossible. In the case of Equation 2.3e being equal to two, that would mean that vehicles i and j are scheduled both before and after one another on the charging strip, which is again impossible.

The last constraints force relationships between arrival time, initial charge time, and departure time. Equation 2.3f states that the initial charge time, u_i, plus the total charge time for, s_i, must equal the departure time, d_i. Equation 2.3g enforces the arrival time, a_i, to be less than or equal to the service start time, u_i, which in turn must be less than or equal to the latest time the vehicle may begin charging and stay within the time horizon. Equation 2.3h simply states that σ_{ij} and ψ_{ij} are binary terms. Equation 2.3i ensures that the assigned value of v_i is within the range, $[0, L]$.

CHAPTER 3
A MIXED INTEGER LINEAR PROGRAM APPROACH WITH LINEAR BATTERY DYNAMICS

3.1 Introduction

The objective of this chapter is to utilize the PAP from Section 2.5 as a basis of deriving the formulation for scheduling a fleet of BEBs. Because the PAP was designed for a set of EVs with predefined arrival times, initial charge times, and final charge times, the PAP must be modified to support the behavior that a fleet of buses exhibit. As such, the contribution of this work is the extension of the PAP's novel approach to BEB charger scheduling. This incorporates a proportional charging model into the MILP framework, includes consideration for multiple charger types, and consideration of each route in the schedule. The last contribution is of importance because both the BAP and PAP consider each arrival to be unique; thus, the tracking of battery charge from one visit to the next must be considered. Furthermore, the input parameters for the model can be predefined in such a manner as to minimize the number of fast and slow chargers utilized as well as minimize the energy consumption. That is, the model will simultaneously minimize the number of chargers as well as the total consumed energy. The result is a MILP formulation that coordinates charging times and charger type for every visit while considering a dynamic charge model with scheduling constraints.

The remainder of this chapter proceeds as follows: Section 3.2 constructs the MILP for BEB scheduling, including modifications to the PAP queuing constraints and the development of a dynamic charging model. Section 3.3 demonstrates an example of using the formulation to coordinate 35 buses over 338 total visits to the station.

3.2 A Rectangle Packing Formulation for BEB Charging

Applying the PAP to BEB charging requires four fundamental changes. The first is that the time that a BEB spends charging must be allowed to vary. That is, u_i, d_i, and s_i become variables of optimization. This is done primarily because chargers of various speeds are to be introduced. Allowing BEBs to have multiple visits that a charger decided upon during the optimization requires that the start and stop times must be changeable to respect the SOC constraints. Second, in the PAP each visit is assumed to be a different vehicle. For the BEB charging problem, each bus may make multiple visits to the station throughout the day. Thus, the resulting SOC for a bus at a given visit is dependent upon each of the prior visits. The third fundamental change is related to the first two. The SOC of each bus must be tracked to ensure that charging across multiple visits is sufficient to allow each bus to execute its route throughout the day. Finally, as previously stated, the PAP models the charger as one continuous bar. For the BEB, it will be assumed that a discrete number of chargers exist. Moreover, it is assumed that these chargers may have different charge rates.

A few assumptions are made in the derivation of the algorithm. As this work is not focused on estimating the discharge of a BEB during its route, the discharge for each route will be pre-calculated by assuming a fixed discharge rate kW multiplied by the route duration in hours. Secondly, it is assumed that the initial SOC of each BEB at the beginning of the day, $\alpha_b \kappa_b$, is larger than the minimum required SOC at the end of the day, $\beta_b \kappa_b$. Therefore, it must be assumed that the difference in the SOC can reach $\alpha_b \kappa_b$ by the beginning of the next working day.

The discussion of the four changes is separated into two sections. Section 3.2.1 discusses the changes in the spatial-temporal constraint formulation to form a queuing constraint. Section 3.2.2 then discusses the addition of bus charge management. This section ends with a brief discussion of a modified objective function and the statement of the full problem in Section 3.2.3. The notation is explained throughout and summarized in Table 3.1.

3.2.1 Queuing Constraints

The queuing constraints ensure that the buses entering the charging queues are assigned feasibly. There are three sets to differentiate between different entities. $\mathbb{B} = \{1, ..., n_B\}$ is the set of bus indices with index b used to denote an individual bus, $\mathbb{Q} = \{1, ..., n_Q\}$ is the set of queues with index q used to denote an individual queue, and $\mathbb{V} = \{1, ..., n_V\}$ is a set of visits to the station with i and j used to refer to individual visits. The mapping $\Gamma : \mathbb{V} \rightarrow \mathbb{B}$ is used to map a visit index, i, to a bus index, b. The notation Γ_i is used as a shorthand to refer to the bus index b for visit i.

The actual physical dimensions of the BEB are ignored, and it is assumed that each BEB will be assigned to charge at a particular charger. Because of this assumption, the PAP spatial variable, l_i, may be removed and v_i is made to be an integer corresponding to which queue visit i will be using, $v_i \in \mathbb{Q}$. That is, the queue position is now discretized over n_Q queues where a BEB occupies single charge queue. Thus, when $\psi_{ij} = 1$, vehicle j is placed in a charging queue with a larger index than vehicle i, $v_j > v_i$. The charger length L is likewise replaced with n_Q. Note that $n_Q = n_B + n_C$, where n_B is the number of buses and n_C is the number of chargers. The rationale for adding additional idle queues is to allow BEBs to be "set aside" if no additional charge is required. Adding one idle queue for each BEB ensures that the constraints will be satisfied if multiple buses sharing overlapping times at the station are placed in idle queues. This method will be applied when defining the parameters in Section 3.3. The modified queuing constraints can be written as shown in Equation 3.1.

$$v_i - v_j - (\psi_{ij} - 1)n_Q \geq 1 \tag{3.1a}$$

$$d_i \leq \tau_i \tag{3.1b}$$

$$s_i \geq 0 \tag{3.1c}$$

$$v_i \in \mathbb{Q} \tag{3.1d}$$

The constraint in Equation 3.1a is nearly identical to Equation 2.3b, but rather than viewing the charger as a continuous strip of length L, it is discretized into n_Q queues each with a width of unit length one. A BEB is also assigned a unit length of one which is reflected in Equation 3.1a by $\cdot \geq 1$. Equation 3.1b ensures that the time the BEB is detached from the charger, d_i, is before its departure time, τ_i seconds. Note the introduction of the new variable τ_i exists to allow the final charge time to be independent a similar manner that the inital charge time need not coincide with the arrival time, $a_i \leq u_i \leq d_i \leq \tau_i$. Equation 3.1d defines the of integers that v_i that represent the n_Q chargers.

3.2.2 Battery Charge Dynamic Constraints

Battery dynamic constraints are now to be introduced. Two constraints are enforced on the SOC for each BEB: the SOC must always remain above a specified percentage to guarantee sufficient charge to execute their respective routes and each bus must end the day with an SOC above a specified threshold, preparatory for the next day.

The SOC upon arrival for visit i is denoted as η_i kWh. Because the SOC for a visit i is dependent on its previous visits, the mapping $\Upsilon : \mathbb{V} \rightarrow \mathbb{V} \bigcup \{\varnothing\}$ is used to determine the next visit that corresponds to the same bus, with Υ_i being shorthand notation. Thus, Γ_j and Γ_{Υ_i}, for $\Upsilon_i = j$, would both map to the same bus index as long as Υ_i is not the null element, $\varnothing$. The null element is reserved for BEBs that have no future visits.

To drive time spent on the charger, s_i, as well as define initial, final, and intermediate bus charges for each visit i, the sets for initial and final visits must be defined. Let the mapping of the first visit by each bus be denoted as $\Gamma^0 : \mathbb{B} \rightarrow \mathbb{V}$. The resulting value of the mapping Γ^0 represents the index for the first visit of bus b. Similarly, let $\Gamma^f : \mathbb{B} \rightarrow \mathbb{V}$ maps the indices for the final visits for each bus $b \in \mathbb{B}$. Let the storthand for each mapping be denoted as Γ_b^0 and Γ_b^f, respectively. The initial and final bus charge percentages, α and β, can then be represented by the constraint equations $\eta_{\Gamma_b^0} = \alpha_b \kappa_b$ and $\eta_{\Gamma_b^f} = \beta_b \kappa_b$, respectively. The intermediate charges must be determined during runtime.

It is assumed that the charge received is proportional to the time spent charging. The rate for charger q is denoted as r_q kW. Note that a value of $r_q = 0$ corresponds to a queue where no charging occurs. A bus in such a queue is simply waiting at the station for the departure time. The queue indices are ordered such that the first n_B queues have $r_q = 0$

to allow an arbitrary number of buses to sit idle at any given moment in time. The next n_C queues are reserved for the slow and fast chargers. The amount of discharge between visits i and Υ_i, the next visit of the same bus, is denoted as Δ_i kWh. If visit i occurred at charger q, the SOC of the BEB's next arrival, Υ_i, would be $\eta_{\Upsilon_i} = \eta_i + s_i r_q - \Delta_i$.

The binary decision variable $w_{iq} \in \{0,1\}$ is introduced to indicate the active charger for visit i in vector form. The form of the SOC for the next visit, Υ_i, can be written using the following constraints.

$$\eta_{\Upsilon_i} = \eta_i + \sum_{q=1}^{n_Q} s_i w_{iq} r_q - \Delta_i \tag{3.2a}$$

$$\sum_{q=1}^{n_Q} w_{iq} = 1 \tag{3.2b}$$

$$w_{iq} \in \{0,1\}. \tag{3.2c}$$

The choice of queue for visit i, becomes a slack variable and is defined in terms of w_{iq} as

$$v_i = \sum_{q=1}^{n_Q} q w_{iq}. \tag{3.3}$$

Maximum and minimum values for the charges are included to ensure that the battery is not overcharged and to guarantee sufficient charge for subsequent visits. The upper and lower battery charge bounds for bus b are κ_b and $\nu_b \kappa_b$, respectively , where κ_b is the battery capacity and ν_b is a percent value. The upper and lower bounds for the current SOC are written as follows.

$$\eta_i + \sum_{q=1}^{n_Q} s_i w_{iq} r_q \leq \kappa_{\Gamma_i} \tag{3.4a}$$

$$\eta_i \geq \nu_{\Gamma_i} \kappa_{\Gamma_i} \tag{3.4b}$$

Equation 3.4a ensures that the BEB SOC does not exceed the battery capacity, and Equation 3.4b enforces that the inital SOC for each visit is above the threshold of $\nu_{\Gamma_i}\kappa_{\Gamma_i}$. Note that the term $s_i w_{iq}$ is a bilinear term. A standard way of linearizing a bilinear term that contains an integer variable is by introducing a slack variable with an either/or constraint [39, 42]. Allowing the slack variable g_{iq} seconds to be equal to $s_i w_{iq}$, g_{iq} can be defined as

$$g_{iq} = \begin{cases} s_i & w_{iq} = 1 \\ 0 & w_{iq} = 0 \end{cases}. \tag{3.5}$$

Equation 3.5 can be expressed as a mixed integer constraint using big-M notation with the following four constraints.

$$s_i - (1 - w_{iq})M \leq g_{iq} \tag{3.6a}$$

$$s_i \geq g_{iq} \tag{3.6b}$$

$$M w_{iq} \geq g_{iq} \tag{3.6c}$$

$$0 \leq g_{iq} \tag{3.6d}$$

where M is a large unitless value. If $w_{iq} = 1$ then Equation 3.6a and Equation 3.6b become $s_i \leq g_{iq}$ and $s_i \geq g_{iq}$, forcing $s_i = g_{iq}$ with Equation 3.6c being inactive. If $w_{iq} = 0$, Equation 3.6a is inactive and Equation 3.6c and Equation 3.6d force $g_{iq} = 0$.

3.2.3 The BEB Charging Problem

The goal of the MILP is to utilize chargers as little as possible to reduce energy costs with fast charging being penalized more to avoid the adverse effects of fast charging on battery health as well as the larger usage cost. Thus, an assignment cost m_q and usage cost ϵ_q are associated with each charger, q. These unitless weights can be adjusted based on charger type or time of day that the visit occurs. The assignment term takes the form

$w_{iq}m_q$, and the usage term takes the form $g_{iq}\epsilon_q$. The resulting BEB charging problem is defined in Equation 3.7.

$$\min \sum_{i=1}^{N}\sum_{q=1}^{n_Q}\left(w_{iq}m_q + g_{iq}\epsilon_q\right) \tag{3.7}$$

Subject to the constraints

$$u_j - u_i - s_i - (\sigma_{ij} - 1)T \geq 0 \tag{3.8a}$$

$$v_j - v_i - (\psi_{ij} - 1)n_Q \geq 1 \tag{3.8b}$$

$$\sigma_{ij} + \sigma_{ji} + \psi_{ij} + \psi_{ji} \geq 1 \tag{3.8c}$$

$$\sigma_{ij} + \sigma_{ji} \leq 1 \tag{3.8d}$$

$$\psi_{ij} + \psi_{ji} \leq 1 \tag{3.8e}$$

$$s_i + u_i = d_i \tag{3.8f}$$

$$\eta_{\Gamma_b^0} = \alpha_{\Gamma_i}\kappa_{\Gamma_i} \tag{3.8g}$$

$$a_i \leq u_i \leq (T - s_i) \tag{3.8h}$$

$$d_i \leq \tau_i \tag{3.8i}$$

$$\eta_i + \sum_{q=1}^{n_Q} g_{iq}r_q - \Delta_i = \eta_{\gamma_i} \tag{3.8j}$$

$$\eta_i + \sum_{q=1}^{n_Q} g_{iq}r_q - \Delta_i \geq \nu_{\Gamma_i}\kappa_{\Gamma_i} \tag{3.8k}$$

$$\eta_i + \sum_{q=1}^{n_Q} g_{iq}r_q \leq \kappa_{\Gamma_i} \tag{3.8l}$$

$$\eta_{\Gamma_b^f} \geq \beta_{\Gamma_f}\kappa_{\Gamma_f} \tag{3.8m}$$

$$s_i - (1 - w_{iq})M \leq g_{iq} \tag{3.8n}$$

$$s_i \geq g_{iq} \tag{3.8o}$$

$$Mw_{iq} \geq g_{iq} \tag{3.8p}$$

$$0 \leq g_{iq} \tag{3.8q}$$

$$v_i = \sum_{q=1}^{n_Q} q w_{iq} \tag{3.8r}$$

$$\sum_{q=1}^{n_Q} w_{iq} = 1 \tag{3.8s}$$

$$w_{iq}, \sigma_{ij}, \psi_{ij} \in \{0, 1\} \tag{3.8t}$$

$$v_i, q_i \in \mathbb{Q} \tag{3.8u}$$

$$i \in \mathbb{V} \tag{3.8v}$$

Equation 3.8a-Equation 3.8i are reiterations of the queuing constraints in Equation 3.1. Equation 3.8g-Equation 3.8m provide the battery charge constraints. Equation 3.8n-Equation 3.8q define the charge gain of every visit/queue pairing. The last constraints Equation 3.8t-Equation 3.8v define the sets of valid values for each variable.

3.3 Example

An example will now be presented to demonstrate the utility of the developed MILP charge scheduling technique. A description of the scenario is first presented followed by a description of an alternative heuristic-based planning strategy called Qin-Modified which is used as a comparison to the MILP PAP. Results are then presented, analyzed and discussed for each of the planning strategies.

3.3.1 BEB Scenario

To display the capabilities of the model, an example scenario is presented. The scenario was run over a time horizon of $T = 24$ hours, utilizing $n_B = 35$ buses with $n_V = 338$ visits divided between the n_B buses. As stated before, the route times are sampled from a set of routes from the UTA. Each bus has a battery capacity of $\kappa_b = 388$ kWh that is required to stay above an SOC of $\nu_b = 25\%$ (97 kWh) $\forall b \in \mathbb{B}$ to ensure that each BEB can complete its next route in addition to maintaining battery health. Each bus is assumed to begin the working day with an SOC of $\alpha_b = 90\%$ $\forall i \in \mathbb{V}$ (349.2 kWh). Additionally, each bus is required to end the day with a minimum SOC of $\beta_b = 70\%$ (271.6 kWh) $\forall i \in \mathbb{V}$. This assumes that overnight charging can account for the deficient 20% SOC. Each bus is assumed to discharge at a rate of $\zeta_b = 30$ kW. Note that many factors play a role in the rate of discharge; however, for the sake of simplicity since the discharge calculation is out of the scope of this work, an average rate is used. A total of $n_C = 30$ chargers are utilized where 15 of the chargers are slow charging (30 kW) and 15 are fast charging (911 kW). The technique to minimize the total charger count will now be employed.

To encourage the MILP PAP problem to utilize the fewest number of chargers, the value of m_q in the objective function, Equation 3.7, is $\forall q \in \{1, 2, ..., n_B\}; m_q = 0$ and $\forall q \in \{n_B + 1, n_B + 2, ..., n_B + n_C\}; m_q = 1000q$. The charge duration scalar, ϵ_q, is defined as $\epsilon_q = r_q$ to create a consumption cost term, $g_{iq}\epsilon_q$ kWh. By consumption cost, it is meant that the total energy consumed by the charge schedule will be accounted for in the objective function. This method encourages the model to minimize active charger times, particularly for the fast chargers.

Another heuristic-based optimization strategy, referred to as Qin-Modified, is also employed as a means of comparison with the results of the MILP PAP. The Qin-Modified strategy is based on the threshold strategy of [34]. The strategy has been modified slightly to accommodate the case of multiple charger types without an exhaustive search for the best charger type. The heuristic is based on a set of rules that revolve around the initial SOC of the bus visit i. There are three different thresholds, low (85%), medium (90%), and high (95%). Buses below the low threshold (SOC $\leq$ 85%) are prioritized to fast chargers and then are allowed to utilize slow chargers if no fast chargers are available. Buses between the low and medium threshold (85% $<$ SOC $\leq$ 90%) prioritize slow chargers first and utilizes fast chargers only if no slow chargers are available. Buses above the medium threshold and below the high (90% $<$ SOC $\leq$ 95%) will only be assigned to slow chargers. Buses above the high threshold (SOC $>$ 95%) will not be placed in a charging queue. Once a bus has been assigned to a charger, it remains on the charger for the duration of the time it is at the station, or it reaches an SOC of 95% charge, whichever comes first.

The total number of constraints resulted in 7,511 continuous and 328,282 integer/binary constraints. The optimization was performed using the Gurobi MILP solver [43] on a machine equipped with an AMD Ryzen 9 5900X 12 - Processor (24 core) at 4.95GHz. The solver was allowed to run for 4.2 seconds.

3.3.2 Results

The schedule generated by the Qin-Modified strategy and the MILP PAP is shown in Figure 3.1a and Figure 3.1b, respectively. The x-axis represents the time in hours. The y-axis represents the assigned charging queue. Rows between 0 and 14 are active times for slow chargers, and rows in the range of 15 and 29 are active times for fast chargers. The circle with an 'X' represent the starting charge time for a bus b with the line to the vertical tick signifying the region of time the charger is active. When a circle vertex contains what appears to be an asterisk, that indicates the previous BEB charge time ending and the next BEB charge beginning.

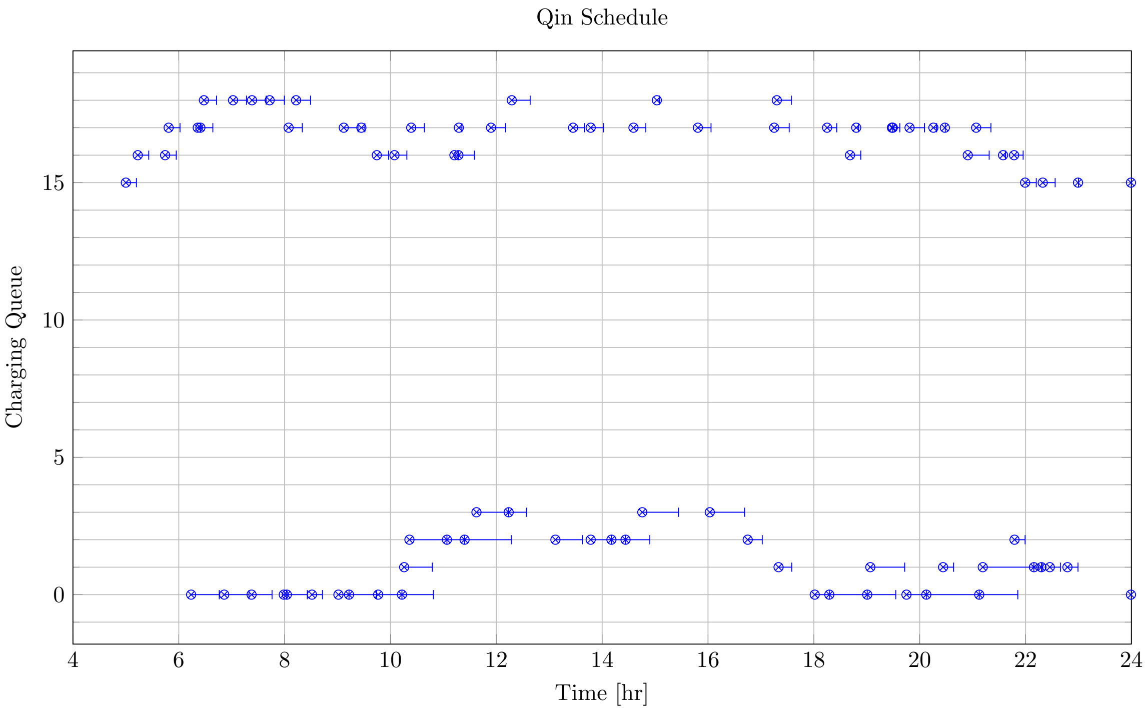
The first observation is in the choice of preferred chargers between the Qin-Modified and MILP scheduler. Looking at Figure 3.2b and Figure 3.2a, the Qin-Modified schedule uses at most four fast chargers and three slow at the same time, whereas the MILP schedule uses at most one fast charger and six slow at the same time. Both the Qin-Modified and MILP schedule used the fast chargers in short bursts (˜0.2-0.5 hours). The main difference lies in the utilization strategy of the slow chargers. The Qin-Modified, for the most part, opted for shorter bursts for the slow chargers (˜0.3-0.7 hours), most heavily placed on the first slow charger. The MILP utilized the slow chargers in short bursts; however, the solver was able to recognize moments where a BEB being placed in a slow charging queue for a longer duration was more cost-effective (with respect to the objective function) than placing the BEB in a fast charging queue. Although one of the MILP's objectives is to minimize the amount of chargers used, the Qin-Modified schedule ended up using fewer chargers than the MILP. Note the MILP schedule packed the first queue for the fast and slow chargers more effectively than the Qin schedule. Although both schedules generated are valid, no comparison of the quality of the schedule can be made directly from Figure 3.1b and Figure 3.1a.

Figure 3.3a and Figure 3.3b depicts the SOC for every bus over the time horizon for the Qin and MILP schedules, respectively. Every vehicle begins with an SOC of $\alpha_b = 90\%$, finishes with an SOC of $\beta_b = 70\%$ in the MILP PAP schedule, and never goes below 25% in the intermediate arrivals as stated in constraint Equation 3.8. There is no guarantee for this in the Qin-Modified strategy which can be seen by some intermediate charges reaching an SOC of 0% as well as the distribution of final charges, the minimum being 0% and the maximum 94.845%. The only sense of guarantee that the Qin-Modified supplies is its predictability within the intermediate visits due to its heuristic nature (i.e. if the BEB charge is within the low threshold, a fast charger will be prioritized); whereas MILP places a bus in the queue that "makes sense" in respect to the larger picture. The MILP PAP does not have an obvious sense of decision-making due to weighted objective function that is affected by the accumulation of decisions made prior.

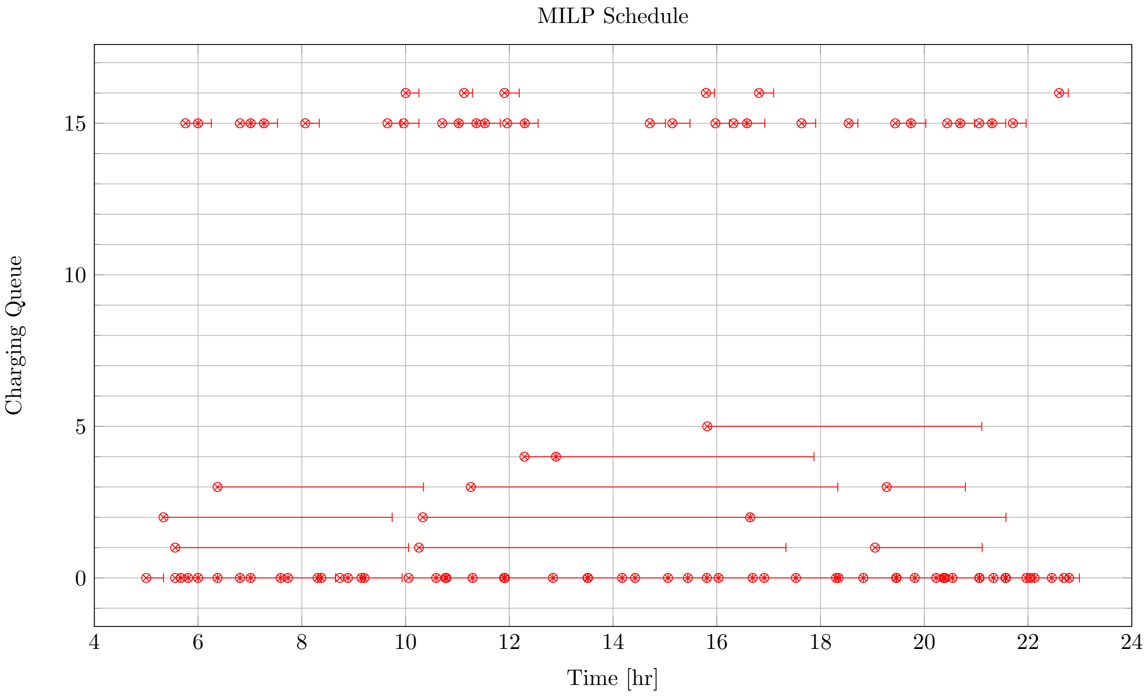
Another important measure for the chargers is to compare the amount of power and energy consumed. Figure 3.4 depicts the power consumption throughout the time horizon. It can be seen that the Qin-Modified power consumption is steadily less or the same as the MILP schedule. This can be accounted for by the MILP's constraints to keep the bus SOC above 25% and to reach a final SOC of 70% at the end of the working day. Along a similar vein, the accumulated energy consumed is shown in Figure 3.5. The MILP schedule is more efficient up until about the eleventh hour. Again, this can be accounted for by the fact the MILP is accommodating the extra constraints. Due to these constraints the MILP PAP consumes about $0.1 \cdot 10^4$ kWh more than the Qin-Modified. The overlap of the MILP PAP can be accounted for by referencing Figure 3.2a and Figure 3.2b. Between the fifth and tenth hour, the MILP schedule heavily uses slow chargers increasing the rate at which power is being consumed. Afterwards, the MILP schedule at a minimum continues to use the same amount of chargers as the Qin Schedule. Again, due to the added constraints, the MILP schedule must utilize more resources to keep within the specified bounds.

