Optimizing the Relocation Operations of Free-Floating Electric Vehicle Sharing Systems

Zulqarnain Haider†, Hadi Charkhgard‡, Sang Won Kim§, Changhyun Kwon¶

November 2019

Abstract

Free-floating electric vehicle sharing (FFEVS) systems require nightly relocation and recharging operations to better meet the next day’s spatial demand with sufficient battery-levels. Such operations involve not only a crew of drivers to move the shared electric vehicles (EVs), but also a fleet of shuttles to transport those drivers. We consider a decision-making problem for routing shuttles and drivers to recharge and relocate EVs in FFEVS systems. The free-floating nature of FFEVS systems makes their operations costly and cumbersome. Comprehensive studies for relocating EVs and routing shuttles are limited in the literature, and an optimal mix of shuttles and drivers is unknown. We fill this gap by providing a modeling framework for joint decision making and efficient computational tools. We formulate mixed integer programs to model the relocation and recharging operations. Two approaches are devised: sequential and synchronized approaches. In the sequential approach, the movement of EVs is first decided, then the routing of shuttles and drivers is determined. In the synchronized approach, all decisions are made simultaneously. To solve large-scale problems, we devise an efficient computational method, called an exchange-based neighborhood-search method. Our computational method can solve real large-scale instances of car2go in Amsterdam within 10 minutes on a generic computer. Our synchronized approach saves the total shuttle route up to 15% compared to the sequential approach. Wait times of drivers and EVs are also reduced. Our extensive numerical experiments show that when the service area is large, increasing the number of shuttles is more cost efficient than increasing the number of drivers. We also find that when the service area is small, the charging infrastructure is scarce, or the recharging requirements are low, increasing the number of drivers can be more beneficial.

Keywords: smart city operations; electric vehicles; free-floating car sharing; vehicle routing

*Acknowledgement: This work was supported by the INFORMS Transportation Science and Logistics (TSL) Society Cross-Regional grant.

†Industrial and Management Systems Engineering, University of South Florida, Email: zulqarnain@mail.usf.edu
‡Industrial and Management Systems Engineering, University of South Florida, Email: hcharkhgard@usf.edu
§CUHK Business School, The Chinese University of Hong Kong, Email: skim@cuhk.edu.hk
¶Industrial and Management Systems Engineering, University of South Florida, Email: chkwon@usf.edu

Electronic copy available at: https://ssrn.com/abstract=3480725
1 Introduction

An important goal in the development of smart cities is improving efficiency and flexibility in resource utilization. In transportation sector, with the rapid advancements of mobile technologies and devices, on-demand vehicle sharing platforms have emerged as a viable alternative to the notion of car ownership, potentially leading to an efficient utilization of vehicles and urban land. Currently, several key players, such as car2go and DriveNow, are gaining traction with their on-demand, free-floating car sharing (FFCS) services, leading fast expansion of the industry. As of 2019, car2go, the largest car sharing company in the world, operates in 31 cities across North America and Europe with over four million registered members and a total fleet of over 20,000 vehicles. It is forecast that such free-floating mobility services would serve 15 million users worldwide with expected annual total revenue of 3.7–5.6 billion euros by 2020 (Bert et al., 2016; MonitorDeloitte, 2017).

Unlike station-based car sharing models like ZipCar, FFCS systems let users pick an available car and drop it off at any legally permissible parking location within a large designated service area. As a result, users can make customized one-way trips in conjunction with other mobility options, directly addressing the first-mile last-mile problem in urban mobility. For this reason, FFCS systems have the potential to be a true alternative to private car ownership, with much higher adoption than traditional car sharing (Formentin et al., 2015). In North America, for example, it is estimated that a single car sharing vehicle can potentially reduce the need for 6 to 23 cars, substantially reducing the total number of vehicles held by households (Shaheen and Cohen, 2007; Martin et al., 2010). Moreover, FFCS systems can also facilitate more efficient use of urban infrastructure and land by reducing the need for perennially occupied or “locked” parking spots in the city centers with 36–84 m$^2$ of public spaces freed-up per vehicle (Loose, 2010).