Table 3.1: Notation used throughout Chapter 3.

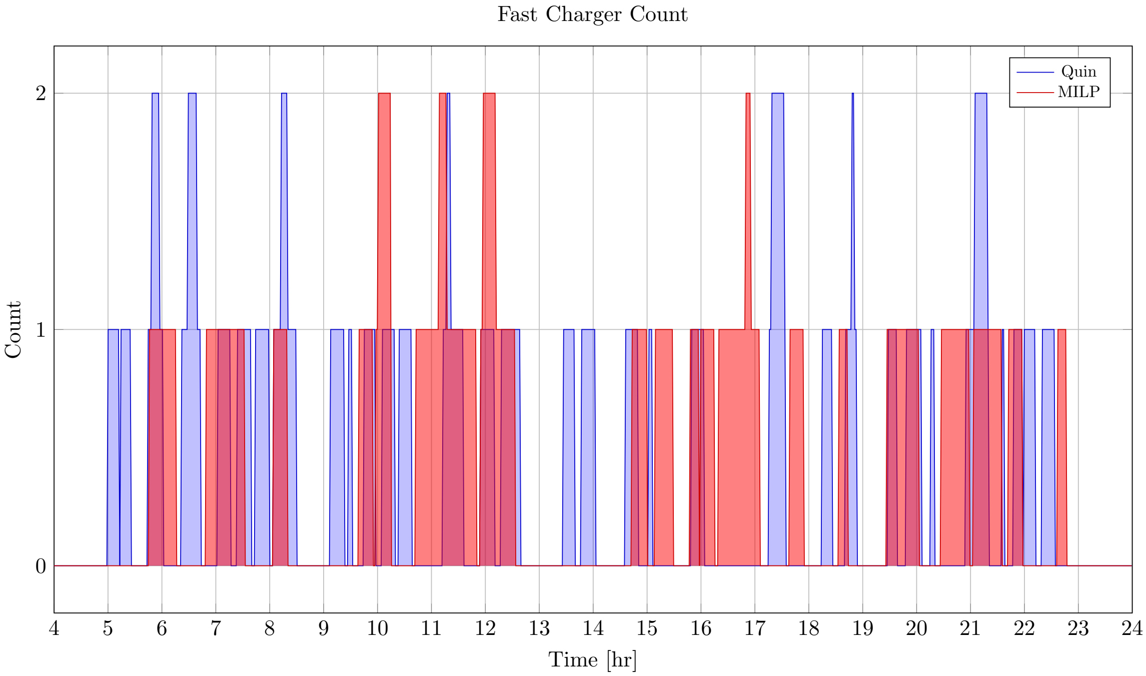
Variable	**Description**	**Variable**	**Description**
Constants		Constants	
n_B	Number of buses	M	An arbitrarily large number
n_V	Number of total visits	n_Q	Number of queues
n_C	Number of chargers	$\mathbb{V}$	Set of visit indices, $\mathbb{V} = \{1, ..., n_V\}$
$\mathbb{B}$	Set of bus indices, $\mathbb{B} = \{1, ..., n_B\}$	$\mathbb{Q}$	Set of queue indices, $\mathbb{Q} = \{1, ..., n_Q\}$
i, j	Indices used to refer to visits	b	Index used to refer to a bus
q	Index used to refer to a queue		
Input Parameters		Input Parameters	
Γ	$\Gamma : \mathbb{V} \to \mathbb{B}$ with Γ_i used as a shorthand to denote the bus b for visit i		
α_b	Initial charge percentage time for bus b	β_b	Final charge percentage for bus b at the end of the time horizon
ϵ_q	Cost of using charger q per unit time	Υ	$\Upsilon : \mathbb{V} \to \mathbb{V}$ mapping a visit to the next visit by the same bus Υ_i being the shorthand.
κ_b	Battery capacity for bus b	Δ_i	Discharge of visit over route i
ν_b	Minimum charge allowed for bus b	τ_i	Time visit i must depart the station
ζ_b	Discharge rate for bus b	a_i	Arrival time of visit i
i_0	Indices associated with the initial arrival for every bus in $\mathbb{B}$	i_f	Indices associated with the final arrival for every bus in $\mathbb{B}$
m_q	Cost of a visit being assigned to charger q	r_q	Charge rate of charger q per unit time
Decision Variables		Decision Variables	
ψ_{ij}	Binary variable determining spatial ordering of vehicles i and j	η_i	Initial charge for visit i
σ_{ij}	Binary variable determining temporal ordering of vehicles i and j	d_i	Ending charge time for visit i
g_{iq}	Linearization term, represents the multiplication of $s_i w_{iq}$	s_i	Amount of time spent on charger for visit i
u_i	Starting charge time of visit i	v_i	Assigned queue for visit i
w_{iq}	Binary assignment variable for visit i to queue q		



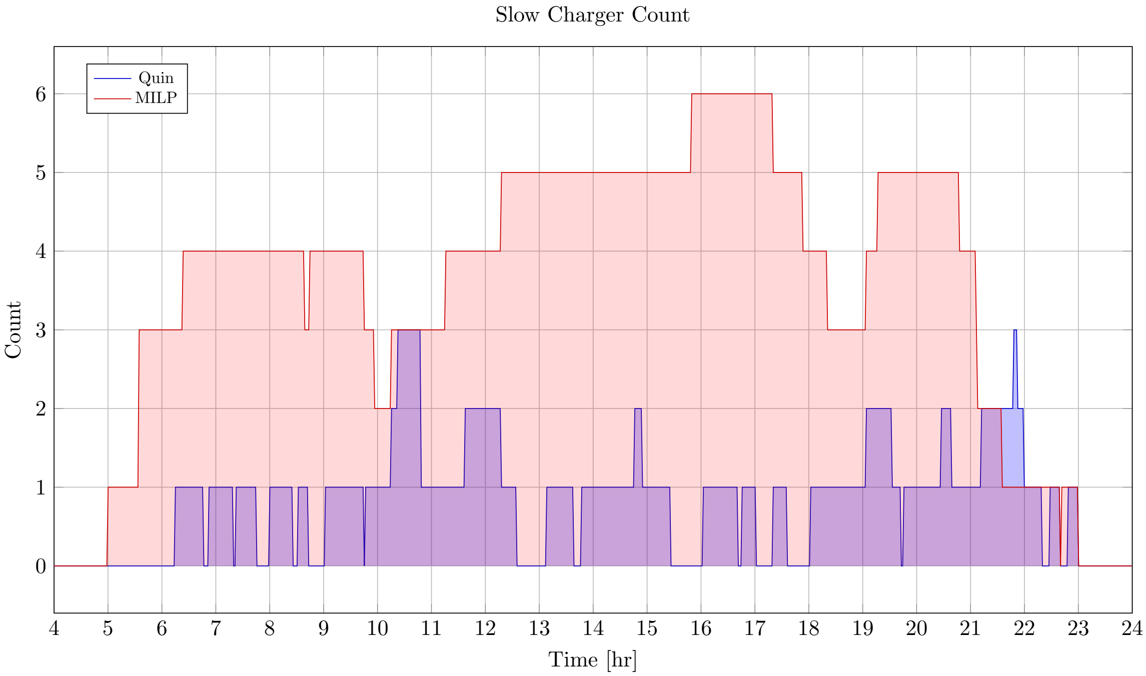
(a) Charging schedule generated by Qin Modified algorithm.



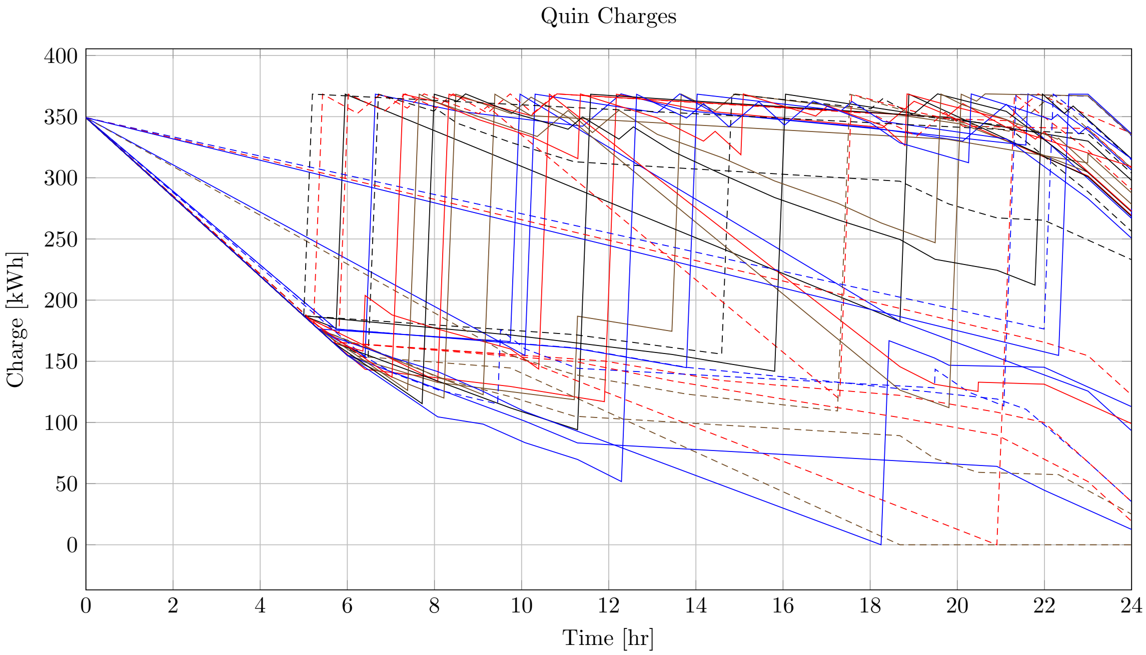
(b) Charging schedule generated by MILP PAP algorithm.



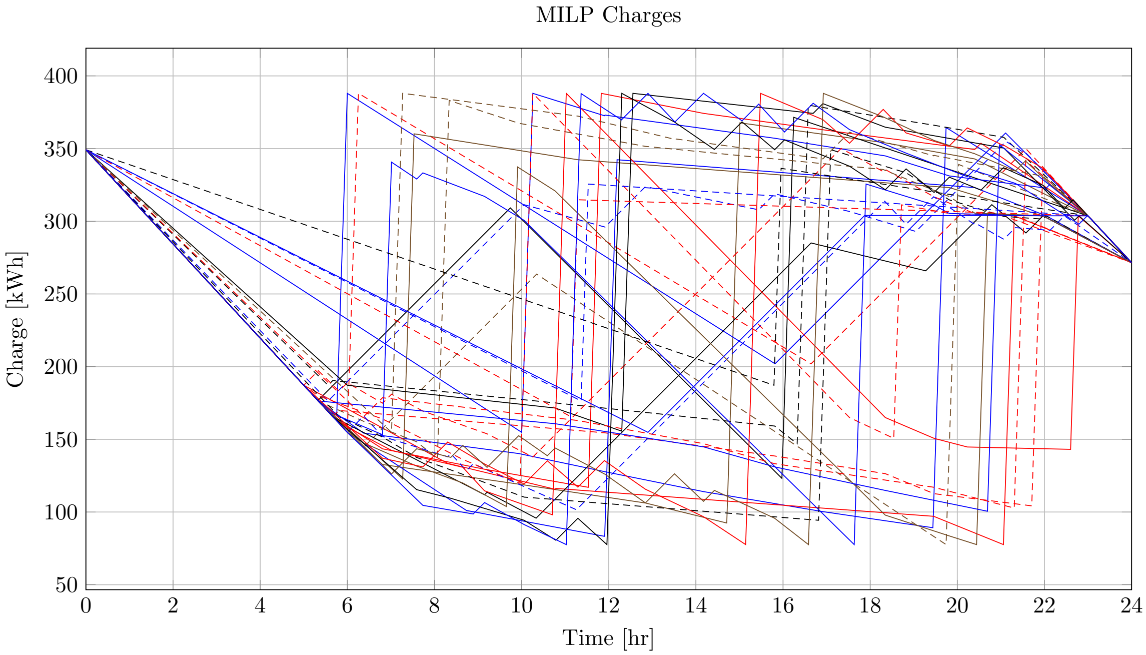
(a) Number of fast chargers for Qin and MILP PAP.



(b) Number of slow chargers for Qin and MILP PAP.



(a) Bus charges for the Qin Modified charging schedule. The charging scheme of the Qin charger is more predictable during the working day.



(b) The bus charges for the MILP PAP charging schedule. The MILP model allows for guarantees of minimum/maximum changes during the working day as well as charges at the end of the day.

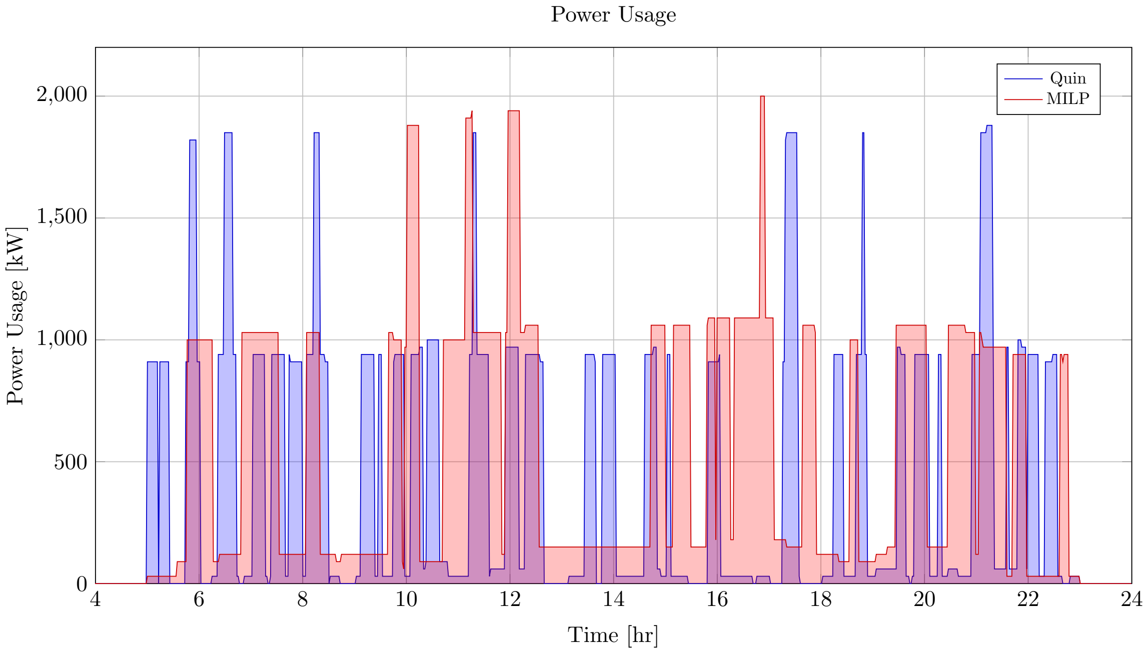


Fig. 3.4: Amount of power consumed by Qin-Modified and MILP schedule over the time horizon.

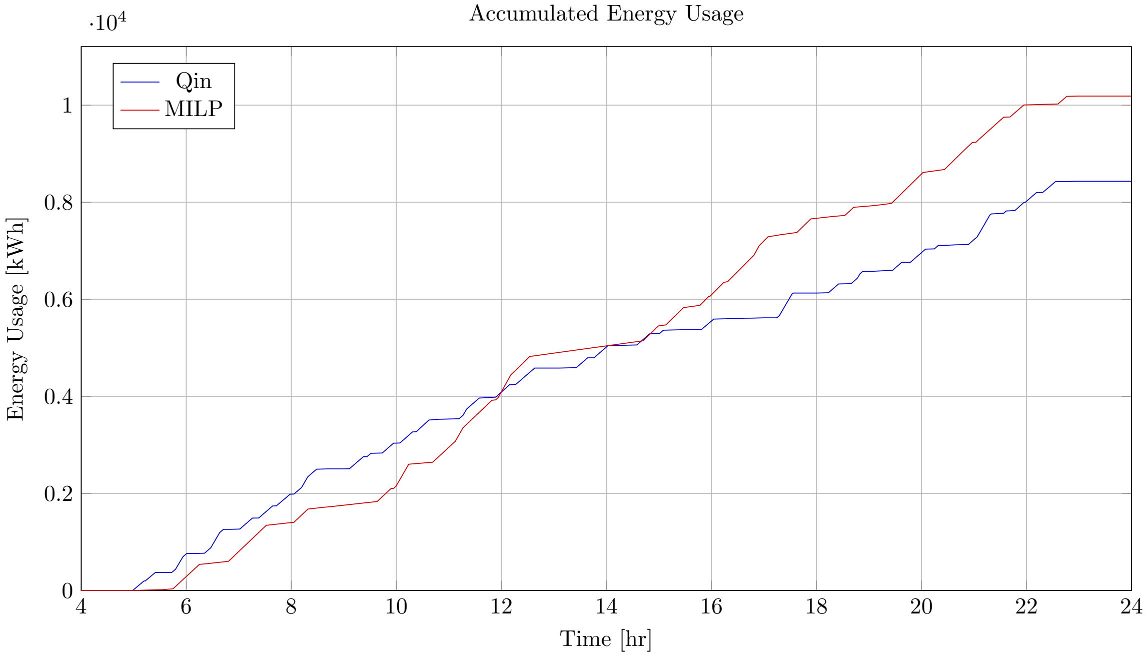


Fig. 3.5: Total accumulated energy consumed by the Qin-Modified and MILP schedule throughout the time horizon.

CHAPTER 4
A SIMULATED ANNEALING APPROACH WITH LINEAR BATTERY DYNAMICS

4.1 Introduction

In the previous chapter a MILP was derived to create an optimal charging schedule for a fleet of BEBs. While MILPs are extensible via the modular nature of the constraints, they are limited by the required linearity of the equations. If a non-linear equation is able to be linearized, that often leads to an introduction of slack variables that can further increase the complexity of the model, as was seen in Chapter 3. The chapter aims to expand on the MILP approach by introducing a Simulated Annealing (SA) framework that utilizes linear battery dynamics, accounts for partial charging, minimizes total charger count, allows for multiple charger types, minimizes consumption cost, and minimizes demand cost. These contributions are demonstrated via simulation and comparison to two other models: the Mixed Integer Linear Program (MILP) implementation of the PAP and what is known as the Qin-Modified technique.

The remainder of this chapter goes as follows. provides the problem statement associated with this work. provides a description of the input parameters and decision variables then introduces the structure of the formulation. In , the concept and theory of SA is introduced. In particular, the algorithms and methods utilized for the SA implementation for this work are discussed. combines the previous sections to introduce the particular pseudocode for the SA PAP. In , an example problem is provided to demonstrate the capability of the work provided in this paper. The results will be presented and discussed.

4.2 Problem Description

Consider a fleet of BEBs scheduled to perform a set of prescribed routes on a given day. An individual BEB from said fleet begins and completes an individual route at the same station from which it also receives its charge. During each route, the BEB's State of Charge (SOC) is depleted by a certain amount. The charge supplied during its visit must be enough to sustain the BEB's SOC at an appropriate level so that it may complete its next route. Provided there is a set of chargers at the station, the bus may be placed in any single queue to receive its charge. Let the term "arrival" describe the time at which a BEB reaches the station. Furthermore, let the term "visit" denote a BEB having arrived, awaited its predetermined time (whether it has received a charge or not), and departed from the station. Each BEB is allowed to have multiple visits throughout the working day.

Because each bus may visit the station more than once, let the previously considered fleet contains n_B BEBs that collectively visit a station n_V times. At said station, let there exist a pool of n_Q charging queues from which a visiting BEB may be assigned. Upon arrival to the station, a bus is admitted to one of the n_Q queues for charging. Each queue represents a charger that supplies the bus with a charge at a particular rate or allows the bus to sit idle when no charging is required (i.e., a charge rate of zero). The set of possible queue indices is denoted as $Q \in \{1, ..., n_Q\} \subset \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers. It is assumed that charger $q \in Q$ has an associated charge rate, denoted as r_q. Let the set of arrivals be written as $I = \{1, ...n_V\} \subset \mathbb{Z}$, and let each BEB be prescribed an identification number from the set $B = \{1, ..., n_B\} \subset \mathbb{Z}$. As such, each visit can be represented by the tuple: $(b_i, a_i, e_i, u_i, d_i, q_i, \eta_i, \xi_i)$, in which the elements within the tuple denote the visit index, $i \in I$, BEB identification number, $b \in B$, arrival time to the station, $a \in \mathbb{R}$, departure time from the station, $e \in \mathbb{R}$, time at which the BEB begins charging, $u \in \mathbb{R}$, time at which the BEB ends charging, $d \in \mathbb{R}$, the charger queue for the BEB to be placed into, $q \in Q$, the SOC upon arrival, $\eta \in \mathbb{R}$, and the index of the next visit for the currently visiting BEB, $\xi \in I \cup \varnothing$. The null element, $\varnothing$, is used to specify when a BEB has no future visits. Let the set of visits be denoted as $\mathbb{I}$ where the i^{th} visit is denoted is $\mathbb{I}_i$. Furthermore, let a particular item from the tuple for visit i to be written as $\cdot_i$. For example, the arrival time for visit i is written as a_i.

The amount of time the BEB is allowed to charge during visit i is dictated by the scheduled arrival time and required departure time, $[a_i, e_i]$. Partial charging is allowed; however, the SOC may not exceed the BEB battery capacity, κ_b, and the SOC is desired to stay above some minimum percentage of the battery capacity, $\nu_b \in [0, 1]$. The battery dynamics in this work is modeled as linear, which remains accurate up to about an SOC of 80% [36]. Note that excessively charging the BEBs is undesirable due to battery health concerns as at higher SOCs overshoot become a concern and may also cause the battery to undergo deep cycles may accelerate battery degradation [6, 7].

Each BEB arrival, except for the last arrival for each BEB, has a paired "route" that the BEB must perform after the visit. This route, as one would expect, causes the BEB to discharge by some certain amount. Each bus route is assumed to have a fixed discharge. Let the discharge of the route for visit i be denoted as $\Delta_i \in \mathbb{R}$. Note that the last visit for each BEB does not have an associated route, implying that there is no discharge after these particular visits, i.e., $\Delta_i = 0$ for all i corresponding to a final visit.

4.3 Optimization Problem

The objective of this work is to present a framework that optimizes the assignment of n_V BEB visits to a set of n_Q charging queues over the interval $[0, \mathcal{T}]$ provided a fleet of n_A BEBs with fixed route schedules. Particularly, the framework aims to minimize over peak power usage, energy consumption, encouraging battery health via slow charging, and maintaining the SOC of each BEB above a minimum SOC threshold.

The optimization problem outlined in this work is presented in form of an objective function with constraints. The constraints ensure that candidate solutions are operationally feasible. The variables of optimization are to be introduced in Section 4.3.1 followed by a discussion of the constraints in Section 4.3.3. The objective function is employed to allow relative comparisons between candidate solutions and is introduced in Section 4.3.2.

4.3.1 Variable Definitions

This section defines the input and decision variables used in this work. The input parameters are assumed to be fixed prior to optimizing, whereas the decision variables are

Table 4.1: Table of variables used Chapter 4.

Variable	**Description**	**Variable**	**Description**
Constants		Constants	
$\mathcal{T}$	Time horizon	n_K	Number of iterations in the repetition schedule
n_M	Total number of steps created by initial temperature, T_0, and cooling schedule	n_Q	Number of chargers
n_V	Total number of visits	n_h	Number of discrete steps in time horizon
n_B	Number of buses in use		
Input Variables		Input Variables	
Δ_i	Discharge of visit over after visit i	α_b	Initial charge percentage time for bus b
ϵ_q	Cost of using charger q	κ_b	Battery capacity for each BEB
ξ_i	The next index bus b will arrive		
a_i	Arrival time of visit i	e_i	Time bus visit i must exit the station
t_h	Discrete step in time horizon	dt	Discrete time slice in time horizon $dt_h = t_h - t_{h-1}$
r_q	Charge rate of charger q	t_m	Element of the temperature vector created by cooling equation, $t_m \in t$
ν_b	Minimum charge percentage allowed for each BEB		
Direct Decision Variables		Direct Decision Variables	
u_i	Initial charge time for visit i	d_i	Final charge time for charger for visit i
q_i	Assigned queue for visit i		
Slack Variables		Slack Variables	
η_i	Charge for the bus upon arrival visit i	s_i	Amount of time spent on charger for visit i
σ_{ij}	Binary variable determining temporal ordering of vehicles i and j	ψ_{ij}	Binary variable determining spatial ordering of vehicles i and j
p_d	Demand cost of the schedule	ϕ_i	Penalty method for visit i
$\mathbb{C}$	Set of available charging times		

the values that the SA algorithm has the freedom to manipulate. The variables definitions used in this work are summarized in .

Input Parameters

Parameters are used to indicate values that are assumed to be known prior to optimization. They will be presented in two sections: packing and discretization parameters then battery dynamic parameters. The spatiotemporal parameters are those that ensure no scheduling overlap in either space or time. The discretization parameters describe the parameters that discretize the time horizon, and the battery dynamic parameters are those associated with the SOC of the BEB.