Car sharing systems with electric vehicle (EV) fleets are also projected to play a crucial role in making urban transportation systems more sustainable (Firnkorn and Müller, 2011; Le Vine and Polak, 2019). Although EVs are often proposed as one of the most promising solutions to curbing green house gas emissions from the transportation sector, their mass adoption has yet to come. In particular, their short driving range and high fixed costs create psychological concerns to the potential drivers, known as range anxiety and resale anxiety, imposing barriers to the adoption (Lim et al., 2015). In 2017, EVs make up only 1.15% of total U.S. car sales (Bellan, 2018). By relieving the burden of car ownership, free-floating EV sharing (FFEVS) systems can effectively mitigate these psychological adoption barriers: With EVs owned by the service provider rather than the individuals, FFEVS systems relieve the drivers’ concerns about technological risks, future resale value, as well as maintenance (He et al., 2017). In fact, car2go has already deployed full EV fleets in four major cities in Europe with more than 2,000 EVs, and DriveNow operates mixed fleets of EVs and combustion engine cars in over ten European cities.

Despite these potential social benefits, FFEVS systems also engender multiple operational challenges.\footnote{It refers to the problem of unavailability of public transit for the first and last mile of commute, forcing commuters to opt for private modes of transit over public ones.}
First, the vehicles should be available in the right place at the right time. Research into users’ booking behavior has shown that distance to an available vehicle is an important determining factor in a booking decision (Herrmann et al., 2014), and users are only willing to walk up to 500 meters to the available vehicle (Weikl and Bogenberger, 2015). Customers’ rental activities may place the vehicles at less favorable locations and cause a spatial mismatch between supply and demand the following day. Hence, repositioning vehicles to ensure availability under imbalanced demand is the key to a successful operation of the service. Second, EVs need to be relocated with sufficient battery level. Studies show users in FFEVS also exhibit range anxiety: they feel uncomfortable booking an EV if their trip length is 75% or more of the remaining range of the EV (Weikl and Bogenberger, 2015). In addition, there is a strong evidence that demand for FFEVS service is sensitive to the vehicles’ battery levels (Kim et al., 2019). Indeed, vehicle unavailability and range anxiety were cited as main reasons when car2go had to close its EV operations in San Diego (Garrick, 2016).

In the presence of these operational challenges, developing an efficient decision support tool for the relocation operations for FFEVS systems is essential for their successful implementation and expansion. Indeed, such a decision tool could be of great academic and practical significance because of the following reasons. First, EV relocation operation can be very costly. Unlike other shared mobility services such as bikes or scooters, vehicle relocation operation in FFCS systems cannot be done by moving multiple vehicles at a time, making it complex and costly—the cost of relocation can be as much as 15 euros per movement (Chianese et al., 2017). Second, vehicle relocation problem in FFEVS systems is computationally hard. The simplest relocation in a FFCS system would require a service vehicle, or a shuttle, to take a driver to a car and the driver to relocate the car to the target location, and then the driver to be picked up by the shuttle again. Then, the additional layer of recharging the EVs, which requires available charging stations nearby and at least a few hours of charging time, drastically increases the complexity of the relocation problem. Third, there are many important yet unanswered operational questions for FFEVS systems. For example, how would operational resource allocation decisions impact the efficiency of EV relocation operations in FFEVS systems? More specifically, with a given number of available drivers, which would be a better operational strategy: running more shuttles or having more drivers available for relocation? Although FFEVS systems have already been operated in many cities for years, comprehensive studies for the EV relocation problem are limited in the literature. This has motivated this work: we aim to provide an efficient computational tool for the EV relocation problems in FFEVS systems.

The relocation operation in FFEVS systems consists of two levels of decisions: i) EV relocation decision; and ii) Service shuttle routing decision. The EV relocation decision is to determine which EV should be moved to where and how it should be recharged. The shuttle routing is to determine how shuttles should move and pickup/dropoff drivers to fulfill such an EV relocation decision. Ideally, it would be desirable to develop a framework where the two decisions are made jointly and synchronously. This approach, which we call a synchronized approach, is rarely studied in the vehicle relocation literature—instead, most of the
existing studies approach the two decision problems separately and sequentially.