Spatiotemporal and Discretization Parameters As previously introduced, ξ_i represents the next arrival index for bus b_i. As an example of its use, suppose the ID of each BEB is recorded in order of arrival as $\{2, 1, 3, 2\}$. Using a starting index of 1, $\xi_1 = 4$ as that is the next visit by bus 2. Each visit is prescribed arrival and departure times, a_i and e_i, respectively. An associated cost is employed when a visit is assigned to a charging queue. Let the assignment cost be represented by ϵ_q. Lastly, the time horizon is to be discretized to assist in computing the peak demand cost, let t_h denote a discrete time step, and let dt denote the discrete time step $dt = t_h - t_{h-1}$.

Battery Dynamic Parameters It is assumed that each bus begins the working day with an initial SOC percentage of α_b. Let the set of initial visits by each BEB be denoted as I_0 where $I_0 \subset I$ and the cardinality of the set is $|I_0| = n_B$. The initial SOC for bus b_i can be represented as $\eta_i = \alpha_{b_i}\kappa_{b_i}; \forall i \in I_0$ where κ_{b_i} is the battery capacity for bus b_i. After each arrival, the BEB is assigned to a charging queue. Let r_q represent the power supplied from the charger in queue $q \in Q$. Each visit, except for the final visit of each BEB, is paired with a subsequent route to be executed with a corresponding energy requirement, Δ_i. As alluded to earlier, there are no routes after the last visit for each BEB. Thus, similarly to

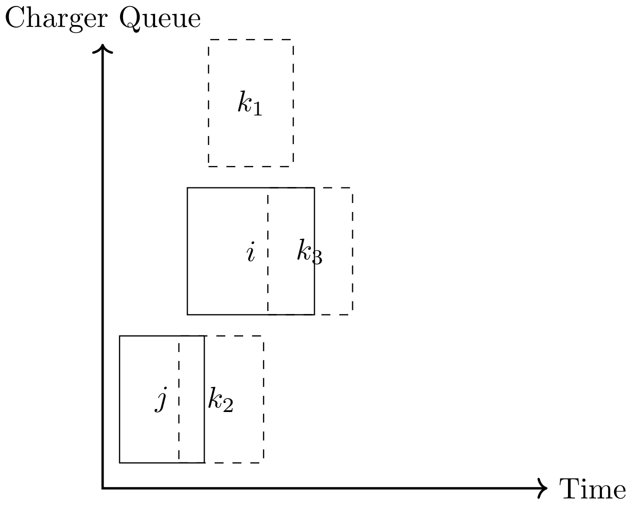


Fig. 4.1: Examples of different methods of overlapping. Space overlap: $q_{k_1} > q_i + 1 \therefore \psi_{ik_1} = 1$. Time overlap $u_{k_2} < u_j + s_j \therefore \sigma_{k_2 j} = 0$. Similarly, $\sigma_{k_3 i} = 0$.

the set of initial visits, let the set of final visits for all BEBs be denoted as I_f. The discharge for the final visit of each BEB is then defined as $\Delta_i = 0; \forall i \in I_f$.

Decision Variables

Decision variables are those chosen by the optimizer. There are three direct decision variables for each visit: the initial and final charging times, u_i and d_i, respectfully, and the selected charging queue, $q_i \in Q$.

The remaining variables are slack variables, which are introduced to track the vehicle charge and queuing position based on the problem parameters and direct decision variables. Recall the initial SOC for a visit is written as η_i, where $i \in I \setminus I_0$. Further recall the set of initial visits, I_0, have an assumed known SOC (i.e., the initial SOC of each BEB at the beginning of the working day is considered as an input parameter). The charge for bus i's next visit is equal to the initial charge for visit i plus the charge added to it by charger q_i over duration $s_i = d_i - u_i$ minus the discharge accumulated after visit i,

$$\eta_{\xi_i} = \eta_i + r_{q_i} s_i - \Delta_i. \tag{4.1}$$

The variables σ_{ij} and ψ_{ij} are used to indicate whether a visit pair (i, j) overlap the same space, as show in Figure 4.1. These spatiotemporal variables uphold the following relationships: for every visit, $\sigma_{ij} = 1 \implies$ the start charge time of visit j is greater than the end charge time of visit i. Similarly, $\psi_{ij} = 1 \implies$ the queue for visit j is of a greater index than visit i. A value of zero for either of these variables conveys no information.

The variable $\mathbb{C}$ is the set that describes the availability for all chargers. That is, $\mathbb{C}$ is a set of n_Q sets that contain available charger times for each queue $q \in Q$. Let a set of available charge times for queue q be defined as $\mathbb{C}_q$.

4.3.2 Objective Function

This work aims to minimize the total "cost" of utilizing a given charge schedule. Let $J(\mathbb{I})$ represent the objective function. The objective function for this problem has four main considerations: an assignment cost, a penalty method for visits with insufficient SOCs, consumption cost, and a demand cost. Each of which will be discussed in turn throughout the subsequent sections.

Assignment Cost

The assignment cost represents the cost of assigning a bus to a queue. Particularly, the cost consists of summing a prescribed weight for the selected charger, ϵ_{q_i}, multiplied by the charge rate, r_q. Formally, the cost is written as follows:

$$\sum_{i=1}^{n_V} \epsilon_{q_i} r_{q_i}. \tag{4.2}$$

This is effectively the cost of selecting queue q_i. While any set of weights may be selected, uses a particular choice for the assignment cost to encourage the use of slow chargers over fast for the sake of battery health. The charger queue indices are ordered such that the first n_B queues correspond to idle queues. This allows all BEBs to simultaneously sit idle if needed. All n_B idle queues have assignment costs of zero to denote that there is no cost when not charging. The next group of chargers is assumed to be the slow chargers subsequently followed by the fast. Letting $P \in \mathbb{R}$, then the set of slow and fast charging queues be of the form $[P, 2P, ..., n_Q P]$. Concatenating these vectors yields $\epsilon = [0, 0, ..., P, 2P, 3P, ...]$, where ϵ describes the vector of assignment costs and the first n_B values are zero.

Penalty Method

A penalty method is to be implemented to provide a soft constraint on the lower bound of the charge. Due to the uncertainty of the initial SOC for each visit, a soft constraint is desired to increase the solution space while penalizing non-operationally feasible solutions. If a hard constraint were to be implemented, the constraint would restrict the set of allowable schedules to only operationally feasible schedules. Let the piecewise function that enables/disables the penalty method be of the form

$$\phi(x) = \begin{cases} 0 & x \geq 0 \\ \\ x^2 & x < 0. \end{cases} \tag{4.3}$$

Letting x be defined by the difference of the initial SOC for visit i, η_i, and the minimum charge threshold, $\nu_{b_i}\kappa_{b_i}$, applies a penalty proportional to the difference of the SOC and the threshold squared. That is, $x = \eta_i - \nu_{b_i}\kappa_{b_i}$. A scalar, z_p, is added which can be utilized either as a monetary conversion or a simple gain. This method is employed as a means of encouraging that the schedule has enough charge for each BEB to complete its next route. Therefore, the penalty method is written as

$$\sum_{i=1}^{n_V} z_p \phi_i(\eta_i - \nu_{b_i}\kappa_{b_i}). \tag{4.4}$$

Consumption Cost

In most cases, utility companies have a portion of the cost related to the total electricity consumed over a billing period, referred to herein as the consumption cost. The consumption cost is the summation of all the energy being used over all the active periods for each charger in the time horizon. A scaling z_c is applied as a weight on the summation (this could correspond to a monetary cost imposed by the utility). This is represented by the summation

$$z_c \sum_{i=1}^{n_V} r_{q_i} s_i. \tag{4.5}$$

Demand Cost

Utility companies often charge a "demand cost" in an effort to reduce peak power use. A particular example of peak demand is the fifteen minute average energy usage employed by Rocky Mountain Power Schedule 8 [44].

A method of calculating the demand charge is done by calculating the average power consumption over a given period of time. Let the average power used over an arbitrary interval, T_p, be represented by

$$p_{T_p}(t) = \frac{1}{T_p} \int_{t-T_p}^{t} p(\tau) d\tau. \tag{4.6}$$

The largest average power usage over T_p is used as the demand cost for the billing period. Therefore, let the cost of the peak power consumption be dictated by the maximum average power:

$$p_{max}(t) = \max_{\tau \in [0,t]} p_{T_p}(\tau). \tag{4.7}$$

Furthermore, a fixed minimum average power is introduced that is intended to act as a base threshold before the cost begins to increase. Let this fixed threshold be defined as p_{fix}, the demand cost is calculated using

$$p_d(t) = \max(p_{fix}, p_{max}(t)). \tag{4.8}$$

For the sake of implementation, the integral in Equation 4.6 is discretized. Let dt denote the discretization time step and p_h the power for the h^{th} step, Equation 4.6 is approximated as

$$p_{T_p,h} = \frac{1}{T_p} \sum_{k=h-\frac{T_p}{dt}+1}^{h} p_k \, dt. \tag{4.9}$$

The discrete demand cost is expressed as

$$p_d = \max(p_{fix}, p_{max}). \tag{4.10}$$

Similarly to the consumption cost, a scaling z_d is applied. Again, this may be a monetary conversion or simply just a gain.

The objective function written in its entirety is

$$J(\mathbb{I}) = z_d p_d + \sum_{i=1}^{n_V} \Big[\epsilon_{q_i} r_{q_i} + z_p \phi_i (\eta_i - \nu_{b_i} \kappa_{b_i}) + z_c r_{q_i} s_i \Big]. \tag{4.11}$$

4.3.3 Constraints

While the objectives are used to compare solutions, constraints are introduced to ensure that the solutions are operationally valid. Operationally validity requires that allocated BEBs do not overlap spatially or temporally. Furthermore, the SOC of a bus at a particular visit is related to the charge from its previous visit by the amount of charging and discharging that has occurred. Finally, buses must leave the charger before their scheduled departure time. These constraints are represented as follows:

$$u_j - d_i - (\sigma_{ij} - 1)T \geq 0 \tag{4.12a}$$

$$q_j - q_i - 1 - (\psi_{ij} - 1)Q \geq 0 \tag{4.12b}$$

$$\sigma_{ij} + \sigma_{ji} \leq 1 \tag{4.12c}$$

$$\psi_{ij} + \psi_{ji} \leq 1 \tag{4.12d}$$

$$\sigma_{ij} + \sigma_{ji} + \psi_{ij} + \psi_{ji} \geq 1 \tag{4.12e}$$

$$s_i = d_i - u_i \tag{4.12f}$$

$$\eta_{\xi_i} = \eta_i + r_{q_i} s_i - \Delta_i \tag{4.12g}$$

$$\kappa_{b_i} \geq \eta_i + r_{q_i} s_i \tag{4.12h}$$

$$a_i \leq u_i \leq d_i \leq e_i \leq \mathcal{T} \tag{4.12i}$$

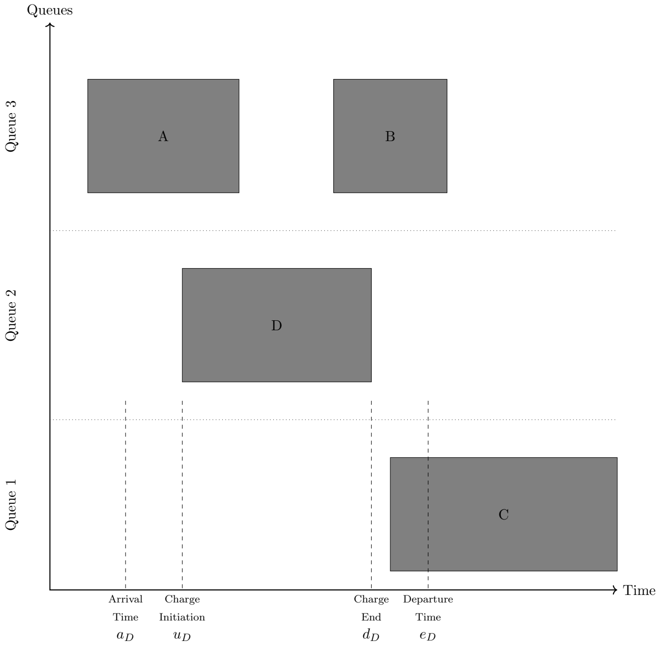


Fig. 4.2: The representation of the queue-time space. The x and y-axis represent time and space, respectively. Along the y-axis, the dashed lines represent discrete queuing locations. The shaded rectangles represent schedules BEBs to be charged. The height of each shaded rectangle represents the space taken on the queue and the width being the time to service said BEB. The vertical dashed lines are associated with vessel D and represent the arrival time, initial charge time, charge completion time, and departure time. Note that the arrival time may be before the initial charge time and the completion time may before the departure time.

Equation 4.12a-Equation 4.12e are denoted as "queuing constraints". They prevent overlap both spatially and temporally as shown in Figure 4.2. The y-axis represents the possible queues for a bus visit to be placed into, and the x-axis represents the time that can be reserved for each visit. The shaded rectangles represent time that has been scheduled in the horizontal direction, and the queue allocated for each bus visit in the vertical direction. In other words, the set of constraints Equation 4.12a - Equation 4.12e aim to ensure that these shaded rectangles never overlap.

Constraint Equation 4.12a states that the starting charge time for BEB u_j must begin after the previous BEB departs, d_i. A value of $\sigma_{ij} = 1 \implies$ bus i has detached from the charger before bus j has begun charging. If $\sigma_{ij} = 0$, then the constraint is of the form $\mathcal{T} + d_i > u_j$ rendering the constraint "inactive". Similarly, for Equation 4.12b, ψ_{ij} determines spacial positioning of BEB i and j relative to one another. A value of $\psi_{ij} = 1 \implies$ BEB i is in a queue index that is less than BEB j. If $\psi_{ij} = 0$ then the constraint is deactivated. Constraints Equation 4.12c - Equation 4.12e enforce spatial and temporal ordering between each queue/vehicle pair. Equation 4.12c and Equation 4.12d ensure that BEB i is not placed before and after j spatially or temporally as that is not possible. Equation 4.12e enforces at least one of the spatial or temporal relationships between each visit is active. This ensures there are no scheduling conflicts (i.e. either charging sessions are ordered temporally or are in different queues).

Equation 4.12f describes the service time of the bus. Equation 4.12g calculates the initial charge for the next visit for bus b_i. Equation 4.12h ensures that the bus is not being over-charged. Equation 4.12i ensures the continuity of the times (i.e. the arrival time is less than the initial charge which is less than the detach time which is less than the time the bus exits the station and all must be less than the time horizon).

4.4 Simulated Annealing

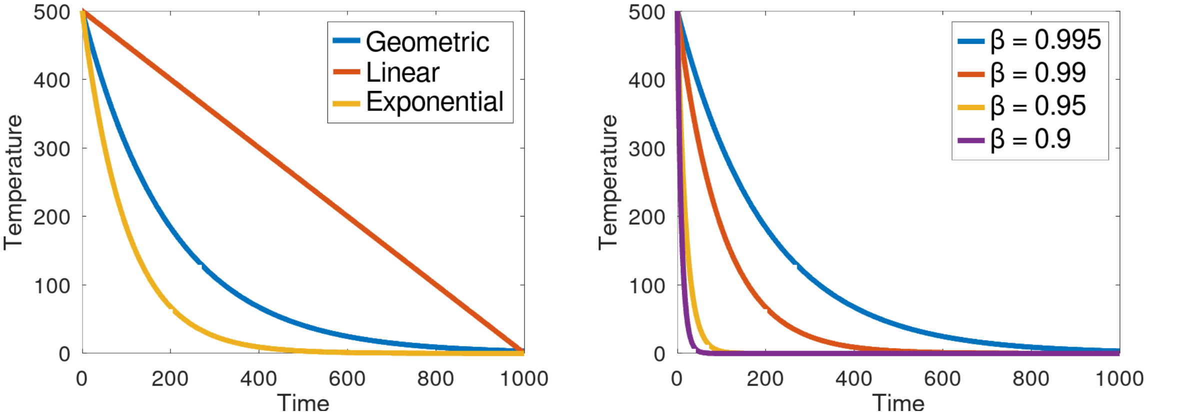
SA is a well-studied local search metaheuristic used to solve various optimization problems [45, 46]. The algorithm is often applied to problems that contain many local solutions as it employs a stochastic approach that explores the solution space for an approximate global optimum. This model is named after its analogized process where a crystalline solid is heated then allowed to cool at a slow rate until it achieves its most regular possible crystal lattice configuration (i.e. lowest energy state) [46, 47]. SA establishes a connection between the thermodynamic process and the search for global optimum in optimization problems. Within the SA process there are three key components: cooling equation, acceptance criteria, and generation mechanisms [46, 48].

The cooling equation describes the speed at which the figurative temperature is decreased in a controlled manner over time. Throughout the SA process, many "candidate" solutions are generated and compared to an "active" solution. The method by which the solutions are accepted is determined by the acceptance criteria. The acceptance criteria is a function of the system temperature that makes the decision whether the system will accept an inferior solution in favor of exploring the solution space. The means by which candidate solutions are generated is via the generation mechanisms. These generators modify the solution by some singular discrete change [45]. Each of these components are elaborated in the subsequent sections.

4.4.1 Cooling Equation

The temperature function models a "rate of cooling" for the SA process. Initially, when the temperature is high, SA encourages exploration. As the process begins to "cools down" (in accordance to the cooling schedule), it begins to encourage local exploitation of the solution (rather than exploration) [47, 49]. There are three common basic types of cooling equations: linear, geometric, and exponential. Each schedule type is depicted in Figure 4.3a [48]. Every plot begins with an initial temperature of $T_0 = 500^\circ\ C$ and a final temperature of $T_f = 1^\circ\ C$. The different cooling schedules dictate the rate at which the algorithm progressively disallows exploration. Let the vector of temperatures described by a cooling schedule be defined as t. Furthermore, let an element of the vector be denoted as $t_m \in t$, where $m \in [0, ..., M]$ and $M = |t|$.

A linear cooling schedule is defined by $t_m = t_{m-1} - \beta_0$. The terms utilized in Figure 4.3a are $t_0 = T_0$ and $\beta_0 = 1/2\ C^\circ$. An exponential cooling schedule is defined by the difference



(a) Geometric, linear, and exponential cooling schedules.

(b) Geometric cooling schedule utilizing various value of β.

equation $t_m = e^{-\beta_2} t_{m-1}$. The values utilized in Figure 4.3a are $\beta_2 = 0.01$. A geometric cooling schedules is as defined in Equation 4.13. This schedule type is most widely used in practice [48]. As such, it will also be employed by this work.

$$t_m = \beta t_{m-1} \tag{4.13}$$

The gain variable, β, in Figure 4.3a evaluated at $\beta = 0.995$. The value of β may vary anywhere between the range $[0, 1)$. The further β is from 1, the quicker the function converges to zero. Figure 4.3b demonstrates this principle by plotting the geometric schedule using varying values of β.

4.4.2 Acceptance Criteria

In SA, the algorithm stores a solution that is continuously compared to newly generated solutions. Let the stored solution be referred to as the "active solution". During each iteration, a new "candidate" solution is generated and compared to the active solution to determine if the candidate solution should replace the active solution. The method of determining whether the active solution should be replaced is defined by an acceptance criteria. In an effort to encourage exploration, inferior candidate solutions have a probability of being accepted. The probability of accepting an inferior candidate solution is determined by the objective functions of the active and candidate solutions, $J(\mathbb{I})$ and $J(\bar{\mathbb{I}})$, respectively, and the current temperature, t_m. Let $\Delta E \equiv J(\mathbb{I}) - J(\bar{\mathbb{I}})$ and let $f(\cdot)$ be the function that describes the probability of accepting a candidate solution $\bar{\mathbb{I}}$. The probability of accepting a candidate solution is thus of the form [48]

$$f(\mathbb{I}, \bar{\mathbb{I}}, t_m) = \begin{cases} 1 & \Delta E > 0 \\ e^{-\frac{\Delta E}{t_m}} & \text{otherwise} \end{cases} . \tag{4.14}$$

4.4.3 Neighbor Generators and Wrappers

Generation mechanisms are used to create a neighboring candidate solution [45]. That is, the generating function creates a solution that can be reached in a single iteration from the active solution. In response to the problem statement made in , five primitive generation mechanism are used: new visit, slide visit, new charger, wait, and new window. The purpose of each of these generators is to assign new visits to a charger, adjust a bus visits initial and final charge time within the same time frame/queue, move a BEB from one charger to another with the same charge schedule, move a bus to its idle queue. Each generator will be discussed in more detail in Section 4.4.3.

These primitive generation mechanisms will, in turn, be utilized by two wrapper functions. The charge schedule generator is to used create an initial candidate solutions for SA and the perturb schedule generator is used to take a candidate solution and alter it slightly in an attempt to step toward a global or local minimum. The wrapper functions will be discussed in Section 7. However, prior to discussing the primitives and wrapper generating functions, their respective inputs and outputs must be defined.

Generator Input/Output

Each generator primitive accepts a tuple $\mathbb{S} \equiv (i, \mathbb{I}, \mathbb{C})$ where i is the visit index being manipulated, $\mathbb{I}$ is the set of visits, and $\mathbb{C}$ is the set that describes the availability for all chargers $q \in Q$. The output of the generating functions is the same as the input, but with changes applied to it by a generator. Let a modified variable be denoted with a bar, $\bar{\cdot}$.

Thus, the modified input tuple is written as $\bar{\mathbb{S}}$.

Generators

The mechanism by which candidate solutions are generated are now introduced. Recall that to satisfy constraints, n_B extra idle queues are added that provide no power to the BEB. Because of this, the set of queues is fully defined where Q is the ordered set of idle queues, slow queues, then fast queue. The use case for the idle queues are for when a bus is not to be placed on a charger. Rather, it will be placed in the queue, $q \in B$, which satisfies the previously defined spatial constraints while allowing the bus to be "set aside". The charge queues are denoted by $q \in \{1, ..., n_B, n_B + 1, ..., n_Q\}$.

For the sake of ease in referring to the various variables associated with a visit, dot notation is used. For example, suppose the arrival time is desired to be extracted from visit i. Given $\mathbb{I}$, the notation that describes extracting the initial charge time for visit i is written as $u_i \equiv \mathbb{I}_{i.u}$.

New visit The new visit generator defined in Algorithm 0 describes the process of moving a BEB, $b \in B$, from a waiting queue, $q \in B$, to a charging queue, $q_i \in \{n_B + 1, ..., n_Q\}$, within its arrival/departure time $[a_i, e_i]$. Let $\mathcal{U}_{\{\cdot\}}$ indicate that an element is selected randomly with a uniform distribution from the set $\{\cdot\}$. For example, $\mathcal{U}_{[a_i, e_i]}$ indicates that a value will be selected between a and e with a uniform distribution. Algorithm 0 begins by extracting variables. Lines 8 and 9 randomly select a charging queue and available time frame with a uniform distribution, respectively. Line 10 attempts to assign the visit to the previously select time slice, if it succeeds, the updated visit is returned. Otherwise, the null value is returned.

The function `FindFreeTime` is the algorithm that determines whether a visit's time at the station $[a_i, e_i]$ can be placed in the time availability of charger q. Let the available time for charger q for visit i be denoted as $C \equiv \mathbb{C}_{i.q}$. Furthermore, let the lower and upper bound of $\mathbb{C}$ be denoted as C_L and C_U, respectively. The algorithm checks whether the BEB time at the station, $[a_i, e_i]$ fits within the charger availability $[C_L, C_U]$. If it does, a random charging time frame is returned such that $a_i \leq u_i \leq d_i \leq e_i$. Otherwise the null value is returned.

Algorithm 0: New visit algorithm

```
Algorithm: New Visit
Input: $\mathbb{S}$
Output: $\bar{\mathbb{S}}$
begin
    $i \leftarrow \mathbb{S}_i$;                    /* Extract visit index */
    $\mathbb{I} \leftarrow \mathbb{S}_{\mathbb{I}}$;                    /* Extract visit tuple */
    $\mathbb{C} \leftarrow \mathbb{S}_{\mathbb{C}}$;                    /* Extract visit charger availability */
    $a \leftarrow \mathbb{I}_{i.a}$;                    /* Extract the arrivial time for visit $i$ */
    $e \leftarrow \mathbb{I}_{i.e}$;                    /* Extract the departure time for visit $i$ */
    $q \leftarrow \mathbb{I}_{i.q}$;                    /* Extract the current charge queue for visit $i$ */
    $\bar{q} \leftarrow \mathcal{U}_Q$;                    /* Select a random charging queue with a uniform distribution */
    $C \leftarrow \mathcal{U}_{\mathbb{C}_{\bar{q}}}$;                    /* Select a random time slice from $\mathbb{C}_{\bar{q}}$ */
    if $(\bar{C}, \bar{u}, \bar{d}) \leftarrow$ FindFreeTime$(C, i, \bar{q}, a, e) \notin \varnothing$ then    /* If there is time available in $C$ */
        $\bar{I}_{i.q} \leftarrow \bar{q}$;                    /* Update visit tuple with new charge queue */
        $\bar{I}_{i.u} \leftarrow \bar{u}$;                    /* Update visit tuple with new inital charge time */
        $\bar{I}_{i.d} \leftarrow \bar{d}$;                    /* Update visit tuple with new final charge time */
        return $(i, \bar{I}, \bar{C})$                    /* Return visit */
    end
    return $(\varnothing)$;                    /* Return nothing */
end
```

Slide visit   This primitive generator is used for visits that have already been scheduled. Because of the constraint Equation 4.12i, there may be some slack to manipulate $[u_i, d_i]$ within the window $[a_i, e_i]$. That is, two new values, u_i and d_i are randomly selected with a uniform distribution that satisfy the constraint $a_i \leq u_i \leq d_i \leq e_i$. Algorithm 1 begins be extracting variables. Line 5 purges the visit from the charger availability schedule. The `Purge` function simply removes an assigned charge time from the set $\mathbb{C}$. Without altering selected queue, the charge time randomly re-assigned with a uniform distribution. Upon success, the updated tuple is returned, otherwise the null value is returned.

Algorithm 1: Slide Visit Algorithm

```
Algorithm: Slide Visit
Input: $\mathbb{S}$
Output: $\bar{\mathbb{S}}$
begin
    $i \leftarrow \mathbb{S}_i$;                                    /* Extract visit index */
    $\mathbb{I} \leftarrow \mathbb{S}_{\mathbb{I}}$;                /* Extract visit tuple */
    $\mathbb{C} \leftarrow \mathbb{S}_{\mathbb{C}}$;                /* Extract visit charger availability */
    $(i, \mathbb{I}, \bar{\mathbb{C}}) \leftarrow$Purge($\mathbb{S}$);   /* Purge visit $i$ from charger availibility matrix */
    $C \leftarrow \bar{C}_{i.q}$;                                   /* Get the time availability of the purged visit */
    /* If there is time available in $C$ */
    if $(\bar{C}, \bar{u}, \bar{d}) \leftarrow$ FindFreeTime($C$, $i$, $\mathbb{I}_{i.q}$, $\mathbb{I}_{i.a}, \mathbb{I}_{i.e}$) $\notin \varnothing$ then
        $\bar{\mathbb{I}}_{i.u} \leftarrow \bar{u}$;                /* Update visit tuple with new inital charge time */
        $\bar{\mathbb{I}}_{i.d} \leftarrow \bar{d}$;                /* Update visit tuple with new final charge time */
        return $(i, \bar{\mathbb{I}}, \bar{C})$                     /* Return updated visit */
    end
    return $(\varnothing)$;                                         /* Return nothing */
end
```

New charger The new charger generator moves a visit $\mathbb{I}_i$ to a new charging queue while maintaining the same charge time, $[u_i, d_i]$. Algorithm 2 begins be extracting variables, and then purges the visit from the charger availability set, a queue is selected at random with a uniform distribution, then the new selection is checked whether the charge time $[u_i, d_i]$ may be assigned to the new queue.

Algorithm 2: New Charger Algorithm

```
Algorithm: New Charger
Input: $\mathbb{S}$
Output: $\bar{\mathbb{S}}$
begin
    $i \leftarrow \mathbb{S}_i$;                                        /* Extract visit index */
    $\mathbb{I} \leftarrow \mathbb{S}_{\mathbb{I}}$;                    /* Extract visit tuple */
    $\mathbb{C} \leftarrow \mathbb{S}_{\mathbb{C}}$;                    /* Extract visit charger availability */
    $(i, \mathbb{I}, \bar{\mathbb{C}}) \leftarrow$Purge($\mathbb{S}$);  /* Purge visit $i$ from charger availibility matrix */
    $\bar{q} \leftarrow \mathcal{U}_Q$;                                 /* Select a random charging queue with a uniform distribution */
    if $(\bar{C}, \bar{u}, \bar{d}) \leftarrow$ FindFreeTime($\bar{\mathbb{C}}_{i.q}$, $i$, $\bar{q}$, $\mathbb{I}_{i.a}, \mathbb{I}_{i.e}$) $\notin \varnothing$ then  /* If there is time available in $C_q$ */
        /* Return visit, note $u$ and $d$ are the original inital/final charge times. */
        $\bar{I}_{i.q} \leftarrow \bar{q}$;                             /* Update visit tuple with new charge queue */
        return $(i, \bar{I}, \bar{\mathbb{C}})$
    end
    return $(\varnothing)$;                                             /* Return nothing */
end
```

Wait The wait generator simply removes a bus from a charger queue and places it in its idle queue, $q_i \in B$. Algorithm 3 begins by purging the visit from the charger availability set, the visit is then assigned to its idle queue for the duration of its time at the station.

Algorithm 3: Wait algorithm

```
Algorithm: Wait
Input: $\mathbb{S}$
Output: $\bar{\mathbb{S}}$
begin
    $(i, \mathbb{I}, \bar{\mathbb{C}}) \leftarrow$Purge($\mathbb{S}$);          /* Purge visit $i$ from charger availibility matrix */
    /* Update the charger availability matrix for wait queue $\bar{\mathbb{C}}_{i.q_i}$ */
    $\bar{\mathbb{C}}'_{\mathbb{I}_{i.\Gamma_i}} \leftarrow \mathbb{C}' \cup \{[\mathbb{I}_{i.a}, \mathbb{I}_{i.e}]\}$;
    $\bar{\mathbb{I}}_{i.q} \leftarrow \mathbb{I}_{i.b}$;          /* Reassign bus to idle queue */
    $\bar{\mathbb{I}}_{i.u} \leftarrow \mathbb{I}_{i.a}$;          /* Set initial charge time to the arrival time */
    $\bar{\mathbb{I}}_{i.d} \leftarrow \mathbb{I}_{i.e}$;          /* Set the final charge time to the departure time */
    return $(i, \bar{\mathbb{I}}, \bar{\mathbb{C}})$          /* Return visit */
end
```

New Window New window, as shown in Algorithm 4, is a combination of Algorithm 0 (new visit) and Algorithm 3 (wait). By this it is meant that visit i is placed in its wait queue then added back in as if it were a new visit. This implies that the BEB may be assigned to a different queue with a new charging time frame. Upon success, the algorithm returns the updated tuple, otherwise return the null value.

Algorithm 4: New window algorithm

```
Algorithm: New Window
Input: $\mathbb{S}$
Output: $\bar{\mathbb{S}}$
begin
    $\bar{\mathbb{S}} \leftarrow$Wait($\mathbb{S}$);          /* Assign visit to its respective idle queue */
    if $\bar{\bar{\mathbb{S}}} \leftarrow$ NewVisit($\bar{\mathbb{S}}$) $\notin \varnothing$ then          /* Add visit $i$ back in randomly */
        return $\bar{\bar{\mathbb{S}}}$          /* Return visit */
    end
    return $(\varnothing)$;          /* Return nothing */
end
```

Generator Wrappers

The generator wrappers provide the highest level of abstraction from which the SA algorithm directly interacts. These wrapper functions utilize the primitive generators previously described to either create a new charge schedule to initialize the SA algorithm, or to modify an existing schedule.

Charge Schedule Generation   The objective of Algorithm 5 is to introduce a method that provides the SA algorithm with an initial charging schedule. The schedule generation is chosen to initialize the algorithm in a greedy manner by looping through each visit and executing Algorithm 0 to place visit i at random queue with a random charge time.

Algorithm 5: Charge schedule generation algorithm

```
Algorithm: Candidate Solution Generator
Input: $(\mathbb{I}, \mathbb{C})$
Output: $(\bar{\mathbb{I}}, \bar{\mathbb{C}})$
begin
    /* Select an unscheduled BEB visit from a randomly indexed set of visits      */
    foreach $\mathbb{I}_i \in \mathbb{I}$ do
        $(i, \bar{\mathbb{I}}, \bar{\mathbb{C}}) \leftarrow$ NewVisit($\mathbb{I}_i$, $\mathbb{I}$, $\mathbb{C}$);      /* Assign the bus to a charger */
    end
    return $(\bar{\mathbb{I}}, \bar{\mathbb{C}})$
end
```

Perturb Schedule   Algorithm 6 describes the method by which the SA algorithm decides how to perturb a given charge schedule. The method that will be employed to generate neighboring solutions is as follows: pick a visit, pick a primitive generator, and execute said primitive generator once. Let $\mathcal{W}^{y}_{[\cdot]}$ denote a random selection with a distribution specified by a weight vector $y \in \mathbb{R}$. Lines 2-12 of Algorithm 6 generate a vector of weights for the visit index selection. The weights have a default value of one. Each visit is then indexed in reverse order. If the SOC of the visit is less than $\nu_b \kappa_b$, then the weight for the visit is calculated as shown on Line 10. The route for BEB b is then set as a "priority" on Line 9 to propagate the previously calculated weight to earlier visits of BEB b as shown on Line 5. This is done in an attempt to encourage the SA algorithm to "fix" the current or previous visits so that the SOC stays above the minimum threshold. The algorithm then selects a visit index with weighted distribution y^v and selects a primitive with a weighted distribution, y^p. Letting n_G denote the number of primitive generating functions, line 15 selects a primitive generating function with a weighted distribution, $\mathcal{W}^{y^v}_{[1,n_G]}$. The primitive is then executed, and the results are returned.

Algorithm 6: Perturb schedule algorithm

```
Algorithm: Perturb Schedule
Input: $(\mathbb{I}, \mathbb{C})$
Output: $(\bar{\mathbb{I}}, \bar{\mathbb{C}})$
begin
    $p \leftarrow [false; n_A]$;          /* Create vector of booleans to track priority status */
    $y^v \leftarrow [1.0; n_V]$;                 /* Create weight vector for index selection */
    /* Loop through the visits in reverse order                                   */
    for $i \leftarrow |\mathbb{I}|$ TO 1 do
        /* Check whether the current visit is part of a priority route            */
        if $p_{\mathbb{I}_{i.b}} = true$ then
            $y^v_{\mathbb{I}_i} = y^v_{\mathbb{I}_{i.\xi}}$;       /* Propagate the priority level to previous visit */
        end
        /* Prioritize if the current visit's SOC falls below the allowed threshold
           */
        else if $\mathbb{I}_{i.\eta} \leq \nu_{\mathbb{I}_{i.b}} \kappa_{\mathbb{I}_{i.b}}$ then
            $p_{\mathbb{I}_{i.b}} = true$;  /* Indicate the current BEB's routes are to be prioritized */
            $y^v_{\mathbb{I}_i} = \kappa_{\mathbb{I}_{i.b}} + \kappa_{\mathbb{I}_{i.b}}(\nu_{\mathbb{I}_{i.b}} \kappa_{\mathbb{I}_{i.b}} - \mathbb{I}_{i.\eta})$;  /* Calculate the weight of the current visit
              */
        end
    end
    $i \leftarrow \mathcal{W}^{y^v}_{\mathbb{I}}$;                  /* Select an index with a weighted distribution */
    $y^p \leftarrow [y^p_1, y^p_2, ...]$;              /* Define the weight of each primitive generator */
    /* Select a primitve generating function with weighted distribution          */
    PrimitiveGeneratingFunction $\leftarrow \mathcal{W}^{y^p}_{[1,n_G]}$;
    $(i, \bar{\mathbb{I}}, \bar{\mathbb{C}}) \leftarrow$ PrimitiveGeneratingFunction($(i, \mathbb{I}, \mathbb{C})$);  /* Excecute the generator function */
    return $(\bar{\mathbb{I}}, \bar{\mathbb{C}})$
end
```

4.4.4 Alternative Heuristic Implementation

As suggested by the works in [50, 51], applying heuristics to the generating functions can manipulate the searched neighborhoods in a way that may assist the SA algorithm with convergence. As a test to assist in minimizing charger utilization, a simple heuristic is applied to Algorithm 0 and Algorithm 2 in the method that they select new charging queues. Rather than selecting a queue at random from $q \in Q$, the algorithms randomly select whether to place a BEB in a slow or fast charging queue with a weighted distribution favoring slow chargers. Once the charger type has been selected, the algorithm will then begin incrementally attempting to place the BEB in a queue of that type beginning from the smallest index of that charger type. For example, if a BEB has been selected to be placed in a queue with a slow charger, the algorithm begins by attempting to place the BEB in the charger queue $q = n_B + 1$. If it is unable to be placed in that queue, it then attempts to be placed in the next queue $q = n_B + 2$. This is done incrementally until all the queues have been exhausted. The objective of this alternative approach is to explore whether the added up-front computation cost by including the heuristic will positively influence the output of the results and to what degree.

4.5 Optimization Algorithm

This section combines the generation algorithms and the optimization problem into a single algorithm (Algorithm 7). Generally, SA assumes that the generated candidate solutions are within the solution space of the problem, $\mathbb{S} \in S$ where S is the solution space. In other words, the initialization and perturbations of a schedule must be verified to ensure that the generated schedule is in the solution space. Therefore, the objective function and constraints introduced in Section 4.3.3 and Section 4.3.2, respectively, must be employed to verify that the outputs of Algorithm 5 and Algorithm 6 are in the feasible space, S.

As previously stated, the generating functions directly influence the values of the assigned charge queue, charge initialization time, and charge completion time: q_i, u_i, and d_i, respectively. Having generated those values, the rest of the decision variables may be derived. Beginning with the packing constraints, Equation 4.12a-Equation 4.12b are employed to enable and disable σ_{ij} and ψ_{ij} and Equation 4.12c-Equation 4.12e ensure the validity of the values. Equation 4.12f can be directly calculated and Equation 4.12i is fully defined.

Changing the focus over to the dynamic constraints, similarly to what was seen with the packing constraints, the battery dynamic constraints are also fully defined. Equation 4.12g is sequentially calculated after a given schedule has been created. Equation 4.12h is evaluated to ensure the BEB is not overcharged. The penalty method implemented in Section 4.3.2 is set in place to allow the SOC to go below the specified threshold, $\nu_{b_i}\kappa_{b_i}$, but punish the solution for doing so. Thus, over time, the candidate solutions will be encouraged toward a solution that does not activate the penalty method (i.e., is solution is operationally feasible).

The implementation of the SA PAP, outlined in Algorithm 7, will now be discussed. The algorithm begins by creating a temperature schedule and creating an initial solution. The algorithm then iterates through the temperature schedule (outer loop). For each iteration of the outer loop, an inner loop is executed n_K times. During this inner loop, the solution is modified by a primitive generating function to create a candidate solution. The candidate is solution is then compared with the active solution, and updated according to the acceptance criteria. These actions are performed until the cooling equation is exhausted.

Algorithm 7: Simulated annealing approach to the position allocation problem

```
Algorithm: SA PAP
Input: ($\mathbb{I}$ , $\mathbb{C}$)
Output: ($\bar{\mathbb{I}}$, $\bar{\mathbb{C}}$)
begin
    /* Generate vector of temperatures given cooling equation $T$ and initial
       temperature $T_0$                                                          */
    $t \leftarrow T(T_0)$
    $\mathbb{S} \leftarrow$ChargeScheduleGenerator(($\mathbb{I}$, $\mathbb{C}$))/* Generate an initial solution   */
    /* For each item in the temperature vector                                   */
    foreach $t_k \in t$ do
        /* For each step in the constant temperature repitition counter          */
        foreach $k \in \{0, 1, ..., n_K\}$ do
            $\bar{\mathbb{S}} \leftarrow$ PerturbSchedule(($\mathbb{I}$, $\mathbb{C}$)) ;        /* Generate a new solution */
            $\Delta E =$ J($\bar{\mathbb{S}}_{\mathbb{I}}$) - J($\mathbb{S}_{\mathbb{I}}$) ;    /* Calculate the difference of fitness scores */
            if $\bar{\mathbb{I}} \in S$ and $\Delta E < 0$ then
                $\mathbb{S} \leftarrow \bar{\mathbb{S}}$
            end
            if $\bar{\mathbb{I}} \in S$ and $\Delta E \geq 0$ then
                $\mathbb{S} \leftarrow \bar{\mathbb{S}}$ with probability $e^{\frac{\Delta E}{t_k}}$
            end
        end
    end
    return ($\mathbb{I}$ , $\bar{\mathbb{C}}$)
end
```

4.6 Example

An example is now provided to demonstrate the utility of the developed SA charge scheduling technique. In Section 4.6.1 a description of the example scenario is presented followed by a brief introduction of the BEB implementation of the PAP (BPAP) and an alternative threshold based strategy called the Qin-Modified technique. Section 4.6.2 presents the results for each of planning strategies. The results are then analyzed and discussed.

4.6.1 BEB Scenario

The test scenario was run over a time horizon of $T = 24$ hours, with a total of $n_V = 338$ visits to the station shared between $n_B = 35$ buses. Each BEB is assumed to have a battery capacity of $\kappa_b = 388$ kWh that is required to stay above an SOC of $\nu_b = 25\%$ (97 kWh). Each bus is assumed to begin the working day with $\alpha = 90\%$ charge (349.2 kWh). A total of 30 chargers are utilized where 15 of the chargers are slow charging (30 kW) and 15 are fast charging (911 kW). The gains of $z_p = 5{,}000$, $z_c = 1$, and $z_d = 10{,}000$ are used. As previously introduced, to encourage slow charging for battery health, the values of ϵ in the objective function are $\forall q \in \{1, 2, ..., n_B\}; \epsilon_q = 0$ and $\forall q \in \{n_B + 1, n_B + 2, ..., n_Q\}; \epsilon_q = 10q$. The SA algorithm utilizes the geometric cooling schedule with an initial temperature of $T_0 = 9{,}000$ with $\beta = 0.997$, resulting in a total of $n_M = 9{,}101$ steps. Rocky Mountain Power utilizes fifteen-minute intervals to calculate the demand cost [44]. To match the method by which Rocky Mountain Power determines its demand cost, this work employed an interval of $T_p =$ 900 seconds in its demand cost calculation. A weight vector of $[0.3333, 0.3333, 0.1667, 0.1667]$ is used to influence the distribution of selecting the new charger, new window, wait, and slide visit primitives, respectively. The algorithm also assumes a total of $n_K = 500$ iterations for the local search at a constant temperature. In total, that results in 4,550,500 configurations being searched in a total runtime of 2,275.25 seconds.

introduced the idea of an alternative heuristic implementation for the SA algorithm. To distinguish the heuristic implementation from the method derived in , let this implementation be referred to as "heuristic" implementation and the previous as the "quick" implementation due to the fact that it is designed to execute more quickly. Using the same weights for randomly selecting the primitive generators, the heuristic approach further implemented a weighted distribution vector of $[0.75, 0.25]$ to decide whether to select a slow or fast charger, respectively. The heuristic approach had a total runtime of 3,640.4 seconds. The heuristic generators were expected to be slightly slower due to its iterative approach.

The Qin-Modified is a threshold-based strategy that is also employed as a means of comparison with the results of the SA BPAP. The Qin-Modified algorithm is a based on the threshold strategy of [34]. The algorithm has been modified slightly to accommodate the case of multiple charger types without a heuristic search for the best charger type. The heuristic is based on a set of rules that revolve around the initial charge of the bus at visit i. There are three different thresholds, low (60%), medium (70%), and high (90%). Buses below the low threshold are prioritized to fast chargers then are allowed to utilize slow chargers if no fast chargers are available. Buses between the low and medium threshold prioritize slow chargers first and utilize fast chargers only if no slow chargers are available. Buses above the medium threshold and below high will only be assigned to slow chargers. Buses above the high threshold will not be charged. Once a bus has been assigned to a charger, it remains on the charger for the duration of the time it is at the station, or it reaches 90% charge, whichever comes first.

Another method utilized to compare with against SA PAP is the BEB implementation of the PAP [52]. The BPAP implementation is utilized in this work as a benchmark for the other schedules as it is implemented utilizing a commercial solver. The inputs to the system are the same as those discussed above. It is of note that the BPAP does not implement the demand cost in its objective function. In an attempt to compare the solution of the BPAP with the SA output more directly, a similar solve time of 1,900 seconds is utilized. The BPAP was executed using the Gurobi MILP solver [43]. The previously described simulations were run on a machine equipped with an AMD Ryzen 9 5900X 12 - Processor (24 core) at 4.95GHz.

4.6.2 Results

The schedules generated by each of the methods is presented in Figure 4.4. For the sake of conciseness of the schedule plots, the waiting queues are excluded. Therefore, rows 0-14 represent slow charging queues and rows 15-29 represent fast charging queues. The hollow circles with an 'X' represent the initial charge times, and the horizontal line with the vertical tick signifies the region of time the charger is active. The Qin-Modified schedule utilized two fast chargers and fourteen slow chargers as can be seen in Figure 4.4a. The BPAP framework generated a schedule that utilizes three fast charges and four slow chargers as shown in Figure 4.4b. The heuristic SA strategy created a schedule with nine slow charger queues and one fast charging queue as shown in Figure 4.4c. The quick strategy for the SA algorithm created a schedule utilizing seven slow chargers and two fast chargers as is demonstrated in Figure 4.4d. That is to say, while each schedule emphasized the utilization of slow chargers, the Qin-Modified required fast charging most frequently followed by the BPAP, quick SA, and then heuristic SA. At the expense of incorporating more slow chargers than the BPAP, the SA techniques chose to utilize fast chargers less frequently in their respective schedules showing an emphasis on battery health.

Table 4.2 tabulates the mean, minimum, and maximum SOC upon arrival for each visit. The BPAP requires each BEB to stay above an SOC of 25% while the quick and heuristic SA approaches heavily penalize a schedule for allowing a BEB to go below the 25% SOC threshold. The BPAP was able to successfully keep the SOC above the threshold while both SA approaches were a few kWh below the threshold. The SOC of the quick SA approach dropped to a minimum of 94.760 kWh and the heuristic had a minimum SOC of 91.265 kWh, as shown in Table 4.2. Due to the threshold constraint being soft, the SA objective function may find it better to allow a small deficit in the threshold penalty function in favor of another action. As a remedy to ensure the SA schedules stays above the threshold, a safety factor could be introduced to artificially increase the threshold, $S_f \nu_{b_i} \kappa_{b_i}$ where $S_f > 1$; $S_f \in \mathbb{R}$.

The Qin-Modified schedule allowed the SOC for one of the BEBs to reach 0% as shown in Table 4.2. The Qin-Modified strategy, being a purely reactive model, does not have foresight to determine whether a set of routes has a particularly taxing route later in the time horizon. As such, and in the case of the example scenario, the BEB that reached a charge of 0% began with a sequence of short routes, much like the other BEBs. However, rather than continuing this trend, these sets of routes had one or two longer routes which the Qin-Modified algorithm was unable to account for. Interestingly, despite having a bus drop to zero charge, the Qin-Modified strategy had the highest mean SOC, followed by the quick SA, heuristic SA, and then the BPAP.

Table 4.2: Table of mean, min, and max SOC (kWh) for each charging schedule.

	BPAP	Qin-Modifid	Heuristic	Quick
Mean	181.327	248.864	182.004	188.327
Min	97.000	0.000	91.265	94.760
Max	382.930	349.200	387.829	388.000

Figure 4.7 depicts the power utilized over the time horizon for each model. Referencing Figure 4.7a, the Qin maintained long periods of steady slow and fast charger use. This is again a symptom of the Qin-Modified strategy placing BEBs on chargers based solely on the SOC upon arrival. The BPAP and SA techniques, having demand peaks in the first half of the time horizon, were able to effectively maintain lower demand profiles during slower moments throughout the day (the SA techniques more so than the BPAP). Figure 4.7 is also of interest as it shows the peak power demand over the time horizon. The peaks for each schedule shown in Table 4.3. Both the quick and heuristic SA techniques were able to maintain peak power use below 1,130 kW whereas the BPAP and Qin had peaks above 1,900 kW demonstrating significant demand cost reduction.

Table 4.3: Table of mean and max power demand for each charging schedule.

	BPAP	Qin-Modified	Heuristic	Quick
Mean	176.550	394.130	180.858	186.858
Max	1,910.000	2,000.000	1,150.950	1,120.950

The total energy consumed by each schedule is shown in Figure 4.8. The ordering of most energy consumed to least is as follows: Qin-Modified, quick SA, heuristic SA, and the BPAP. The respective energy consumption for each technique is: 9,459.120 kWh, 4,428.670 kWh, 4,295.660 kWh, and 4,237.200 kWh with the heuristic SA consuming about

58.5 kWh more than the BPAP. While the quick and heuristic SA techniques were slightly above the BPAP in energy consumption, it is expected that the BPAP would have the lowest consumed energy as it only considers consumption cost. Despite considering peak demand, the SA methods had nearly the same consumption as the BPAP. Referencing Table 4.2 and Table 4.3 for the mean SOC and mean demand, respectively, the descending order of consumed energy is correlated to the descending order of the mean SOC and the descending order of the mean power demand. This makes sense as a higher mean SOC implies the chargers being active more often; similarly for the mean demand.

Table 4.4: Table of objective function scores for each of the schedules.

Schedule	Score
BPAP	18,500,000
Qin-Modified	34,578,526
Heuristic	11,673,937
Quick	11,234,577

As a final comparison, the scores for the Qin-Modified, quick SA, BPAP, and heuristic SA are shown in Table 4.4. The Qin-Modified strategy naturally has the highest score as it performed the worst in each metric of the objective function. Although the BPAP was able to maintain the SOC of each BEB above the minimum charge threshold, due to large peaks in the power demand in the BPAP schedule, both SA techniques were able to achieve lower scores. In other words, although the SA techniques allowed small breaches in the minimum SOC, the objective function found the quick and heuristic SA schedule configurations to be more desirable than that of the BPAP. The quick SA was able to successfully obtain the lowest score due to its substantial reduction in the demand cost and its smaller breach of the minimum SOC threshold.