In this paper, we develop a comprehensive modeling and computational framework applicable to the joint optimization of relocation/recharging operations and shuttle routing. Then, based on the computational framework, we conduct a case study using actual data to address important operational questions regarding the relocation operations of FFEVS systems. Our contributions can be summarized as follows.

- We present a combined MIP formulation for the problem of EV relocation for given demand and charging level considerations and the subsequent shuttle routing problem to carry out the recommended relocation. The MIP formulation can be directly applied to small-scale problems and solved by optimization solvers such as CPLEX and Gurobi. Our formulation makes the following important methodological contributions.

  1. Our path-based formulation for relocation problem models each EV route as a separate variable and improves the computability of the EV relocation problem significantly, whereas the existing link-based formulations for EV relocation problem have suffered from increasing complexity as the problem size increases.

  2. This work is the first in the literature that formulates a synchronized EV routing and shuttle routing problem. Our extensive numerical study using real FFEVS data shows that the synchronized decision making improves the total shuttle route length up to 15% relative to the sequential approach. When accumulated over time, this improvement can be a significant benefit for the operation.

- We develop an efficient heuristic algorithm for large-scale problems that can obtain a quality synchronized solution in a few minutes. To derive synchronized decisions, our heuristic algorithm integrates the path-based EV relocation problem with the shuttle routing problem, which is a unique variant of dial-a-ride-problem (DARP) involving charging stops for EVs. The heuristic algorithm, which we call an exchange-based neighborhood-search method (EBNSM), draws upon clustering, exchange based neighborhood search, and a customized exchange algorithm for multiple precedence-constrained DARP.

- We conduct a case study using real data from car2go service in Amsterdam, the Netherlands. Our findings from extensive full-scale numerical experiments, summarized below, inform several important operational decisions.

  1. Our results suggest that, in most cases, increasing the number of shuttles is more cost effective than increasing the number of drivers on each shuttle. In particular, it is especially the case when

---

2This is the result when both the sequential and synchronized approaches use our efficient path-based formulation. Therefore, it is expected the performance gap could be higher if compared with a sequential approach using other computationally less efficient formulation such as link-based ones.

Electronic copy available at: https://ssrn.com/abstract=3480725
the service area is large or when the charging requirements are low (e.g., the initial battery levels are high). In many cases, we find that pairing a separate shuttle with each driver and having the shuttle only to support that driver’s movement, is optimal.

2. Our results also show that reducing the number of shuttles, and therefore, increasing the number of drivers per each shuttle can be beneficial when the service area is small, and the charging requirement is high. In particular, when the charging requirement is high (e.g., the initial battery levels are low), adding more shuttles may not be cost effective as it can only increase each shuttle’s wait time to pick up the drivers.

The remainder of the paper is organized as follows. In Section 2, we review the related literature. In Section 3, we develop and present MIP formulations for synchronized approach and sequential approach for the problem of electric vehicle relocation for given demand and charging level considerations and shuttle routing problem to carry out the recommended relocation. In Section 4 we develop heuristic methods including an exchange-based neighborhood-search method (EBNSM) for synchronized approach. In Section 5, we set up computational experiments using the exact and heuristic approaches, show the efficacy of our algorithms for full sized instances of the problem, and compare the results with the sequential approach. In Section 6, we present managerial insights. Finally, in Section 7 we give some concluding remarks.

2 Literature Review

In this section, we focus mainly on reviewing the literature on operational aspects of the relocation problem in vehicle-sharing systems to highlight our contributions. We refer the reader to Laporte et al. (2015) for a more comprehensive review of other relevant operational problems.

The problem of relocation has been well recognized in one-way car sharing and bike sharing systems. The current vehicle relocation strategies fall into two broad categories. The user-based relocation strategies include incentives, pricing mechanisms and policy interventions to influence user behavior (Weikl and Bogenberger 2013). In one-way car sharing systems, repositioning can be carried out either through operator intervention, e.g., using relocation personnel (Kek et al., 2009; Jian et al., 2016; Bruglieri et al., 2014) and using a trip choice mechanism (de Almeida Correia and Antunes, 2012) or through customers by controlling their actions, e.g., through incentives (Pfrommer et al., 2014). Similarly, in case of bike sharing systems, the problem of system-wide imbalance is further compounded by the two-sided demand for bicycles to rent and empty racks to return. The focus of repositioning is to achieve certain desirable inventory levels either through manual rebalancing using trucks (Raviv et al., 2013) or through incentive mechanisms designed to influence customer behavior (Fricker and Gast, 2016; Haider et al., 2018).