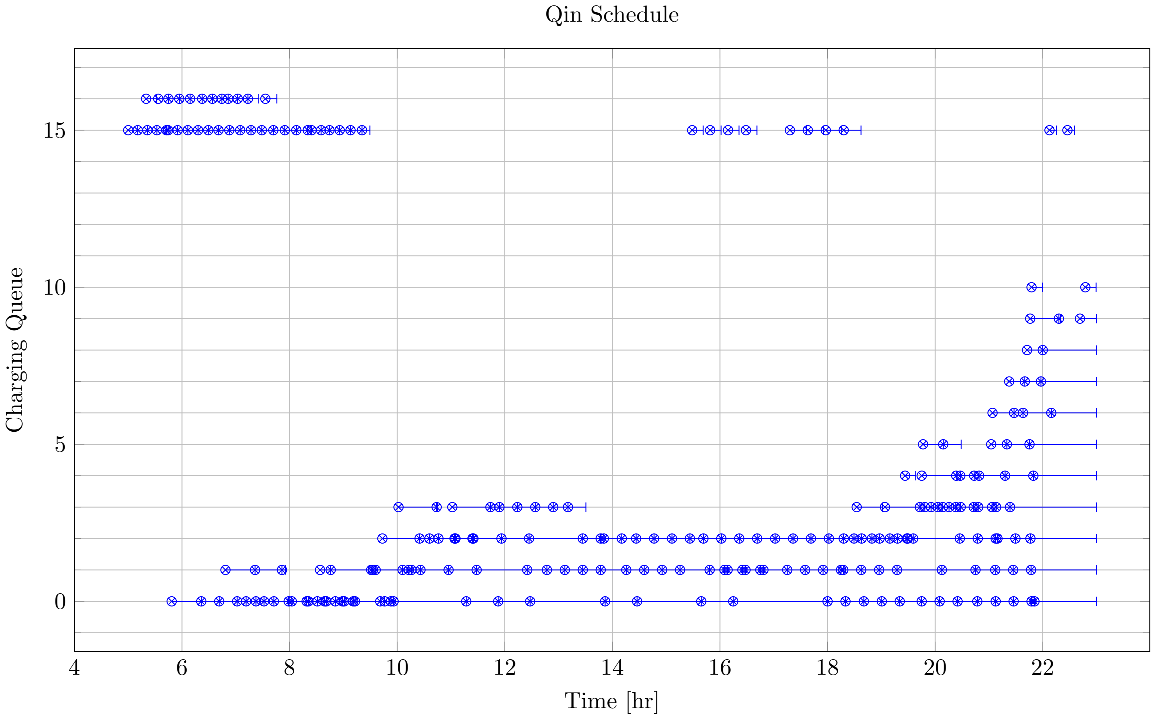


Figure 4.4(a)

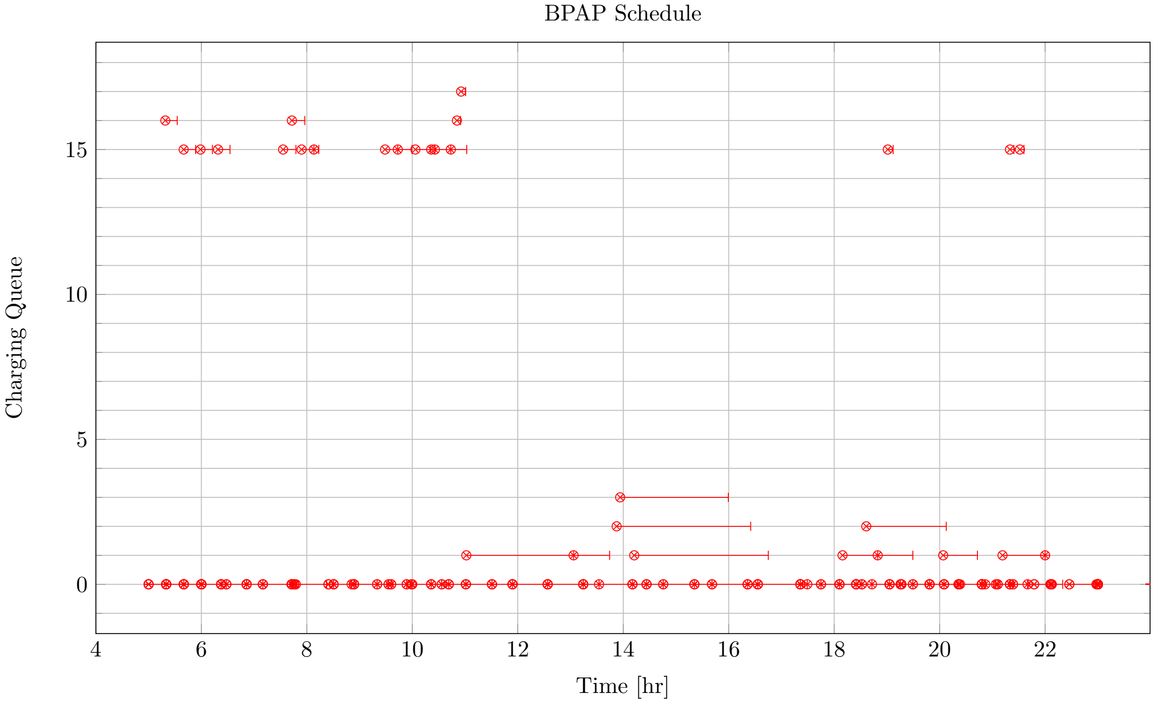


Figure 4.4(b)

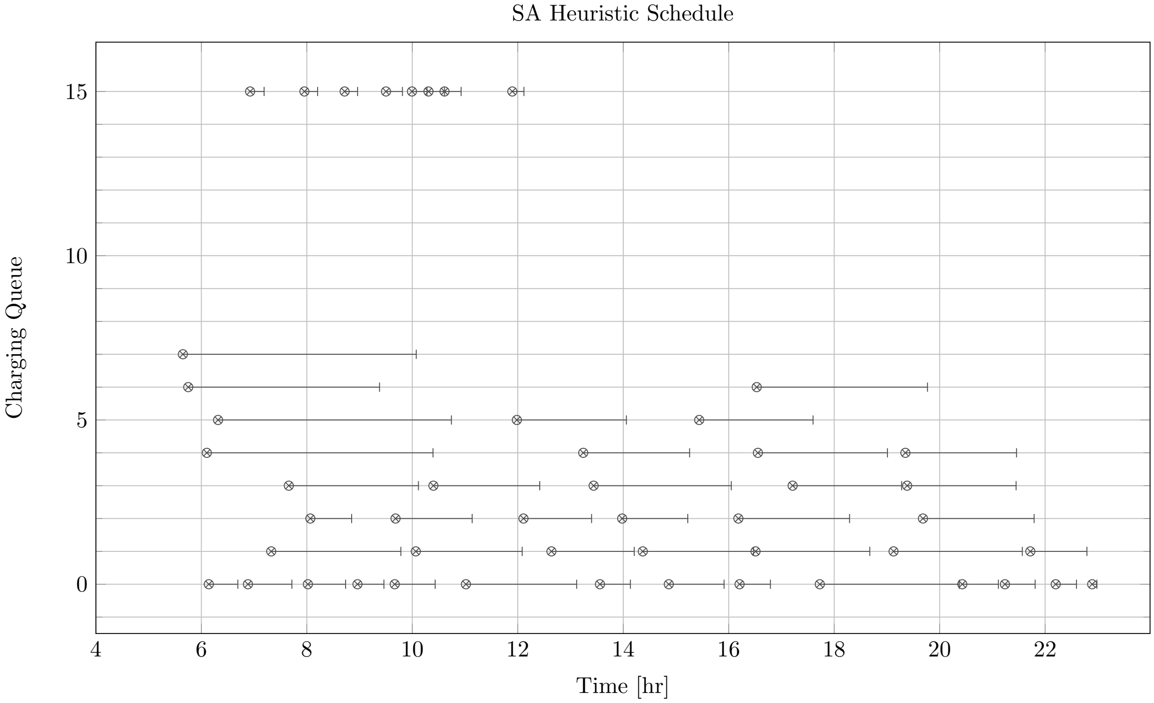


Figure 4.4(c)

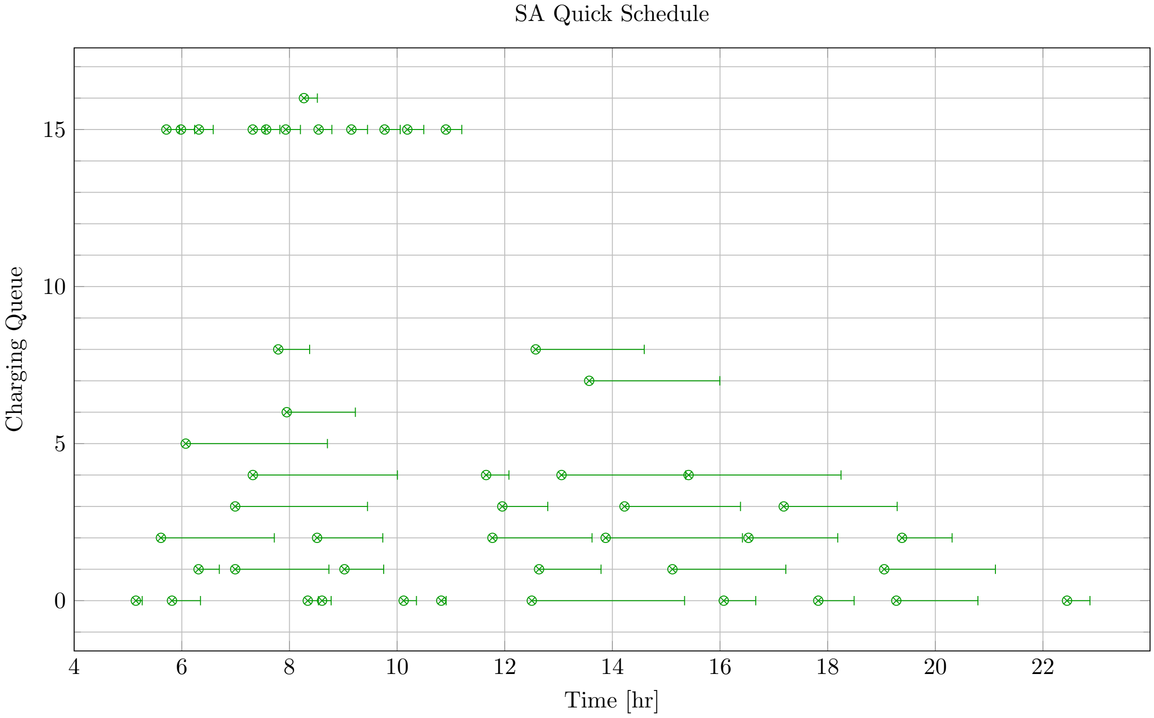


Figure 4.4(d)

Fig. 4.4: Various schedules generated by the different frameworks. Figure 4.4a is the Qin-Modified schedule, Figure 4.4b is the BPAP schedule, Figure 4.4c is the heuristic SA schedule, and Figure 4.4d is the quick SA schedule. The horizonontal line stemming from the nodes ending with a vertical tick indicate the charge duration for that particular visit.

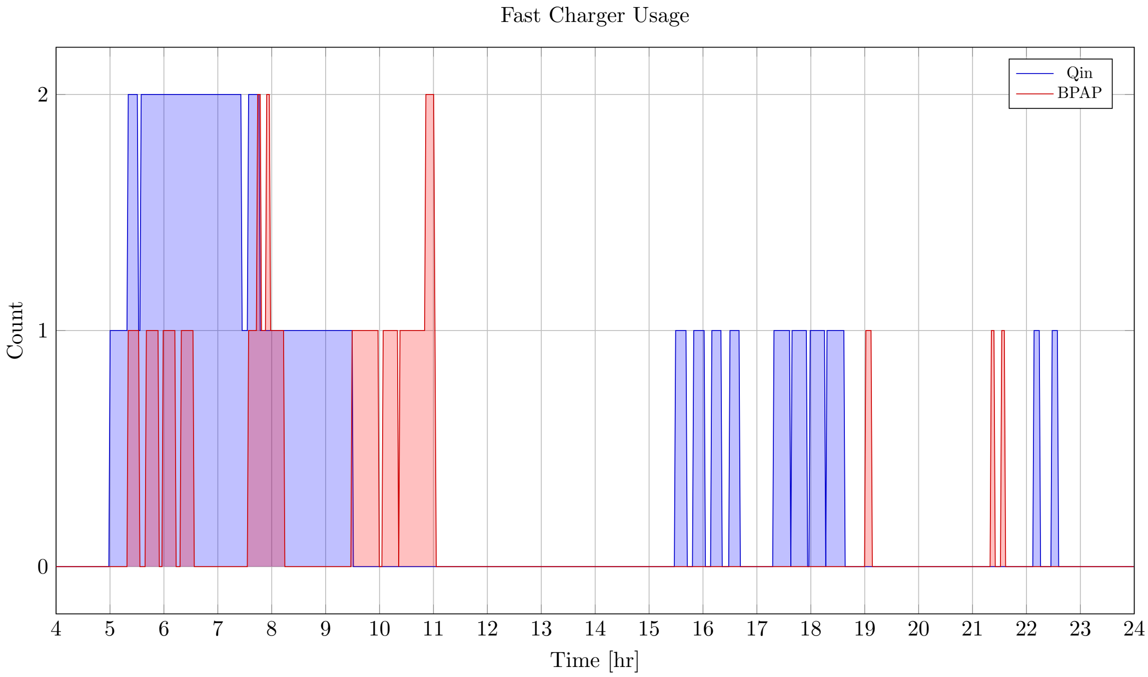


Figure 4.5(a)

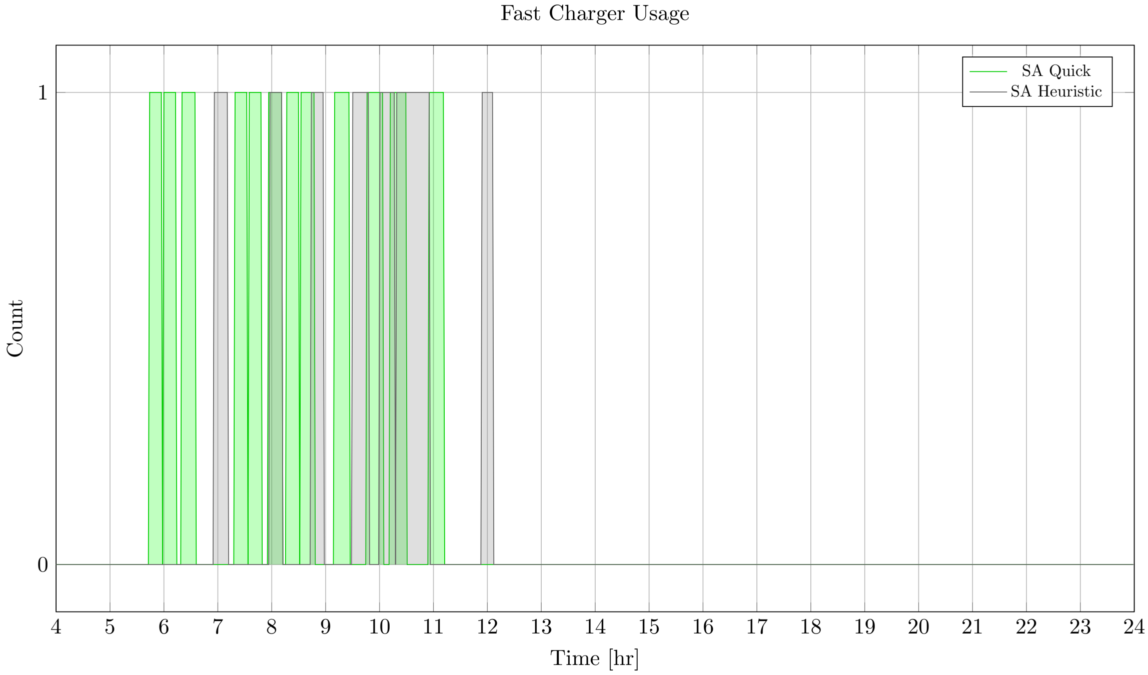


Figure 4.5(b)

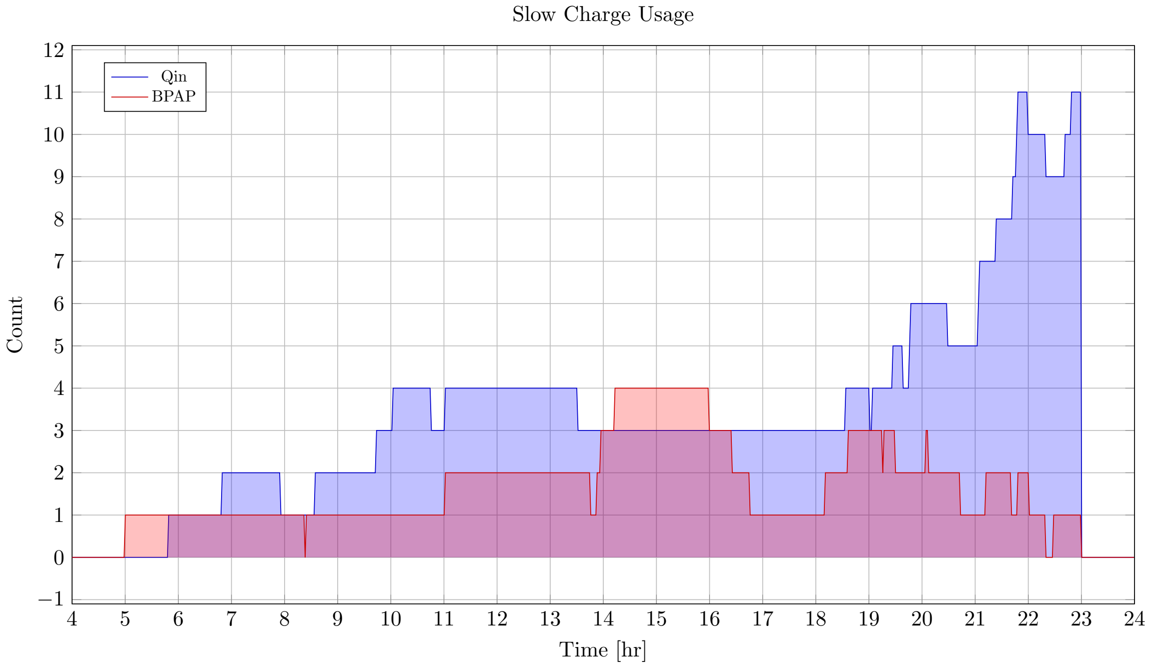


Figure 4.6(a)

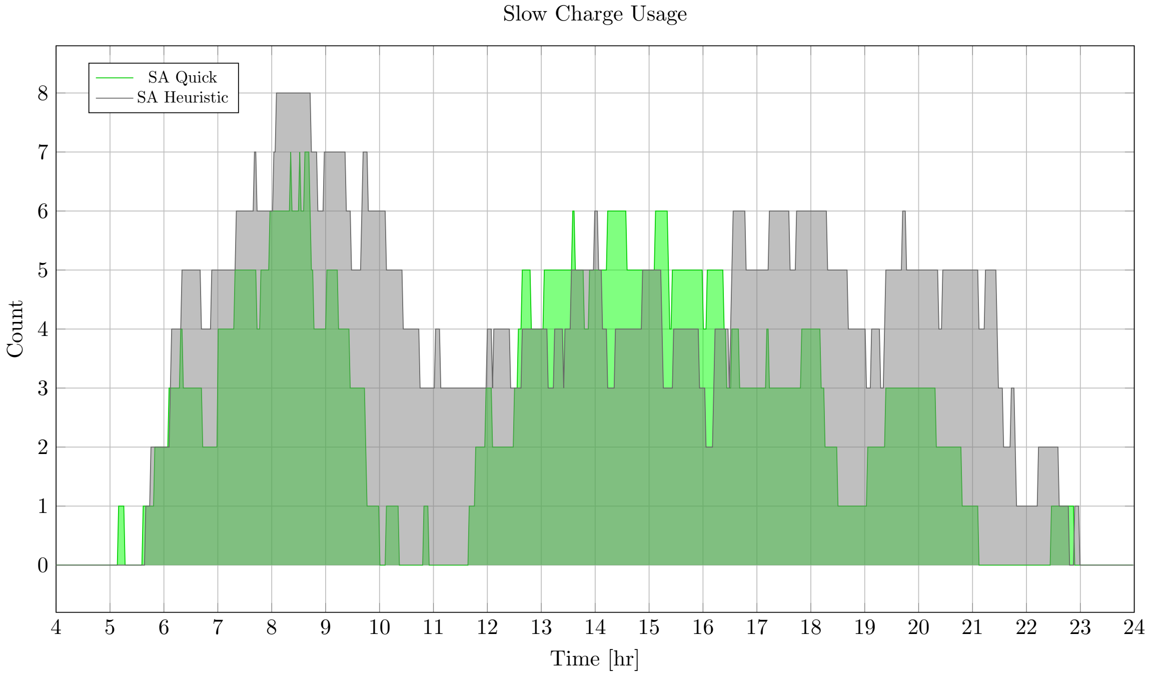


Figure 4.6(b)

Fig. 4.6: Number of fast chargers utilized in parallel over the time horizon. Figure 4.5a plots the fast charger count for the BPAP and Qin schedules and Figure 4.5b plots the fast charger count for the quick and heuristic SA schedules.

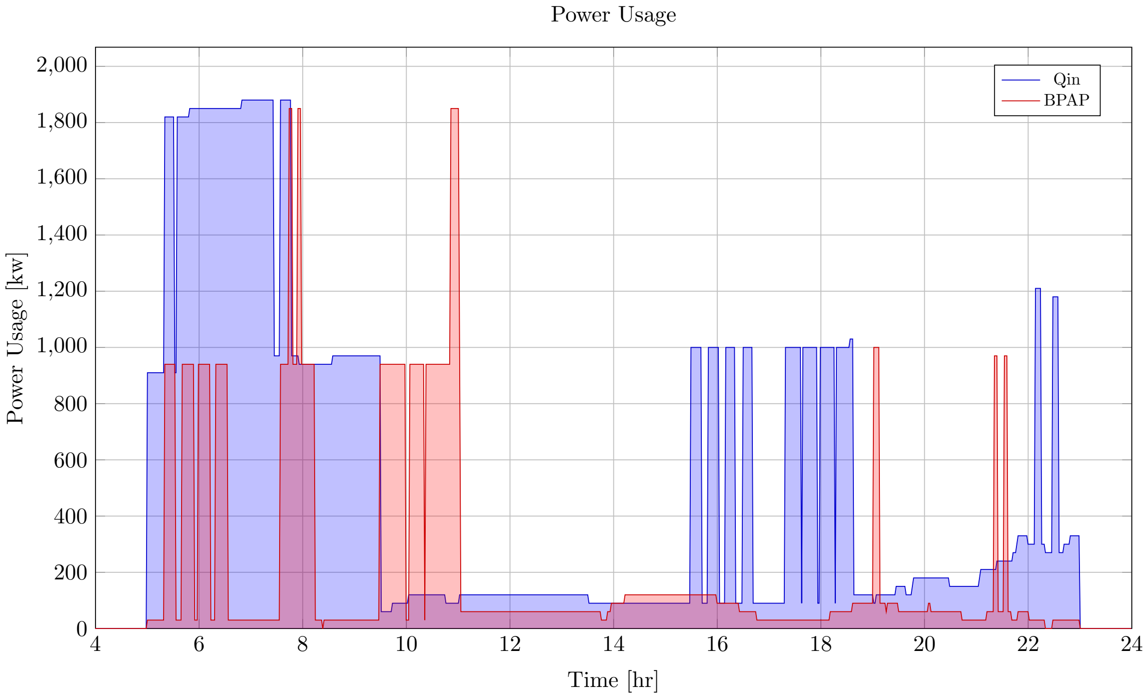


Figure 4.7(a)

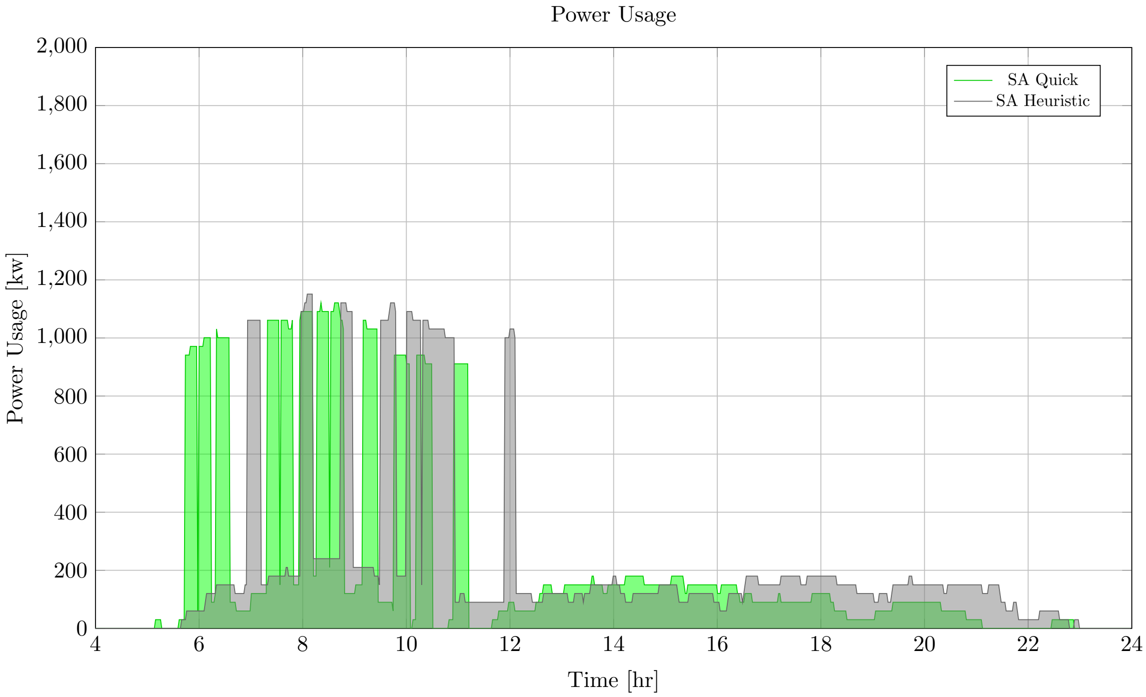


Figure 4.7(b)

Fig. 4.7: Power demand for each schedule over the time horizon. Figure 4.7a plots the power demand for the Qin and BPAP schedules and Figure 4.7b plots the power demand for the quick and heuristic SA schedules.

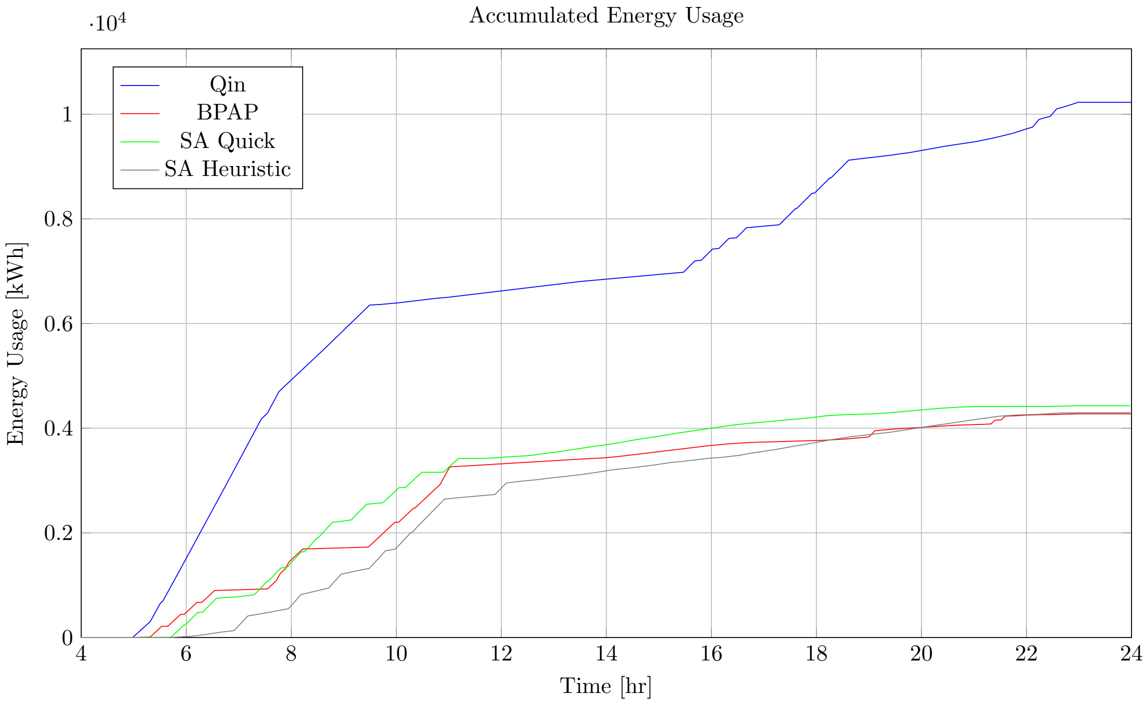


Fig. 4.8: Total accumulated energy consumed by the Qin-Modified, MILP, quick and heuristic SA schedules throughout the time horizon.