In an initial conceptual paper, Weikl and Bogenberger (2013) present and evaluate several user-based and operator-based relocation strategies for FFCS systems. In a subsequent paper, Weikl and Bogenberger (2015) propose a practice ready six step relocation model for a mixed FFCS system with traditional and
electric vehicles. Based on historical data, the area is categorized into macro zones and an optimization model is used to achieve desired macro level relocation. Rule based methods are used for making intra zone micro-level relocation and refueling/recharging decisions. A similar model for demand-based relocation in FFCS systems is presented by Schulte and Voß (2015) and Herrmann et al. (2014). Caggiani et al. (2017) propose dynamic clustering method to identify the size and number of flexible zones in which to perform repositioning operations. He et al. (2019a) study robust repositioning strategies in dynamic environments.

Only a few papers, however, have considered a unique and critical component of EV relocation operations: shuttle routing. Gambella et al. (2018) present a time-space-network-based formulation for relocating the vehicles in car sharing systems given some demand and battery considerations, but they only consider station-based car sharing systems. In their work, the relocation is carried out by the so called relocators (drivers) who are on board the vehicle itself when traveling between a pair of stations. Kypriadis et al. (2018) propose a minimum-walking car repositioning problem for FFEVS systems. In their model, the drivers walk between the relocation assignments as opposed to travelling on board a shuttle. The problem of shuttle routing to carry out the recommended relocation for FFCS systems is also underserved. Maintaining a dedicated fleet of shuttles and drivers can be expensive and minimizing the cost of relocation operation is one of the key system objectives. For an FFCS system, Santos et al. (2017) consider the problem of shuttle routing given a fleet of shuttles and drivers and provided a set of pre-determined relocation assignments; hence their approach is sequential, rather than joint or synchronized.

Closely related to our work, Folkestad et al. (2020) consider joint decision making for EV relocation and shuttle routing. The modeling approach, formulation, and algorithms in our paper, however, are distinct. In Folkestad et al. (2020), EVs are moved to charging stations rather than close to actual demand points, assuming a situation with ubiquitous charging infrastructure whereby a charging station can be blocked indefinitely. In case of non-availability of charging stations, postponement of charging is considered. This makes their relocation model similar to the one for station-based systems rather than free-floating systems, especially when charging infrastructure is scarce. Many cities and FFCS systems can have saturated charging infrastructure and station blocking may not be an option. Considering access to limited charging infrastructure in recharging planning is a key factor for success of FFEVS systems (He et al., 2019b). Our work models the charging process more explicitly and flexibly. Specifically, our model can handle partial recharging, and therefore the operator can achieve different charging levels in different service zones, to better respond to battery level sensitive customers. The operator can also choose to partially charge all fleet vehicles up to desired charging levels as opposed to charging only a subset of vehicles fully while postponing the charging operation for others. We also note that Folkestad et al. (2020) only relocate EVs with battery levels below a certain threshold whereas our model is flexible to different charging / relocation requirements: Our model can handle purely demand based relocation (i.e., a fully charged EV may need to be relocated to fulfill demand requirements) or a purely recharging based relocation (i.e., a low charged EV needs charging but must stay in its current neighborhood). Also, whereas a hybrid genetic algorithm is
proposed in Folkestad et al. (2020), we employ an exchange-based neighborhood search method to address computational challenges in joint decision making. Lastly, our case study provides important managerial insights for running the relocation operation more efficiently.