CHAPTER 5
A SIMULATED ANNEALING APPROACH WITH NON-LINEAR BATTERY DYNAMICS

The models presented up to this point have employed linear battery dynamic models to estimate the SOC of the BEB during its charging phase. While linear battery dynamics are accurate up to about an 80% SOC [36], fidelity is lost after this point due to the non-linearity of the charge profile. This chapter introduces a method of replacing the linear battery dynamics throughout this work with a non-linear dynamics model. The non-linear model will be employed in the SA algorithm from Chapter 4. The chapter proceeds as follows: an introduction to the non-linear battery dynamics model is shown in Section 5.1 along with a proof and description of how the model is incorporated. Section 5.2 presents and discusses the results.

5.1 Non-linear Battery Dynamics Model

Modeling the charging dynamics is imperative to the model's accuracy as it is one of the main factors in terms of the decision variables. If the SOC is improperly modeled, that will produce an erroneous depiction of the state of BEB charges and could result in over or under charging. Thus, care must be taken into considering the BEB's charging model. There are various methods of modeling the SOC of a battery and can vary in complexity based on the attempt to incorporate temperature, battery degradation, and current [9, 28, 53].

Some of the conventional methods to charge batteries are: Constant Voltage (CV), Constant Current (CC), and Constant Current Constant Voltage (CCCV) [54]. In CCCV, a constant current is applied to a battery until it reaches terminal voltage. Once this point has been reached, a constant voltage is applied as the charge current decreases and the battery reaches full charge [9]. Thus, by extension, CV merely applies a constant voltage and CC a constant current [54].

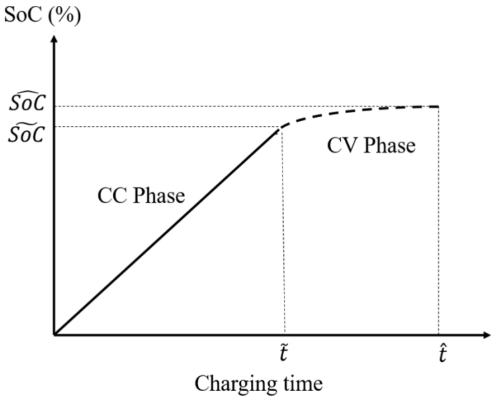


Fig. 5.1: Illustration of non-linear charging profile.

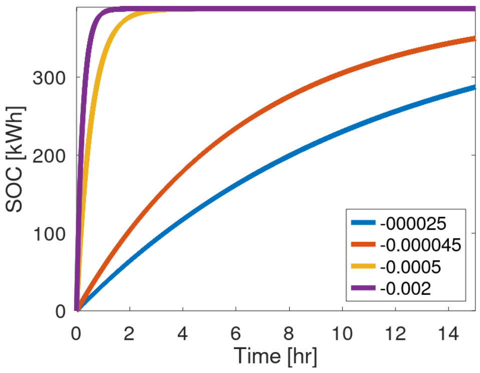


Fig. 5.2: Charging profiles for various convergence rates.

As previously stated, the SOC can be accurately modeled until the battery reaches a charge of about 80% [36]. At this point the SOC becomes non-linear. Naturally, it has been suggested by [28] that the SOC can be broken down into a linear and non-linear component. A plot of the SOC for a battery is shown in to demonstrate these components [28].

While some leverage the linear and non-linear components to derive their SOC model [10], others have derived a first order equation to model this behavior [22]. Assume that a charge will occur over dt seconds. The SOC on the time step $h+1$ for bus i can be determined by the simple discrete first order equation

$$\eta_{\xi_i} = \bar{a}_q \eta_i - \bar{b}_q \kappa_{\Xi_i}, \tag{5.1}$$

where

$$\bar{a}_q = e^{a_q dt} \quad \bar{b}_q = e^{a_q dt} - 1. \tag{5.2}$$

The equation is developed by using exact discretization [55], and is proved in [22]. The lemma and proof are subsequently shown for completeness.

Lemma. *Assume that the charge will occur over intervals over Δ seconds, the charge at time step $k+1$ for visit i can be related to the charge at time step k using charger q as*

$$\eta_{i,k+1} = \bar{a}_{q_i}\eta_{i,k} - \bar{b}_{q_i}M_i, \tag{5.3}$$

where $\eta_{i,k}$ represents the SOC for bus i at step k and

$$\bar{a}_{q_i} = e^{a_{q_i}\Delta},\ \bar{b}_{q_i} = e^{a_{q_i}\Delta} - 1. \tag{5.4}$$

Proof. A first-order, continuous model converging to M_j at an exponential rate of a_{q_i} can be expressed as

$$\dot{s}_i = a_{q_i}\eta_i(t) - a_{q_i}M_i. \tag{5.5}$$

The resulting discrete model in Equation 5.4 is obtained by using the exact discretation of an LTI system as is [55]. Assuming $u(t)$ is held constant over the discrete step Δ, the exact discretation of a general LTI system, represented as in $\dot{x}(t) = Ax(t) + Bu(t)$, is given by

$$\begin{aligned} x_{k+1} &= \bar{A}x_k + \bar{B}u_k \\ \bar{A} &= e^{A\Delta} \\ \bar{B} &= \textstyle\int_0^\Delta e^{A-\tau} d\tau B. \end{aligned} \tag{5.6}$$

In Equation 5.5, both a_{q_i} and M_i are constants with no actual control input. To utilize this general discretization formula, Equation 5.5 is rewritten as $\dot{s}_i = a_{q_i}\eta_i(t) - b_{q_i}u(t)$ where $b_{q_i} = a_{q_i}$ and $u(t) = -M_i$. Viewing this new equation in reference to Equation 5.6, the state $x(t)$ is replaced with $\eta_i(t)$ and the matrices, A and B, are replaced with a_{q_i} and b_{q_i}, respectively. Performing these substitutions the discretized forms of a_{q_i} and b_{q_i} become

$$\begin{aligned} \bar{a}_{q_i} &= e^{a_{q_i}\Delta} \\ \bar{b}_{q_i} &= a_{q_i} \textstyle\int_0^\Delta e^{a_{q_i}(\Delta-\tau)} d\tau. \end{aligned} \tag{5.7}$$

The integral in $\bar{b}_{q_i}$ can be solved analytically by taking the antiderivative as

$$\bar{b}_{q_i} = a_{q_i}\Big(-\frac{1}{a_{q_i}} e^{a_{q_i}(\Delta-\tau)} \Big|_{\tau=0}^{\tau=\Delta} \Big) = e^{a_{q_i}\Delta} - 1. \tag{5.8}$$

□

Thus, a visit may leverage this model by taking the initial SOC, η_i, substituting Δ for the charge duration, s_i, and letting q_i represent the convergence rate, the accumulated charge over visit i can be directly calculated using the exact discretization defined above. Figure 5.2 depicts the charge profile estimation utilizing different convergence rates. Note from the figure, the values for the rate of convergence are negative.

5.2 Results

The following results are generated utilizing the same parameters and problem setup as Section 4.6. To approximate the linear charge speeds, the convergence rate for each charger type was adjusted such that the non-linear charge curve intersected the respective linear charge profile at the 80% SOC mark (i.e. the linear portion of the SOC charge profile). It was found that convergence rates of 0.002 and 0.000045 for fast and slow chargers, respectively, intersected their linear counterparts at 80% SOC as shown in Figure 5.3. Furthermore, the heuristic technique was executed utilizing the non-linear battery dynamics. The BPAP and linear heuristic SA plots are included as comparisons against the presented non-linear SA technique. Note that the BPAP and heuristic schedules are generated utilizing linear battery dynamics.

Figure 5.4a depicts the charge schedule generated utilizing non-linear battery dynamics. Similarly to before, the symbol represents the point at which a BEB begins charging. The horizontal line with vertical tic indicate the time over which said BEB receives its charge. Idle chargers were removed for the sake of legibility and space, therefore rows 0-14 of Figure 5.4 represent the slow charging queues and rows 15-29 represent the fast charging queues. The non-linear schedule utilized a total of nine slow charging queues and one fast charger. It is of note that schedules generated are of a similar structure. However, the charge times for the fast chargers are of longer durations due to the non-linear charge profile. That

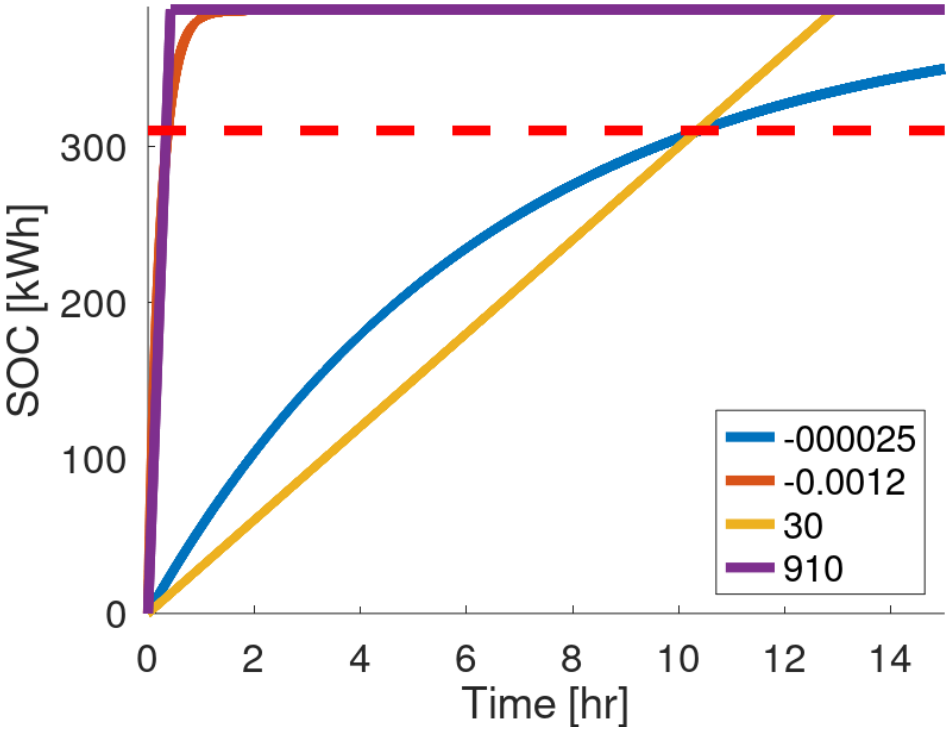


Fig. 5.3: Plot of the linear and non-linear charge profiles. Note the intersections at the 80% SOC line indicated by the horizontal red dashed line.

is, the rate at which the non-linear battery dynamics model accepts additional charge at SOCs above 80% is much lower than that of the linear battery dynamics, causing longer charge times.

Table 5.1: Table of mean, min, and max SOC (kWh) for each charging schedule.

	BPAP	Non-Linear	Linear
Mean	181.327	180.601	182.004
Min	97.000	83.092	91.265

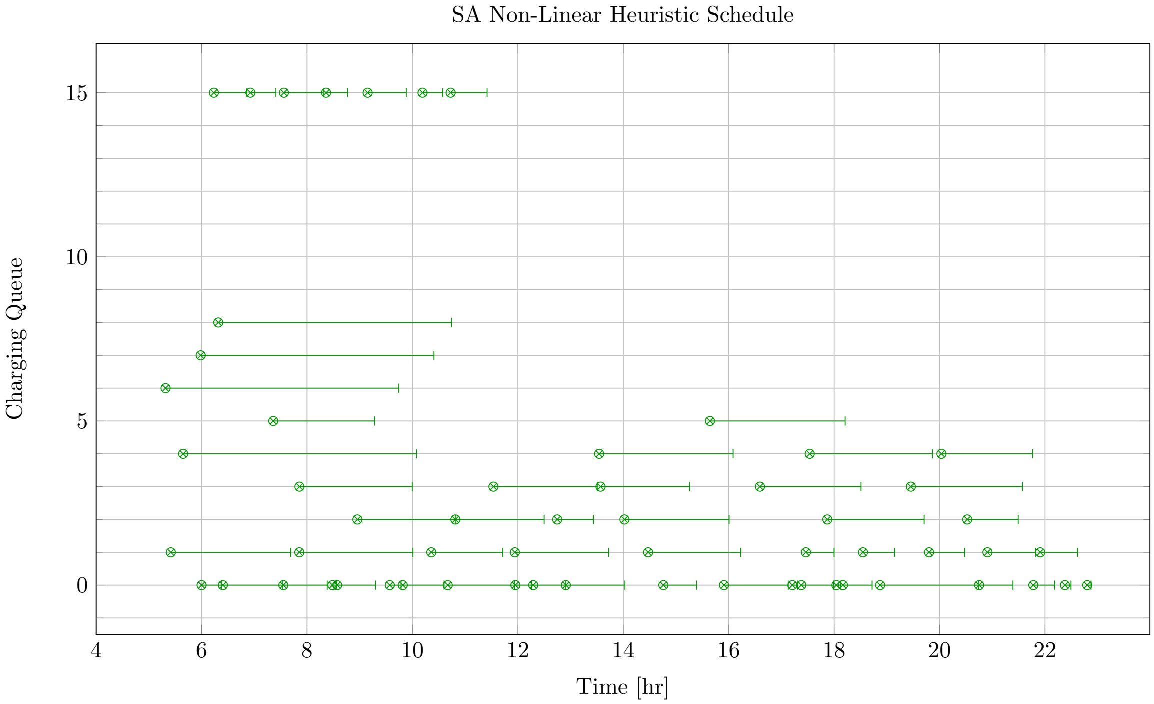
Table 5.1 tabulates the mean and max demand SOC for each. The mean SOC for both the linear and non-linear heuristic SA techniques were approximately the same, but the minimum SOC for the non-linear heuristic SA is lower than that of the linear. This is again due to the non-linear charge profile and the fact that the minimum SOC constraint is soft. This issue may be remedied by including a safety factor term to artificially elevate the minimum SOC threshold, $S_f \nu_i \kappa_i$ where $S_f > 1$; $S_f \in \mathbb{R}$.

Table 5.2: Table of mean and max power demand for each charging schedule.

	BPAP	Non-Linear	Linear
Mean	177.550	246.621	180.858
Max	1910.000	1180.950	1150.950

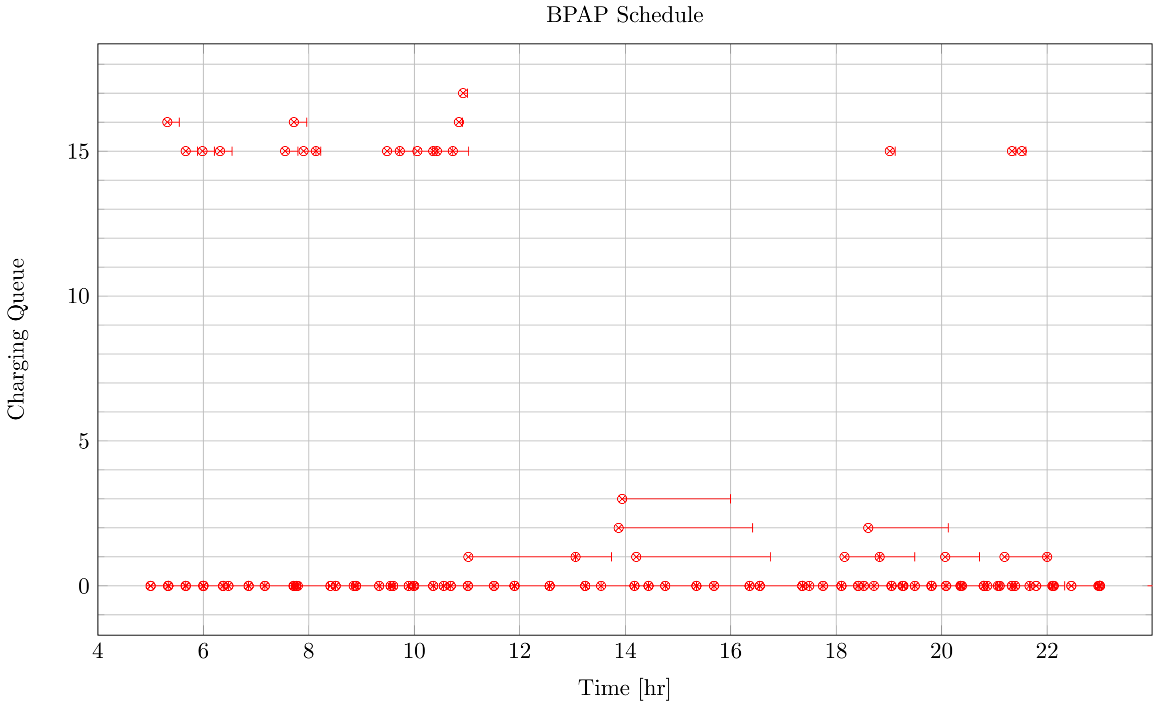
Figure 5.6 plots the power usage over the time horizon and Table 5.2 tabulates the mean and max demand for each. The BPAP and linear SA techniques maintain the lowest mean power demand. This is expected due to the longer charge durations required by the non-linear battery dynamics to reach the same SOC as the linear. However, it is vital to note that even with the non-linear charge profile, the peak demand was still below that of the BPAP and only 30 kW more than the linear SA. That is, even due to the longer charge durations, the schedule was still able to successfully minimize the power demand.

The consumed energy by each schedule is shown in Figure 5.7. The ordering of most energy consumed to least is as follows: non-linear SA, linear SA, and the BPAP with the

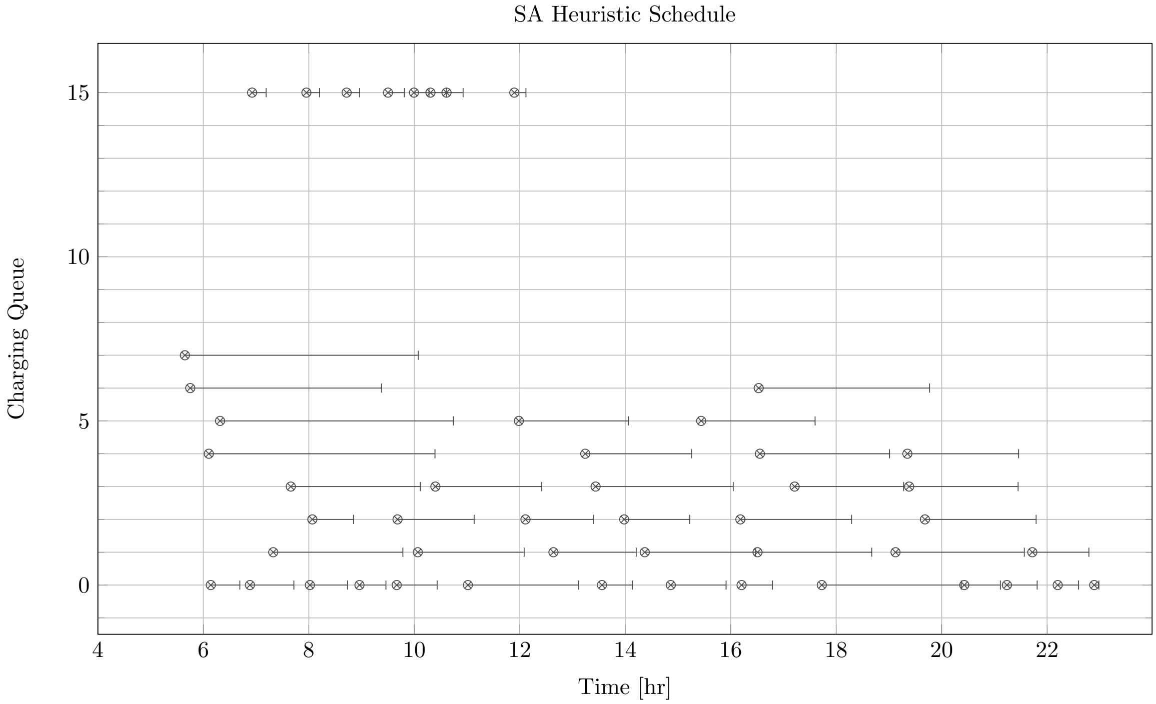


(a) Charge schedule generated by the non-linear SA algorithm.

respective energy consumption for each being 5,919.230 kWh, 4,295.660 kWh, and 4,237.200 kWh. The non-linear SA consumed about 1,163.07 kWh more than the BPAP. The energy consumed is considerably higher again due to the non-linear charge profile.

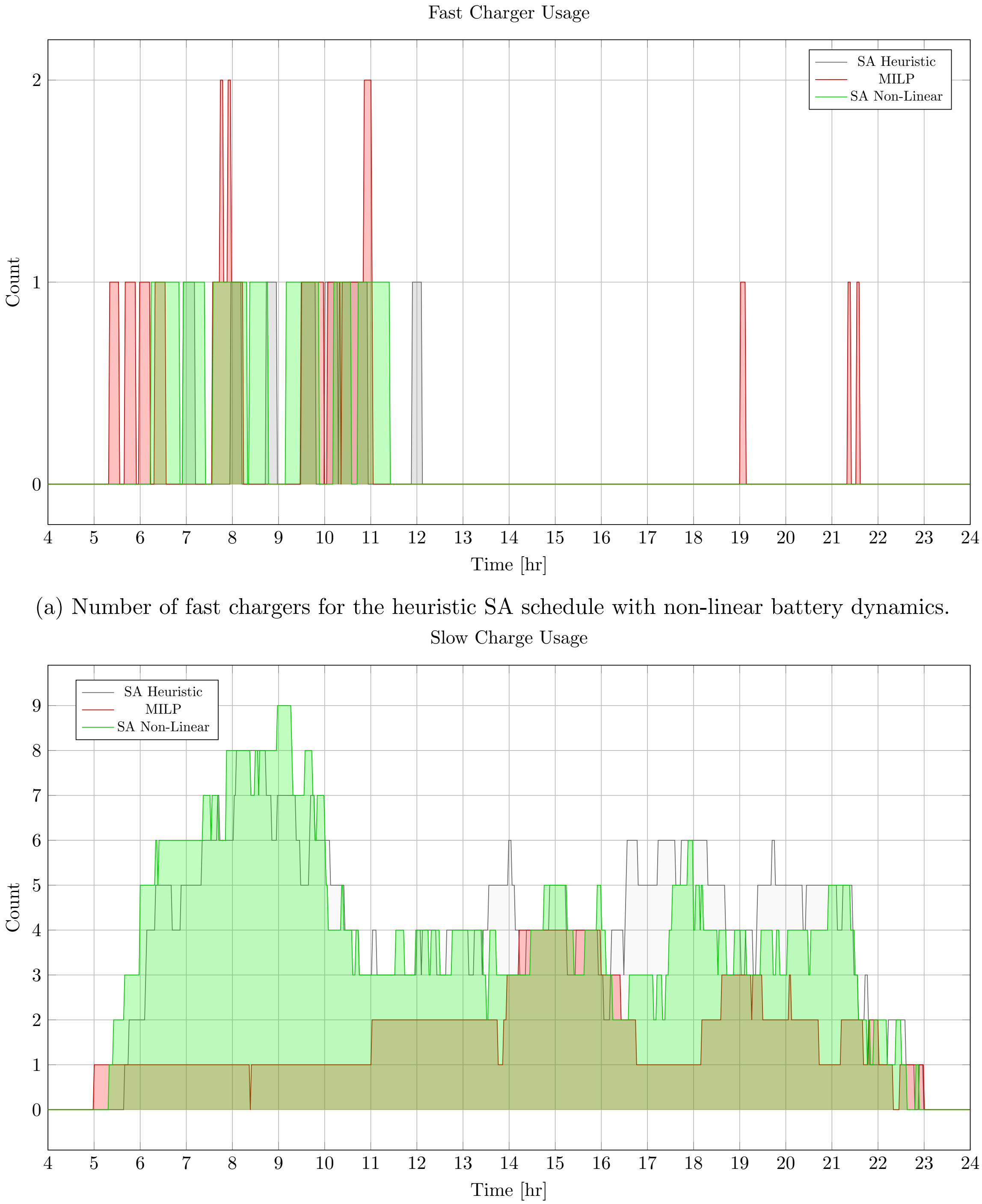


(b) Charging schedule generated by the MILP PAP algorithm.



(c) Charging schedule generated by the SA PAP algorithm using the heuristic strategy.

Fig. 5.4: Charge schedules generated by the Non-linear SA, MILP, and Heuristic SA algorithms.



(a) Number of fast chargers for the heuristic SA schedule with non-linear battery dynamics.

(b) Number of slow chargers for the heuristic SA schedule with non-linear battery dynamics.

Fig. 5.5: Number of slow and fast chargers for each of the schedules.