3 The Model

In this section, we present a MIP formulation for our problem. Our MIP formulation is divided into three distinct, but related, sub-problems; namely, the EV relocation and recharging problem, the shuttle routing problem, and the synchronization problem. These sub-problems are put together in a single model in Section 3.5

3.1 Modeling Demand and Charging Satisfaction for Each Neighborhood

The relationship between location and demand is incorporated into our model to ensure the optimal placement of the EV fleet across the study area. To incorporate the demand information into our relocation decision, we divide the service area into small neighborhoods \( h \in \mathcal{H} \). The size of the neighborhoods is small enough—less than 250.000 m\(^2\)—so that assumptions of demand uniformity and similarity of demand characteristics throughout the neighborhood hold reasonably well. We find the neighborhood level status data at 12 am at night to find the initial inventory levels, \( I^0_h \). The average values of historical demand data from 6 am to 9 am the subsequent day are calculated for the neighborhood specific desired inventory levels, \( I^d_h \).

Besides the requirements for demand fulfillment, we also consider the relationship between charging levels and the location of a vehicle. We posit that neighborhoods located in different areas of a town may have different charging level requirements for the vehicles located therein. For example, users in downtown could be more sensitive to the vehicles battery levels than those in the suburbs possibly because of the uncertainty in road traffic conditions. The system manager can thus have neighborhood specific charging requirements. The initial battery level of EVs located at each supplier node, denoted by \( c^0 \), can be found using system status data. We use historical trip data and average the battery levels at the beginning of all trips originating in a neighborhood \( h \) to find the desired charging level \( c^f \) for all demander nodes located in the neighborhood. The system manager can, however, update the desired charging level for any neighborhood or decide to charge all vehicles fully. Later, these charging levels are used in determining crucial parameters for our path based electric vehicle relocation and recharging problem.

Let us consider a set of permissible parking spots in the service area. We associate two boolean characteristics with every node in the original network: the occupancy of the node (occupied or unoccupied) and the availability of charging infrastructure at the node (Yes or No). An occupied node without charging infrastructure, called Type 1 node, is designated as a supplier node and represents current EV locations. An unoccupied node without charging infrastructure, called Type 2 node, is designated as a demander node.
An unoccupied node with charging infrastructure, called Type 3 node, is designated as a charger node. In the special case where an occupied node may also be a charging node, called Type 4 node, we create two nodes at the same location; a supplier node for the occupancy and a charger node for the charging station. A demand relocation happens when an EV moves from one of the occupied spots, a supplier to one of the empty spots, or demander. In case the EV needs to be charged, the EV must first visit a charging node. Since the charging process takes time, we create a dummy node for each charging node called a dummy charger node. The EV movement between a charger and its sister dummy charger represents the charging process. In Figure 1, we show the process of conversion to an extended network.

In the extended network, the nodes are designated as supplier, demander, charger or dummy charger based on their functions. Let us call the initial sets of these nodes \( \hat{S}_h, \hat{D}_h, \hat{C}_h \) and \( \hat{C}_d^h \) for each neighborhood \( h \) and \( \hat{S}, \hat{D}, \hat{C} \) and \( \hat{C}_d \) for the whole network, respectively. The supplier nodes correspond to the initial location of electric vehicles in each neighborhood, i.e., \( |\hat{S}| = \sum_{h \in H} I_h^0 \). Since street level parking spots are very close and virtually indistinguishable, we use the central measure of all the spots in a neighborhood and create as many demander nodes in a neighborhood as the size of desired inventory in each neighborhood, i.e., \( |\hat{D}| = \sum_{h \in H} I_h^d \). Finally, each neighborhood can also have dozens of closely located charging stations. If the number of charging stations is greater than \( \max \left( |\hat{S}_h|, |\hat{D}_h| \right) \), we use central measure of all charging stations to create as many charging nodes as the larger of the number of demanders or the number of suppliers in a neighborhood, i.e., \( |\hat{C}_h| = \max \left( |\hat{S}_h|, |\hat{D}_h| \right) \). In case the number of these charging stations is less than \( \max \left( |\hat{S}_h|, |\hat{D}_h| \right) \), number of charging nodes created is equal to the number of charging stations for each neighborhood.

We can further reduce the size of this network by deleting certain nodes depending on neighborhood-level desired inventory and desired charging level using Algorithm 1. In all neighborhoods, only suppliers with initial battery level greater than the desired charging level can be removed, since the rest requires charging. For neighborhoods where \( I_h^0 \leq I_h^d \), suppliers that do not require charging can be removed. An equal number of chargers and demanders can also be removed, while the rest are kept for suppliers which

---

**Figure 1** – Conversion of an original network to an extended network with dummy nodes shown in blue. In the extended network, nodes are shown with their respective designations as supplier, demander and charger nodes.
require charging and suppliers incoming from other neighborhoods to satisfy the leftover demand. On the other hand, for neighborhoods where $I^0_h > I^d_h$, $I^0_h - I^d_h$ suppliers are retained to satisfy demand in other neighborhoods. Out of $I^d_h$ suppliers left, those that require charging are retained, while the rest are removed. An equal number of chargers and demanders are also removed while the rest are kept for suppliers that require charging.

**Algorithm 1: Procedure for removing nodes from the network**

```
Input: \( \hat{S}, \hat{C}, \hat{C}^d, \hat{D}, I^0_h, I^d_h \)
Output: \( S, C, C^d, D \)
1 for \( h \in \mathcal{H} \) do
2 \( \text{removeCount}_h = 0 \); 
3 for \( i \in \hat{S}_h \) do
4 \( \text{if } c^0_i > c^f_h \) then
5 \( \text{removeCount}_h = \text{removeCount}_h + 1 \); 
6 if \( I^0_h \leq I^d_h \) then
7 \( S^r_h = \{ i \in \hat{S}_h | c^0_i < c^f_h \} \); 
8 for \( k \leftarrow 1 \) to \( \text{removeCount}_h \) do 
9 \( \hat{S}_h \leftarrow \hat{S}_h \setminus S^r_h[k], \hat{C}_h \leftarrow \hat{C}_h \setminus \hat{C}^d_h[1], \hat{D}_h \leftarrow \hat{D}_h \setminus \hat{D}^d_h[1] \); 
10 if \( I^0_h > I^d_h \) then
11 \( S^r_h = \{ i \in \hat{S}_h : c^0_i < c^f_h \} \); 
12 for \( k \leftarrow 1 \) to \( \min\{\text{removeCount}_h, I^d_h\} \) do 
13 \( \hat{S}_h \leftarrow \hat{S}_h \setminus S^r_h[k], \hat{C}_h \leftarrow \hat{C}_h \setminus \hat{C}^d_h[1], \hat{D}_h \leftarrow \hat{D}_h \setminus \hat{D}^d_h[1] \); 
14 \( S \leftarrow \bigcup_{h \in \mathcal{H}} \hat{S}_h, \mathcal{D} \leftarrow \bigcup_{h \in \mathcal{H}} \hat{D}_h \); 
15 \( C \leftarrow \bigcup_{h \in \mathcal{H}} \hat{C}_h \); 
```

3.2 Modeling the EV Relocation and Recharging (EVRR) Problem

In second part of our modeling approach, we describe the relocation and recharging decision for electric vehicles (EVs). We call this problem the EV relocation and recharging (EVRR) problem. The traditional relocation problem in car sharing and bike sharing systems involves the decision to relocate vehicles from current node \( i \) to a future node \( j \) to cater to shifting demand needs as shown in Figure 2a. In case of electric cars, node \( j \) can be a charging station. The relocation scenario we consider is decidedly different than traditional setting. In our model, a vehicle can visit up to four nodes during its journey as shown in Figure 2b. From its current node \( i \), a supplier node, it can go to a pair of charger-dummy charger nodes \( k \) and \( l \), one each for receiving and dispatching the vehicle, and once recharged to the required level, it can move to a demander node \( j \). Finally, a vehicle may also choose to stay put at its current node if it is sufficiently charged and does not have to fulfill a demand-based relocation. Our model only specifies the neighborhood
A vehicle needs to be relocated to and the required charging level. Our model, therefore, chooses an optimal route for an EV, not only deciding the terminal node \( j \), if any, of its journey but also deciding which pair of charger nodes, if any, it should visit for recharging purposes.