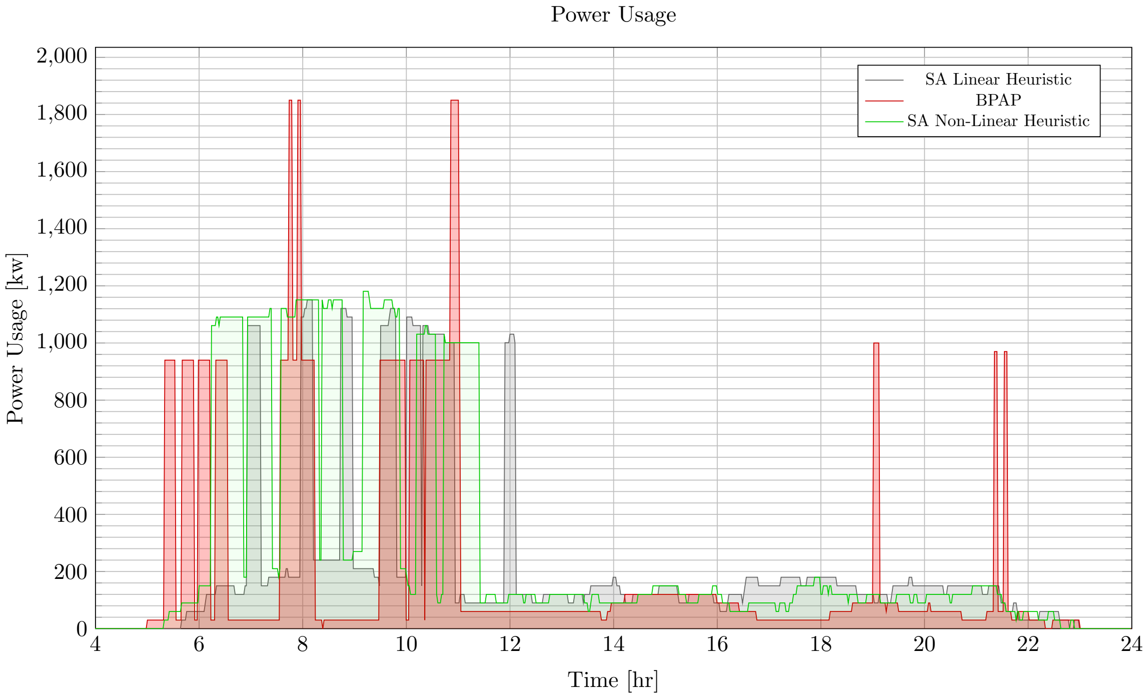


Fig. 5.6: Amount of power consumed by heuristic SA with non-linear battery dynimacs schedule over the time horizon.

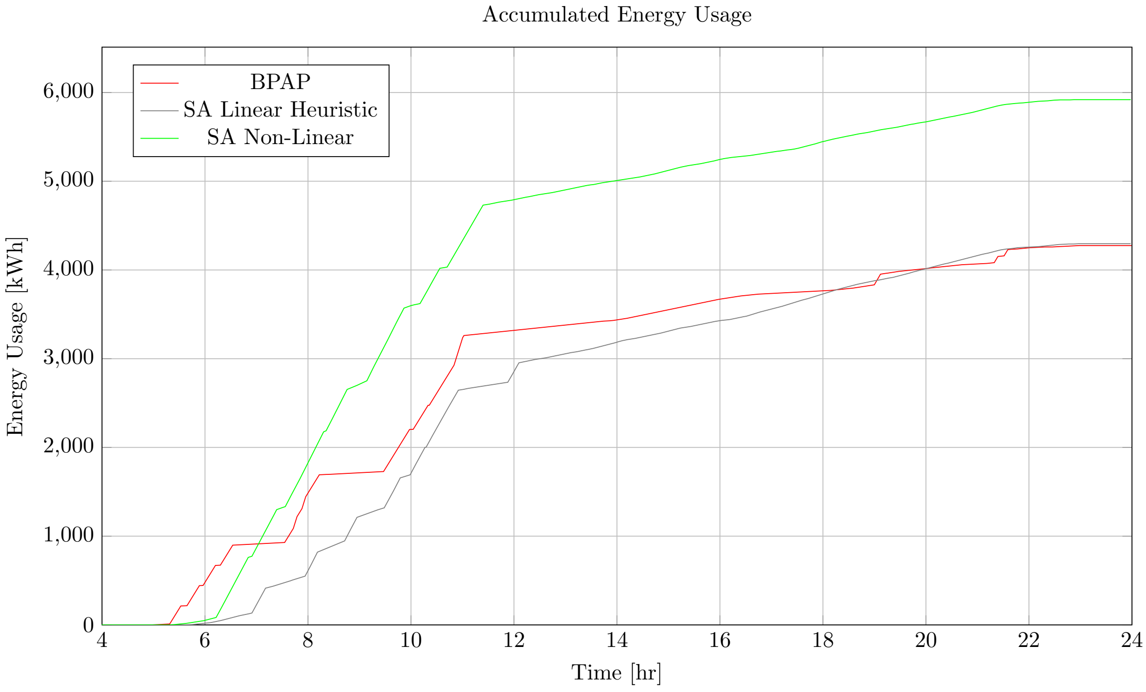


Fig. 5.7: Total accumulated energy consumed by the heuristic SA with non-linear battery dynamics schedule throughout the time horizon.

CHAPTER 6
CONCLUSION

This work presented a novel approach to the BEB charge scheduling problem. Chapter 3 introduced the concept of the Berth Allocation Problem (BAP) which solves the problem of optimally assigning vessels to be serviced. From the BAP, the Position Allocation Problem (PAP) was derived and introduced as the basis from which the work revolved. A Mixed Integer Linear Program (MILP) was developed and from the PAP model that minimized the total charger count, energy consumption, and was able to maintain a minimum State of Charge (SOC) throughout the working day. While MILP models maintain a level of in the form of the objective function and constraints, certain limitations to mathematical modeling are made in order to maintain linearity of the model. Chapter 4 introduced a Simulated Annealing implementation of the MILP PAP the same considerations in the objective function while included an additional cost known as the demand cost. Chapter 5 further introduces an implementation of non-linear battery dynamics that were implemented in the SA PAP model.

Chapter 3 demonstrated an example of the MILP PAP formulation and compared its results to a heuristic-based schedule, referred to as Qin-Modified. The Qin-Modified and MILP schedule utilized a similar amount of fast chargers; however, the MILP schedule more readily used the slow chargers to its advantage when the objective function saw fit. More importantly, the MILP PAP schedule utilized approximately $0.1 \cdot 10^4$ kWh more than the Qin-Modified; however, the charges for the MILP schedule remained above the constrained minimum SOC of 25% , and charged all the buses to 70% at the end of the working day. The Qin-Modified schedule, on the other hand, allowed the SOC of certain BEBs to drop to 0%.

In Chapter 4 an example of the SA PAP algorithm was presented and compared against the MILP PAP and Qin-Modified techniques. The MILP PAP was introduced as a baseline from which to compare the other models due to the fact that it utilizes is modeled is such a way which guarantees optimality (unlike the SA approach). The SA PAP was run utilizing two different neighborhood searching techniques named the quick and heuristic techniques, respectively. The quick SA's objective was to randomly search a wide neighborhood while the heuristic technique was designed to incrementally search a neighborhood by randomly selecting a fast or slow charging queue and then stepping through the queues one at a time. The quick and heuristic have comparable run times at 2275.25 seconds and 3640.4 seconds, respectively, but yielded vastly different results. The Qin-Modified utilized the fewest amount of chargers followed by the MILP, heuristic SA, then the quick SA. The assignment cost applied to the objective function had no effect on the results of the quick SA; however, the heuristic SA was more effective in minimizing the total chargers required. Furthermore, the heuristic SA technique generated a solution approximating that of the MILP, but was unable to minimize the charger count as efficiently. The quick SA utilized all the chargers available (i.e. was unable to minimize the charger count).

Both of the SA techniques were unable to keep the SOC above the 25% SOC threshold with SOC falling to 6.34 kWh for the heuristic SA and 29.8 kWh. The Qin-Modified had the SOC of two BEBs fall to 0% SOC. The schedule that consumed the least amount of energy is the MILP PAP (4256.16 kW) with the heuristic SA coming in second (4797.75 kW). The difference between the two being about 541.586 kWh. The peak demands between the heuristic SA, quick SA, and the MILP were very similar. The MILP had a peak demand of 1910 kW and the quick and heuristic SA had demand peaks of 1911.9 kW. Overall, the heuristic SA was able to generate a schedule that was "in the ballpark" of that of the MILP while further taking the demand cost into consideration.

In Chapter 5, non-linear battery dynamics were derived and introduced in the SA PAP model. The behavior of different convergence rates were demonstrated, and then an example was presented. Convergence rates of 0.1 and 0.002 were utilized for the fast and slow chargers, respectively. Overall, the non-linear SA performed better than the heuristic technique except in terms of packing the schedule. This, however, is taken to be a factor of the SA algorithm only being able to estimate optimality. Similarly to the results of Chapter 4, the non-linear SA was able to approximate MILP PAP schedule well enough while further taking peak demand into consideration the non-linear battery dynamics model.

Further fields of interest are to investigate the performance of the quick and heuristic SA approaches utilizing a denser set of routes to schedule as compared to the MILP. Non-linear dynamics would be of interest to incorporate into the MILP model to further explore comparisons to the SA implementation. Another area of interest would be initializing a MILP solver with a solution generated from an SA algorithm in an attempt to increase performance. Furthermore, an interest in creating a robust strategy by accounting for uncertain arrival times by "fuzzifying" the arrival times. An introduction into this method is provided in Appendix A.

APPENDICES

APPENDIX A

The Fully Fuzzy Linear Program Model

This section introduces an area of interest for future work. As far as the research for this thesis has shown, no other work has provided robustness in their resulting charge schedule. This section outlines a mathematical method to introduce robustness into the MILP constraints by allowing uncertainty in the arrival times of the BEBs via "fuzzification" [38, 56]. The process fuzzifying the model introduces fuzzy variables that contain bounds for which the model can provide solutions for. The mathematics to be introduced promises much potential for further research and development in regard to scheduling BEBs. What follows is an outline and extension of the MILP PAP utilizing Fully Fuzzy Mixed Integer Linear Programming (FFMILP) constraints.

In a realistic scenario, multiple factors such as technical problems, weather conditions, road detours, and various others factor may arise causing buses to arrive earlier or later than anticipated to the station/depot. For crisp models (a traditional MILP), there is no sense of lateness or earliness, thus a model's solution loses validity at the moment any bus does not adhere to the route timing. Fuzzifying the model in turn produces a fuzzy solution that encodes ranges of times that buses may arrive while still remaining a valid solution.

The appendix proceeds as follows. Section A.1 introduces some of the basic concepts of Mixed Integer Linear Programming (MILP), fuzzy set theory, and Fully Fuzzy Linear Programming (FFLP). Section A.4 introduces and derives a Fuzzy BAP (FBAP) model. Section A.5 then adapts the PAP into the FPAP by utilizing the results from the previous sections.

A.1 Preliminaries

This section introduces the concept of fuzzy set theory by providing some basic definitions. The theory is then built on to discuss Fully Fuzzy Linear Programming (FFLP).

FFLP is a branch of Linear Programming where some of the parameters are allowed to have uncertainty which will be further elaborated on. In this section a method of constructing FFLP is introduced. Once the FFLP model has been introduced, a derivation of the Fuzzy BAP (FBAP) is introduced and discussed.

A.2 Fuzzy Sets Theory

This section introduces the notion of fuzzy numbers and some basic definitions. Concepts from this section are referenced from [38, 56–59].

Fuzzy Sets

A sensible method of introducing fuzzy sets is to begin by describing the familiar classic set. A classical (crisp) set is defined as a collection of elements $x \in X$. Crisp sets are binary, either an element belongs in the set, or it does not [57]. For a fuzzy set, what is known as the characteristic function applies various degrees of membership for elements of a given set [57]. The membership of a value in a fuzzy set may differ among other characteristic functions, but their intended purpose remains the same. The membership function is said to be normalized if $\sup_x \mu_{\tilde{A}}(x) = 1$. As an example, Figure A.1a demonstrates a membership function of an LR flat fuzzy number. For a formal definition, consider the Definition A.0.1:

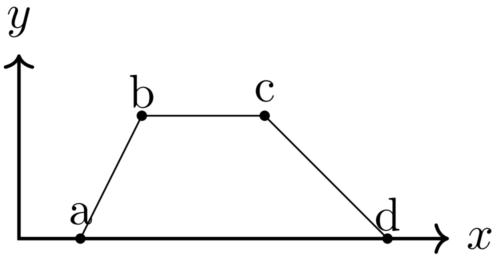


Fig. A.1a: Example of a characteristic function for an LR flat fuzzy number. The line segments $[a, b)$ and $(c, d]$ may be any function that satisfies Definition A.0.2.

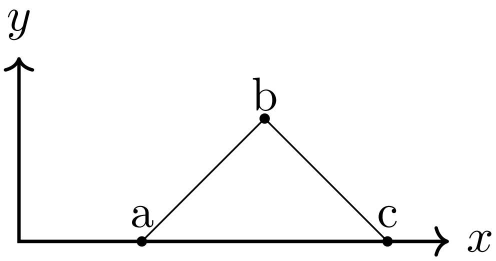


Fig. A.1b: Example plot of a characteristic function for a triangular fuzzy number.

Definition A.0.1. *Let X be a collection of objects (often called the universe of discourse [38]). If X is denoted generically by x, then a fuzzy set $\tilde{A}$ in X is a set of ordered pairs as shown in Equation A.1.*

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\} \tag{A.1}$$

$\mu_{\tilde{A}}$ is called the membership function where $\mu_{\tilde{A}}$ is the mapping $\mu_{\tilde{A}} : X \rightarrow [0,1]$; which assigns a real number to the interval $[0,1]$. The value of $\mu_{\tilde{A}}$ represents the degree of membership of x in $\tilde{A}$.

The shape of a fuzzy number type is defined by membership function. The general definition of fuzzy numbers is known as LR fuzzy numbers [56,57]. Definition A.0.2 describes the property that an L and R functions must have. The L function describes the properties that the left portion of the fuzzy number has, and the R function describes the properties of the right.

Definition A.0.2. *A function $L : [0,\infty) \rightarrow [0,1]$ (or $R : [0,\infty) \rightarrow [0,1]$) is said to be a reference function of the fuzzy number if and only if*

1. *$L(0) = 1$ (or $R(0) = 1$)*
2. *L (or R) is non-increasing on $[0,\infty)$*

The definition of an LR fuzzy number may now be developed given the basis of what properties an L (or R) function must have. Consider Definition A.0.3-Definition A.0.4.

Definition A.0.3. *A fuzzy number $\tilde{A}$ defined on the set of real numbers, $\mathbb{R}$, denoted as the tuple $(m, n, \alpha, \beta)_{LR}$, is said to be an LR flat fuzzy number if its membership function $\mu_{\tilde{A}}(x)$ is defined as shown in Equation A.2. Note that the underscore in the tuple, $(\cdot)_{LR}$ is used to indicate that the tuple is for an LR fuzzy number.*

$$\mu_{\tilde{A}}(x) = \begin{cases} L(\frac{m-x}{\alpha}) & x \leq m, \alpha > 0 \\ R(\frac{m-n}{\beta}) & x \geq m, \beta > 0 \\ 1 & m \leq x \leq n \end{cases} \tag{A.2}$$

Definition A.0.4. *An LR flat fuzzy number $\tilde{A} = (m, n, \alpha, \beta)_{LR}$ is said to be a non-negative LR flat fuzzy number if and only if $m - \alpha \geq 0$ and is said to be non-positive LR flat fuzzy number if and only if $m - \alpha \leq 0$ is a real number.*

A simplification to the LR flat fuzzy number is the triangular fuzzy number, which is what will be utilized in this work (Figure A.1b). The triangular fuzzy numbers shall also be defined over the set of real numbers $\mathbb{R}$. Consider Definition A.0.5 - Definition A.0.8

Definition A.0.5. *A fuzzy number that is represented by $\tilde{A} = (a, b, c)$ is said to be triangular if its membership function is defined as Equation A.3. Figure A.1b depicts a visual representation of a triangular fuzzy number.*

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{(x-a)}{(b-a)} & a \leq x \leq b \\ \frac{(c-x)}{(c-b)} & c \leq x \leq d \\ 0 & otherwise \end{cases} \tag{A.3}$$

Definition A.0.6. *A fuzzy set $\tilde{A}$ in $\mathbb{R}$ is convex if and only if the membership function of $\tilde{A}$ satisfies the inequality*

$$\mu_{\tilde{A}}[\beta x_1 + (1-\beta)x_2] \geq min[\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)]\ \forall x_1, x_2 \in \mathbb{R}\ \beta \in [0, 1]$$

Definition A.0.7. *A fuzzy number is a normal and convex fuzzy set in $\mathbb{R}$.*

Definition A.0.8. *The triangular fuzzy number $\tilde{A}$ is nonnegative* $\iff a \geq 0$.

Fuzzy Arithmetic

If two triangular fuzzy numbers $\tilde{a}_1 = (a_1, a_2, a_3)$ and $\tilde{b}_1 = (b_1, b_2, b_3)$ are nonnegative then the following operations are defined in Equation A.4.

$$\begin{aligned} \tilde{a} \oplus \tilde{b} &= (a_1 + b_1, a_2 + b_2, a_3 + b_3) \\ \tilde{a} \ominus \tilde{b} &= (a_1 + b_3, a_2 + b_2, a_3 + b_1) \\ \tilde{a} \otimes \tilde{b} &= (a_1 b_1, a_2 b_2, a_3 b_3) \end{aligned} \tag{A.4}$$

Comparing Fuzzy Numbers

Fuzzy numbers do not directly provide a method of ordering nor do they always provide an ordered set like real numbers [38]. There are multiple methods for ordering fuzzy numbers, each coming with advantages and disadvantages [60]. Different properties have been applied to justify comparison of fuzzy numbers, such as: preference, rationality, and robustness [38, 56, 61]. These methods are commonly known as ranking functions or ordering functions [38, 56, 58]. Commonly, including in this work, the First index of Yager [62] is used. Let a fuzzy number be represented as $\tilde{A} = (a_1, a_2, ...)$, then the First index of Yager is defined as Equation A.5

$$\Re(\tilde{A}) = \frac{\sum_i a_i}{|\tilde{A}|} \tag{A.5}$$

where $|\cdot|$ represents the cardinality of the fuzzy number. In words, Equation A.5 is merely the average of the values in the fuzzy number. As a result, $A \leq B$ when $\Re(\tilde{A}) \leq \Re(\tilde{B})$ [38].

A.3 Fully Fuzzy Linear Programming

Much like the Linear Programs (LP), Fully Fuzzy Linear Programs (FFLP), it is a class of constrained optimization in which one seeks to find a set of continuous variables that either maximizes or minimizes an objective function, J, while satisfying a set of constraints. The key difference in FFLP is that it is designed to accommodate imprecise information

[38, 56]. In FFLP, the parameters and decision variables are fuzzy and linear. A general FFLP is represented as shown in Equation A.6. The subscripts $\cdot_e$, $\cdot_l$, and $\cdot_g$ indicate to equality, less than, and greater than constraints, respectively. As an example, the notation $\tilde{a}_{ej}$ is read as the e^{th} equality constraint for the j^{th} value in the fuzzy number tuple for the fuzzy number $\tilde{a}$. All variables besides $\tilde{X} = (x_1, x_2, ...)$ are input variables.

$$\begin{array}{lll} \max\limits_{\tilde{x}} & J = \sum_j \tilde{C}_j \otimes \tilde{X}_j & \\ \text{subject to} & \sum_j \tilde{a}_{ej} \otimes \tilde{x}_j = \tilde{b}_e & \forall e = 1, 2, 3, ... \\ & \sum_j \tilde{a}_{lj} \otimes \tilde{x}_j \leq \tilde{b}_l & \forall l = 1, 2, 3, ... \\ & \sum_j \tilde{a}_{gj} \otimes \tilde{x}_j \geq \tilde{b}_l & \forall g = 1, 2, 3, ... \end{array} \tag{A.6}$$

There are many methods of solving FFLP [38, 56, 63, 64]; however, a common strategy is to convert the fuzzy model into a crisp model that can be solved using traditional methods [38]. In [38, 64], the method of converting the FFLP into a crisp MILP is simply done by applying the ranking function to the objective function and breaking the constraints down into a set of crisp constraints as shown in Equation A.7. The constraints are separated according to the definition of fuzzy set multiplication defined in Equation A.4. The fuzzy number index is represented in the exponent rather than the subscript to clearly distinguish between the indexed value in the fuzzy number and the constraint index (i.e. $\tilde{A} = (a^1, a^2, a^3)$). Furthermore, it is assumed that the fuzzy numbers are nonnegative. Although the following equation can be written in terms of general nonnegative LR fuzzy numbers, the parameters and decision variables are written in terms of nonnegative triangular fuzzy numbers. Consider the equality constraint in Equation A.6. For each equality constraint there will be a lower, middle, and upper bound to the constraint. That constitutes three equality constraints. Equation A.7 expands each constraint.

$$\begin{array}{llll} \underset{\tilde{x}}{\max} & J = \mathfrak{R}\Big(\sum_j (c_j^1, c_j^2, c_j^3)(x_j^1, x_j^2, x_j^3)\Big) & & \\ \text{subject to} & \sum_j a_{ej}^1 x_j^1 = b_e^1 & & \forall e = 1,2,3,... \\ & \sum_j a_{lj}^1 x_j^1 \leq b_l^1 & & \forall l = 1,2,3,... \\ & \sum_j a_{gj}^1 x_j^1 \geq b_g^1 & & \forall g = 1,2,3,... \\ & \sum_j a_{ej}^2 x_j^2 = b_e^2 & & \forall e = 1,2,3,... \\ & \sum_j a_{lj}^2 x_j^2 \leq b_l^2 & & \forall l = 1,2,3,... \\ & \sum_j a_{gj}^2 x_j^2 \geq b_g^2 & & \forall g = 1,2,3,... \\ & \sum_j a_{ej}^3 x_j^3 = b_e^3 & & \forall e = 1,2,3,... \\ & \sum_j a_{lj}^3 x_j^3 \leq b_l^3 & & \forall l = 1,2,3,... \\ & \sum_j a_{gj}^3 x_j^3 \geq b_g^3 & & \forall g = 1,2,3,... \\ & x_j^2 - x_j^1 \geq 0 & x_j^3 - x_j^2 \geq 0 & \end{array} \tag{A.7}$$

Note the last constraint is defined to ensure the ordering of the triangular fuzzy number, $x_j^1 \leq x_j^2 \leq x_j^3$. To be more succinct, the FFLP can also equivalently be written as Equation A.8.

$$\begin{array}{lll} \underset{\tilde{x}}{\max} & J = \mathfrak{R}\Big(\sum_j (c_j^1, c_j^2, c_j^3) \otimes (x_j^1, x_j^2, x_j^3)\Big) & \\ \text{subject to} & \sum_j a_{ej}^k x_j^k = b_e^k & \forall e = 1,2,3,... \\ & \sum_j a_{lj}^k x_j^k \leq b_l^k & \forall l = 1,2,3,... \\ & \sum_j a_{gj}^k x_j^k \geq b_g^k & \forall g = 1,2,3,... \\ & x_j^2 - x_j^1 \geq 0 & x_j^3 - x_j^2 \geq 0 \\ & \forall k \in \{1,2,...\} & \end{array} \tag{A.8}$$

Where k has a max value equal to the cardinality to the type of fuzzy number being utilized. This can be further elaborated on by rewriting the inequality constraints as equality constraints by introducing slack variables. This is useful as it represents the formulation in a standard form [39, 65].

The given method is called the Kumar and Kaurs method [56] which is similar in presentation of the Nassiri method presented in [38]. Generally speaking, it is designed to solve FFLP problems with inequality constraints having LR flat fuzzy numbers. Given the FFLP Equation A.6 and assuming that $\tilde{x}_j$ is an LR flat fuzzy number, the problem can be reformulated as Equation A.9 [56].

$$\begin{array}{lll} \max\limits_{\tilde{x}} & J = \sum_j \tilde{C}_j \otimes \tilde{X}_j & \\ \text{subject to} & \sum_j \tilde{a}_{ej} \otimes \tilde{x}_j = \tilde{b}_e & \forall e = 1, 2, 3, ... \\ & \sum_j \tilde{a}_{lj} \otimes \tilde{x}_j \oplus \tilde{S}_l = \tilde{b}_l \oplus \tilde{S'}_l & \forall l = 1, 2, 3, ... \\ & \sum_j \tilde{a}_{gj} \otimes \tilde{x}_j \oplus \tilde{S}_g = \tilde{b}_g \oplus \tilde{S'}_g & \forall g = 1, 2, 3, ... \\ & \mathfrak{R}(\tilde{S}_l) - \mathfrak{R}(\tilde{S}'_l) \geq 0 & \forall l = 1, 2, 3, ... \\ & \mathfrak{R}(\tilde{S}_g) - \mathfrak{R}(\tilde{S}'_g) \leq 0 & \forall g = 1, 2, 3, ... \end{array} \tag{A.9}$$

Expanding the set of equations and using the condensed notation in Equation A.8 we find Equation A.10 [56].

$$\begin{array}{llll} \max\limits_{\tilde{x}} & J = \mathfrak{R}\Big(\sum_j (c_j^1, c_j^2, c_j^3) \otimes (x_j^1, x_j^2, x_j^3) \Big) & & \\ \text{subject to} & \sum_j a_{ej}^k x_j^k = b_e^k & & \forall e = 1, 2, 3, ... \\ & \sum_j a_{lj}^k x_j^k + s_l^k = s_l^{'k} + b_l^k & & \forall l = 1, 2, 3, ... \\ & \sum_j a_{gj}^k x_j^k + s_g^k = s_l^{'k} + b_l^k & & \forall g = 1, 2, 3, ... \\ & \mathfrak{R}(\tilde{S}_l) - \mathfrak{R}(\tilde{S}'_l) \geq 0 & & \forall l = 1, 2, 3, ... \\ & \mathfrak{R}(\tilde{S}_g) - \mathfrak{R}(\tilde{S}'_g) \leq 0 & & \forall g = 1, 2, 3, ... \\ & x_j^2 - x_j^1 \geq 0 & x_j^3 - x_j^2 \geq 0 & \\ & \forall k \in \{1, 2, ...\} & & \end{array} \tag{A.10}$$

Example

To demonstrate the process of decomposing an FFLP into its crisp counterpart, a simple example is to be provided. Consider the following convex non-negative triangular fuzzy FFLP show in Equation A.11. The example is pulled from [64].

$$\begin{aligned} \max_{\tilde{x}} \quad & (1,2,3) \otimes \tilde{x}_1 \oplus (2,3,4) \otimes \tilde{x}_2 \\ \text{subject to} \quad & (0,1,2) \otimes \tilde{x}_1 \oplus (1,2,3) \otimes \tilde{x}_2 \leq (1,10,27) \\ & (1,2,3) \otimes \tilde{x}_1 \oplus (0,1,2) \otimes \tilde{x}_2 \leq (2,11,28) \end{aligned} \tag{A.11}$$

Using the method described in , the FFLP can be expanded into the following form described in Equation A.12. The objective function is expanded using the First Index of Yager. Each constraint is then decomposed into three constraints with slack variables appended to the left-hand side and right-hand side of their respective equation. The constraints for the slack variables are then included to ensure values of the triangular fuzzy numbers for the slack variables are valid. Equation A.12 is now said to be a crisp representation of Equation A.11 in standard form. Solving the FFLP utilizing the Octave LP module (using both the Nasseri and Kumar methods to verify the results), the example problem has a solution as displayed in Table A.1.

$$\begin{aligned} \max_{x} \quad & J = (\tfrac{1+2+3}{3})(\tfrac{x_1^1+x_1^2+x_1^3}{3}) + (\tfrac{2,3,4}{3})(\tfrac{x_2^1+x_2^2+x_2^3}{3}) \\ \text{subject to} \quad & 0x_1^1 + 1x_2^1 + s_1^1 = 1 + s_1^{1'} \\ & 1x_1^2 + 2x_2^2 + s_1^2 = 10 + s_1^{2'} \\ & 2x_1^3 + 3x_2^3 + s_1^3 = 27 + s_1^{3'} \\ & 1x_1^1 + 0x_2^1 + s_1^1 = 2 + s_1^{1'} \\ & 2x_1^2 + 1x_2^2 + s_1^2 = 11 + s_1^{2'} \\ & 3x_1^3 + 2x_2^3 + s_1^3 = 28 + s_1^{3'} \\ & \mathfrak{R}(\tilde{S}_1) - \mathfrak{R}(\tilde{S}_1') \geq 0 \\ & \mathfrak{R}(\tilde{S}_2) - \mathfrak{R}(\tilde{S}_2') \geq 0 \\ & x_j^2 - x_j^1 \geq 0 \\ & x_j^3 - x_j^2 \geq 0 \end{aligned} \tag{A.12}$$

Table A.1: Solution to the crisp representation of the FFLP.

x_1^1	x_1^2	x_1^3	x_2^1	x_2^2	x_2^3	s_1^1	s_1^2	s_1^3	$s_1^{1'}$	$s_1^{2'}$	$s_1^{3'}$
2	4	6	1	3	5	0	0	0	0	0	0

A.4 The Fuzzy BAP

The following is a FFLP model to the continuous dynamical BAP that is able to allocate a quay to an incoming vessel [38]. The model assumes that the arrival time, $\tilde{a}$, berthing time, $\tilde{u}$, and handling time, $\tilde{s}$, are assumed to be known, but imprecise. The objective of the model is to allocate all the vessels to different quays, according to several constraints minimizing the total waiting time for all the vessels. The model in its entirety is presented in Equation A.13. Note that a few modifications are made to the notation to accommodate the notation of [38] while attempting to remain as consistent as possible with the notation presented in this paper. In this section, Q refers to the set of quays. Furthermore, the model is defined over multiple quays, thus indexing for position shall be written as v_{iq}. Similarly, because of the added degree of freedom, the spatiotemporal binary decision variables are represented as σ_{ij}^{q} and ψ_{ij}^{q}. The $\cdot^{q}$ term represents the quay of interest.

$$
\begin{array}{lll}
 & \min \sum_{q\in Q} \sum_{i\in I} (\tilde{u}_{iq} - \tilde{a}_i) & \\
\text{Subject to:} & & \\
 & \sum_{q\in Q} v_{iq} = 1 & \forall i \in I;\ \forall q \in Q \\
 & \tilde{u}_{iq} \geq \tilde{a}_i & \forall i \in I;\ \forall q \in Q \\
 & v_{iq} + l_i \leq L_q & \forall i \in I;\ \forall q \in Q \\
 & v_{iq} + l_i \leq v_{jq} + M(1 - \sigma_{ij}^{q}) & \forall i, j \in I;\ i \neq j;\ \forall q \in Q \\
 & \tilde{u}_{iq} + \tilde{s}_i \leq T & \forall i \in I;\ \forall q \in Q \\
 & \tilde{u}_{iq} + \tilde{s}_i \leq \tilde{u}_{iq} + M(1 - \psi_{ij}^{q}) & \forall i, j \in I;\ i \neq j;\ \forall q \in Q
\end{array}
\tag{A.13}
$$

One may note the similarities to the previously presented BAB and PAP models in Equation A.13. It has yet been defined in a way useful for the purposes for this work. The adaptation of the Fuzzy BAP (FBAP) to the Fuzzy PAP (FPAP) is discussed next in . An example solution of Equation A.13 for 2 quays, provided by [38], is shown in Figure A.2. The figure represents the robustness of the fuzzy berthing plan.

The lines below the small triangle represent the possible early berthing time. Similarly, the lines above the triangle represent the possible late berthing of the vessel. The point where the small point resides is the optimum berthing time. In Figure A.2 it appears that there are conflicts between some of the departures and arrivals of the vessels due to the overlapping lines (e.g. V8 and V2). These overlapping arrivals and departures merely represent the relationship of the fuzzy departure/arrival times of the vessels. For example, if V8 were to depart late, V2 has slack to allow a late berthing and servicing. On the graph, that would represent a berthing time located somewhere above the small triangle.

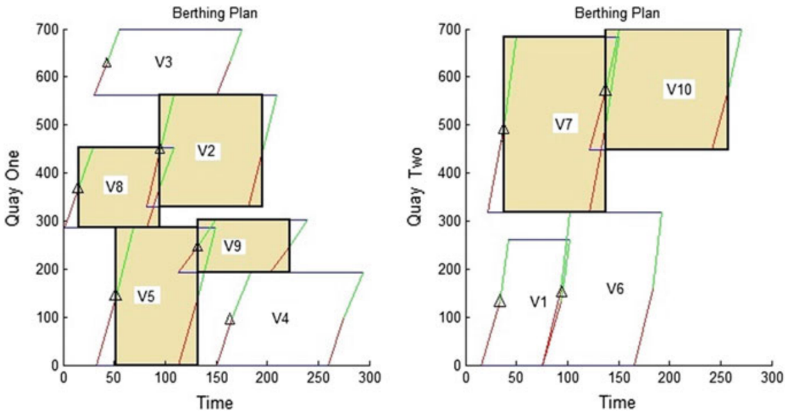


Fig. A.2: Example solution of a fuzzy BAP model over multiple quays. The parallelograms represent the fuzzy solution visually by depicting the lower and upper bounds for the arrival and departure times with the vertial sloped lines. The yellow filled squares represent the average crisp solution.

A.5 Objective Function

This section derives the Fuzzy PAP (FPAP). It is separated into three parts to construct each component individually: objective function, queuing constraints, and charging constraints. Before deriving the FPAP, it is prudent to define the set of fuzzy variables. Similarly to the Fuzzy BAP, the arrival times, $\tilde{a}_i$, berthing time $\tilde{u}_i$, and detach time from the charger $\tilde{d}_i$ are assumed to be imprecise. Thus, indirectly, the service time, $\tilde{s}_i$, the linearization constraint in Equation 3.6, $\tilde{g}_{iq}$, and the initial charge for visit i, $\tilde{\eta}_i$ will also become fuzzy values. Note for the sake of brevity, constraints and objective functions shall be written in their fuzzy form; however, in practice, the equations will be expanded as described in .

Objective Function

Begin by rewriting the crisp objective function with the fuzzy linearization term as shown in Equation A.14.

$$\min \sum_{i=1}^{n_I} \sum_{q=1}^{n_Q} \left(w_{iq} m_q + \tilde{g}_{iq} \epsilon_q \right) \tag{A.14}$$

As shown in , the method of handling the fuzzy values in the objective function is to simply apply a ranking function. Let $\mathfrak{R}$ be defined as the First Index of Yager as shown in Equation A.5. The resulting objective function is shown in Equation A.15.

$$\min \sum_{i=1}^{n_I} \sum_{q=1}^{n_Q} \left(w_{iq} m_q + \mathfrak{R}(g_{iq}) \epsilon_q \right) \tag{A.15}$$

Queueing Constraints

To derive the queuing constraints, the set of crisp constraints that are of interest are rewritten in terms of the fuzzy variables that have been identified prior as shown in Equation A.16. Equation A.16a fuzzifies all the temporal terms. Equation A.16b and Equation A.16c duplicate the constraints $|\tilde{u}_i|$ times. Using triangular fuzzy numbers, that would imply constraints for the lower, middle, and upper bounds of the temporal impreciseness. Similarly, Equation A.16d-Equation A.16f are equivalent to their crisp counterparts, but with fuzzified terms. Equation A.16g simply updates the binary decision term.

$$\tilde{u}_i - \tilde{u}_j - \tilde{s}_j - (\sigma_{ij} - 1)T \geq 0 \tag{A.16a}$$

$$\sigma_{ij} + \sigma_{ji} + \psi_{ij} + \psi_{ji} \geq 1 \tag{A.16b}$$

$$\sigma_{ij} + \sigma_{ji} \leq 1 \tag{A.16c}$$

$$\tilde{s}_i + \tilde{u}_i = \tilde{d}_i \tag{A.16d}$$

$$\tilde{a}_i \leq \tilde{u}_i \leq (T - \tilde{s}_i) \tag{A.16e}$$

$$\tilde{d}_i \leq \tau_i \tag{A.16f}$$

$$\sigma_{ij}, \psi_{ij} \in \{0, 1\} \tag{A.16g}$$

Charging Constraints

Similarly to the other derivations, the crisp constraints are rewritten with their equivalent fuzzy terms. Equation A.17a-Equation A.17e are equivalent to their crisp counterpart with the added lower and upper bounds for the battery charge for visit i. Note that the discharge amount, Δ_i, is not fuzzy. It is assumed that the bus traveled the same distance during the route, thus the same amount of discharge. This is done as a simplification to the estimation of the discharge of a BEB over its given route. Equation A.17f-Equation A.17i fuzzify the linearization terms in a similar manner, but the crisp spatial term, w_{iq} is included. The inclusion of the crisp terms with the fuzzy values are allowed because although the time may be allowed to fluctuate, the specified queue, discharge amounts, or initial/final charge times are the same for each element of the fuzzy temporal values. Thus, when the fuzzy constraints are converted into crisp constraints, the constraint validity is maintained [66].

$$\tilde{\eta}_{\Gamma_i^0} = \alpha \kappa_{\Gamma_i^0} \tag{A.17a}$$

$$\tilde{\eta}_i + \sum_{q=1}^{n_Q} \tilde{g}_{iq} r_q - \Delta_i = \tilde{\eta}_{\gamma_i} \tag{A.17b}$$

$$\tilde{\eta}_i + \sum_{q=1}^{n_Q} \tilde{g}_{iq} r_q - \Delta_i \geq \nu \kappa_{\Gamma_i} \tag{A.17c}$$

$$\tilde{\eta}_i + \sum_{q=1}^{n_Q} \tilde{g}_{iq} r_q \leq \kappa_{\Gamma_i} \tag{A.17d}$$

$$\tilde{\eta}_{\Gamma_i^f} \geq \beta \kappa_{\Gamma_i^f} \tag{A.17e}$$

$$\tilde{s}_i - (1 - w_{iq})M \leq \tilde{g}_{iq} \tag{A.17f}$$

$$\tilde{s}_i \geq \tilde{g}_{iq} \tag{A.17g}$$

$$M w_{iq} \geq \tilde{g}_{iq} \tag{A.17h}$$

$$0 \leq \tilde{g}_{iq} \tag{A.17i}$$

REFERENCES

[1] S. Khan and H. Maoh, "Investigating attitudes towards fleet electrification – an exploratory analysis approach," *Transportation Research Part A: Policy and Practice*, vol. 162, p. 188–205, Aug. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.tra.2022.05.009

[2] J.-Q. Li, "Battery-electric transit bus developments and operations: a review," *International Journal of Sustainable Transportation*, vol. 10, no. 3, pp. 157–169, 2016.

[3] U. Guida and A. Abdulah, "Zeeus ebus report# 2-an updated overview of electric buses in europe," International Association of Public Transport (UITP), Tech. Rep. 2, 2017. [Online]. Available: http://zeeus.eu/uploads/publications/documents/zeeus-ebus-report-2.pdf

[4] M. Xylia and S. Silveira, "The role of charging technologies in upscaling the use of electric buses in public transport: Experiences from demonstration projects," *Transportation Research Part A: Policy and Practice*, vol. 118, pp. 399–415, 2018.

[5] N. Lutsey and M. Nicholas, "Update on electric vehicle costs in the united states through 2030," *The International Council on Clean Transportation*, vol. 2, 2019.

[6] J. S. Edge, S. O'Kane, R. Prosser, N. D. Kirkaldy, A. N. Patel, A. Hales, A. Ghosh, W. Ai, J. Chen, J. Yang, S. Li, M.-C. Pang, L. Bravo Diaz, A. Tomaszewska, M. W. Marzook, K. N. Radhakrishnan, H. Wang, Y. Patel, B. Wu, and G. J. Offer, "Lithium ion battery degradation: what you need to know," *Physical Chemistry Chemical Physics*, vol. 23, no. 14, p. 8200–8221, 2021. [Online]. Available: http://dx.doi.org/10.1039/d1cp00359c

[7] A. Millner, "Modeling lithium ion battery degradation in electric vehicles," in *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*. IEEE, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/CITRES.2010.5619782

[8] R. Deng, Y. Liu, W. Chen, and H. Liang, "A survey on electric buses—energy storage, power management, and charging scheduling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, p. 9–22, Jan 2021. [Online]. Available: http://dx.doi.org/10.1109/TITS.2019.2956807

[9] L.-R. Chen, R. C. Hsu, and C.-S. Liu, "A design of a grey-predicted li-ion battery charge system," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 10, pp. 3692–3701, 2008.

[10] A. Abdollahi, X. Han, G. Avvari, N. Raghunathan, B. Balasingam, K. Pattipati, and Y. Bar-Shalom, "Optimal battery charging, part i: Minimizing time-to-charge, energy loss, and temperature rise for ocv-resistance battery model," *Journal of Power Sources*, vol. 303, pp. 388–398, 2016.

[11] R. Kühne, "Electric buses – an energy efficient urban transportation means," *Energy*, vol. 35, no. 12, p. 4510–4513, Dec 2010. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2010.09.055

[12] A. Hoke, A. Brissette, K. Smith, A. Pratt, and D. Maksimovic, "Accounting for lithium-ion battery degradation in electric vehicle charging optimization," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 691–700, 2014.

[13] M. T. Sebastiani, R. Lüders, and K. V. O. Fonseca, "Evaluating electric bus operation for a real-world brt public transportation using simulation optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2777–2786, 2016.

[14] R. Wei, X. Liu, Y. Ou, and S. K. Fayyaz, "Optimizing the spatio-temporal deployment of battery electric bus system," *Journal of Transport Geography*, vol. 68, pp. 160–168, 2018.

[15] X. Wang, C. Yuen, N. U. Hassan, N. An, and W. Wu, "Electric vehicle charging station placement for urban public bus systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 128–139, 2017.

[16] Y. He, Z. Liu, and Z. Song, "Optimal charging scheduling and management for a fast-charging battery electric bus system," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102056, Oct 2020. [Online]. Available: http://dx.doi.org/10.1016/j.tre.2020.102056

[17] X. Tang, X. Lin, and F. He, "Robust scheduling strategies of electric buses under stochastic traffic conditions," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 163–182, Aug 2019. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2019.05.032

[18] Y. Zhou, X. C. Liu, R. Wei, and A. Golub, "Bi-objective optimization for battery electric bus deployment considering cost and environmental equity," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2487–2497, 2020.

[19] Y. Wang, Y. Huang, J. Xu, and N. Barclay, "Optimal recharging scheduling for urban electric buses: a case study in davis," *Transportation Research Part E: Logistics and Transportation Review*, vol. 100, pp. 115–132, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554516305725

[20] G.-J. Zhou, D.-F. Xie, X.-M. Zhao, and C. Lu, "Collaborative optimization of vehicle and charging scheduling for a bus fleet mixed with electric and traditional buses," *IEEE Access*, vol. 8, pp. 8056–8072, 2020. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2020.2964391

[21] Y. Bie, J. Ji, X. Wang, and X. Qu, "Optimization of electric bus scheduling considering stochastic volatilities in trip travel time and energy consumption," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 12, pp. 1530–1548, May 2021. [Online]. Available: http://dx.doi.org/10.1111/mice.12684

[22] J. Whitaker, G. Droge, M. Hansen, D. Mortensen, and J. Gunther, “A network flow approach to battery electric bus scheduling,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 9098–9109, 2023.

[23] K. Buhrkal, S. Zuglian, S. Ropke, J. Larsen, and R. Lusby, “Models for the discrete berth allocation problem: a computational comparison,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 4, pp. 461–473, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554510001201

[24] A. Imai, E. Nishimura, and S. Papadimitriou, “The dynamic berth allocation problem for a container port,” *Transportation Research Part B: Methodological*, vol. 35, no. 4, pp. 401–417, may 2001. [Online]. Available: https://doi.org/10.1016/S0191-2615(99)00057-0

[25] P. Frojan, J. F. Correcher, R. Alvarez-Valdes, G. Koulouris, and J. M. Tamarit, “The continuous berth allocation problem in a container terminal with multiple quays,” *Expert Systems with Applications*, vol. 42, no. 21, pp. 7356–7366, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417415003462

[26] A. J. Qarebagh, F. Sabahi, and D. Nazarpour, “Optimized scheduling for solving position allocation problem in electric vehicle charging stations,” in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, 2019, pp. 593–597.

[27] F. Rodrigues and A. Agra, “Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey,” *European Journal of Operational Research*, vol. 303, no. 2, p. 501–524, Dec 2022. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2021.12.040

[28] L. Zhang, S. Wang, and X. Qu, “Optimal electric bus fleet scheduling considering battery degradation and non-linear charging profile,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 154, p. 102445, Oct 2021. [Online]. Available: http://dx.doi.org/10.1016/j.tre.2021.102445

[29] M. Duan, G. Qi, W. Guan, C. Lu, and C. Gong, “Reforming mixed operation schedule for electric buses and traditional fuel buses by an optimal framework,” *IET Intelligent Transport Systems*, vol. 15, no. 10, pp. 1287–1303, Jul 2021. [Online]. Available: http://dx.doi.org/10.1049/itr2.12098

[30] M. Rinaldi, E. Picarelli, A. D’Ariano, and F. Viti, “Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications,” *Omega*, vol. 96, p. 102070, Oct 2020. [Online]. Available: http://dx.doi.org/10.1016/j.omega.2019.05.006

[31] J.-Q. Li, “Transit bus scheduling with limited energy,” *Transportation Science*, vol. 48, no. 4, pp. 521–539, Nov 2014. [Online]. Available: http://dx.doi.org/10.1287/trsc.2013.0468

[32] A. Jahic, M. Eskander, and D. Schulz, “Preemptive vs. non-preemptive charging schedule for large-scale electric bus depots,” in *2019 IEEE PES Innovative Smart*

Grid Technologies Europe (ISGT-Europe). IEEE, Sep. 2019. [Online]. Available: http://dx.doi.org/10.1109/ISGTEurope.2019.8905633

[33] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Open source algorithm for smart charging of electric vehicle fleets," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, p. 6014–6022, Sep. 2021. [Online]. Available: http://dx.doi.org/10.1109/TII.2020.3038144

[34] N. Qin, A. Gusrialdi, R. Paul Brooker, and A. T-Raissi, "Numerical analysis of electric bus fast charging strategies for demand charge reduction," *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 386–396, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S096585641630444X

[35] D. Mortensen, J. Gunther, G. Droge, and J. Whitaker, "Cost minimization for charging electric bus fleets," *World Electric Vehicle Journal*, vol. 14, no. 12, p. 351, Dec. 2023. [Online]. Available: http://dx.doi.org/10.3390/wevj14120351

[36] T. Liu and A. (Avi) Ceder, "Battery-electric transit vehicle scheduling with optimal number of stationary chargers," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 118–139, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19304061

[37] J. Dai, W. Lin, R. Moorthy, and C.-P. Teo, *Supply Chain Analysis*. Cambridge University Press, 2008, vol. 119.

[38] M. Bello, G. Nápoles, I. Fuentes, I. Grau, R. Falcon, R. Bello, and K. Vanhoof, *Fuzzy Activation of Rough Cognitive Ensembles Using OWA Operators*, ser. Uncertainty Management with Fuzzy and Rough Sets. Springer International Publishing, 2019, pp. 317–335. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-10463-4_16

[39] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied integer programming*. Wiley, 2010.

[40] F. de Bruin, "Rectangle packing," Master's thesis, University of Amsterdam, 2013.

[41] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-packing-based module placement," in *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*, 1995, pp. 472–479.

[42] M. A. Rodriguez and A. Vecchietti, "A comparative assessment of linearization methods for bilinear models," *Computers and Chemical Engineering*, vol. 48, pp. 218–233, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S009813541200289X

[43] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021. [Online]. Available: https://www.gurobi.com

[44] R. M. Power, "Large general service," https://www.rockymountainpower.net/content/dam/pcorp/documents/en/rockymountainpower/rates-regulation/utah/rates/008_Large_General_Service_1_000_kW_and_Over_Distribution_Voltage.pdf, 2021, [Accessed 03-04-2024].

[45] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics*, 3rd ed., ser. Internationalseries in operation research & management science. Springer International Publishing, oct 2018.

[46] William H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C book set: Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, England: Cambridge University Press, Oct. 1992.

[47] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The theory and practice of simulated annealing," in *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, 1989, pp. 287–319. [Online]. Available: https://doi.org/10.1007%2F0-306-48056-5_10

[48] A. A. Keller, *Multi-Objective Optimization In Theory and Practice II: Metaheuristic Algorithms*. BENTHAM SCIENCE PUBLISHERS, mar 2019. [Online]. Available: https://doi.org/10.2174%2F97816810870541190101

[49] R. Rutenbar, "Simulated annealing algorithms: an overview," *IEEE Circuits and Devices Magazine*, vol. 5, no. 1, pp. 19–26, jan 1989. [Online]. Available: https://doi.org/10.1109%2F101.17235

[50] D. Zhang, Y. Liu, R. M'Hallah, and S. C. Leung, "A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems," *European Journal of Operational Research*, vol. 203, no. 3, p. 550–558, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2009.09.014

[51] Z. Xinchao, "Simulated annealing algorithm with adaptive neighborhood," *Applied Soft Computing*, vol. 11, no. 2, p. 1827–1836, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2010.05.029

[52] A. Brown, G. Droge, and J. Gunther, "A position allocation approach to the scheduling of battery-electric bus charging," 2024.

[53] N. Watrin, R. Roche, H. Ostermann, B. Blunier, and A. Miraoui, "Multiphysical lithium-based battery model for use in state-of-charge determination," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3420–3429, 2012.

[54] B. Arabsalmanabadi, N. Tashakor, A. Javadi, and K. Al-Haddad, "Charging techniques in lithium-ion battery charger: Review and new solution," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct 2018. [Online]. Available: http://dx.doi.org/10.1109/IECON.2018.8591173

[55] W. L. Brogan, *Modern Control Theory*, 3rd ed. Upper Saddle River, NJ: Pearson, Oct. 1990.

[56] J. Kaur and A. Kumar, *An Introduction to Fuzzy Linear Programming Problems*, ser. Studies in Fuzziness and Soft Computing. Springer International Publishing, 2016. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-31274-3

[57] H.-J. Zimmermann, *Fuzzy Set Theory-and Its Applications*, ser. []. Springer Netherlands, 2001. [Online]. Available: http://dx.doi.org/10.1007/978-94-010-0646-0

[58] S. K. Das, T. Mandal, and S. A. Edalatpanah, "A mathematical model for solving fully fuzzy linear programming problem with trapezoidal fuzzy numbers," *Applied Intelligence*, vol. 46, no. 3, pp. 509–519, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10489-016-0779-x

[59] M. Yaghobi, M. Rabbani, M. A. Firozja, and J. Vahidi, "Comparison of fuzzy numbers with ranking fuzzy and real number," *Journal of Mathematics and Computer Science*, vol. 12, no. 01, pp. 65–72, 2014. [Online]. Available: http://dx.doi.org/10.22436/jmcs.012.01.06

[60] C. S. McCahon and E. Stanley Lee, "Comparing fuzzy numbers: The proportion of the optimum method," *International Journal of Approximate Reasoning*, vol. 4, no. 3, p. 159–181, May 1990. [Online]. Available: http://dx.doi.org/10.1016/0888-613X(90)90019-X

[61] M. Jiménez, M. Arenas, A. Bilbao, and M. V. Rodriguez, "Linear programming with fuzzy parameters: an interactive method resolution," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1599–1609, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2005.10.002

[62] R. R. Yager, "A procedure for ordering fuzzy subsets of the unit interval," *Information Sciences*, vol. 24, no. 2, pp. 143–161, 1981. [Online]. Available: http://dx.doi.org/10.1016/0020-0255(81)90017-7

[63] A. Ebrahimnejad, "New method for solving fuzzy transportation problems with lr flat fuzzy numbers," *Information Sciences*, vol. 357, no. nil, pp. 108–124, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2016.04.008

[64] S. Nasseri, E. Behmanesh, F. Taleshian, M. Abdolalipoor, and N. N. TAGHI, "Fully fuzzy linear programming with inequality constraints," *INTERNATIONAL JOURNAL OF INDUSTRIAL MATHEMATICS*, 2013.

[65] R. J. Vanderbei, *Linear Programming*, ser. International Series in Operations Research and Management Science. Springer International Publishing, 2020. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-39415-8

[66] R. Ghanbari, K. Ghorbani-Moghadam, N. Mahdavi-Amiri, and B. De Baets, "Fuzzy linear programming problems: models and solutions," *Soft Computing*, vol. 24, no. 13, p. 10043–10073, Nov 2019. [Online]. Available: http://dx.doi.org/10.1007/s00500-019-04519-w