Let \( N \) be the set of permissible parking spots in the reduced network, i.e., \( N = S \cup D \cup C \). Let \( N' \) be the network with addition of dummy nodes, \( C^d \). The notation for EVRR problem is given in Table 1. Given \( S, C, C^d, \) and \( D \), we can enumerate all the possible EV paths and associate a binary decision variable \( x_p \) with each path \( p \in \hat{P} \). The path variable is 1 if an electric vehicle moves on path \( p \) and 0, otherwise. An EV may or may not require charging. In the former case, the number of possible paths is equal to \( |S| \times |D| \). In case of charging, the number of possible EV paths is \( |S| \times |C| \times |D| \). The total number of paths \( |\hat{P}| \) is equal to \( |S| \times |D| (1 + |C|) \).

We reduce the size of \( \hat{P} \) by removing infeasible or unnecessary paths. Associated with each path \( p \) are parameters \( c^0_p, c^f_p, \) and \( w_p \) representing initial battery level at the supplier, desired charging level at demander, and the charging time required to achieve \( c^f_p \), respectively. Since we have already enumerated all the paths, then for a path \( i \to k \to l \to j \), knowing the initial battery level at the supplier, the final charging level at the demander and the discharging on arcs \( (i,k) \) and \( (l,j) \), one can determine path dependent charging time \( w_p \) on charging arc \( (k,l) \) as follows:

\[
    w_p = \frac{1}{\beta_2} \left( c^f_p - c^0_p + \beta_1 \sum_{(i,j) \in A^p} t_{ij} \right) \quad \forall p \in \hat{P}.
\]

If \( w_p \leq 0 \) for a path, charging stop is not needed and hence the corresponding path \( p : i \to k \to l \to j \) is removed and only direct path \( p : i \to j \) is kept. Conversely, if \( w_p > 0 \), direct path \( p : i \to j \) is removed since a charging station must be visited. The path set \( \hat{P} \) is reduced to \( P \):

\[
    P = \hat{P} \setminus \{ p = i \to k \to l \to j : i \in S, j \in D, k \in C, w_p \leq 0 \} \setminus \{ p = i \to j : i \in S, j \in D, w_p > 0 \}.
\]

The full enumeration of paths and the subsequent reduction of path set ensures that all paths \( p \) will fulfill charging requirements at their respective demander node because of charging time \( w_p \) associated with them. Furthermore, for each path \( p \in P \), the total path time \( l_p \) can be calculated as \( l_p = t_{ik} + w_p + t_{lj} \).
Sets
\(N\) Set of permissible parking spots
\(N'\) Set of permissible parking spots plus the dummy nodes
\(C\) Set of charger nodes
\(D\) Set of demander nodes
\(S\) Set of supplier nodes
\(C_d\) Set of dummy charger nodes
\(H\) Set of demand neighborhoods indexed by \(h \in H\)
\(P\) Set of all feasible paths indexed by \(p \in P\) where each path \(p = (i_0, i_1, \ldots, i_s)\)
\(A\) Set of all possible arcs indexed by \(a \in A\) where each arc \(a = (i, j)\)
\(A(p)\) Set of arcs in path \(p\), where \(A(p) = A^c(p) \cup A^t(p)\), i.e., charging and travelling arcs on path \(p\)
\(N(p)\) Set of nodes in EV path \(p\), where \(N(p) = \{S(p), C(p), C^d(p), D(p)\}\), i.e., supplier, charger, dummy charger and demander nodes of a particular path \(p\)
\(\phi(i)\) Set of paths \(p\) that contain node \(i\), i.e., \(\{p \in P : i \in p\}\)

Variables
\(x_p\) Binary variable; 1 when a vehicle is to be relocated along a path \(p\), 0 otherwise

Parameters
\(t_{ij}\) travel time alongside a travel arc \((i, j)\)
\(w_p\) charging time between charger and dummy charger nodes for path \(p\)
\(c^0_p\) initial battery level at supplier node for path \(p\)
\(c^f_p\) desired charging level at demander node for path \(p\)
\(\beta_1\) rate of depletion; the decrease in battery level of a vehicle per unit of travel time
\(\beta_2\) rate of charging; the increase in battery level of a vehicle per unit of time

Table 1 – Mathematical notation for EVRR

Given the notation in Table 1, we can write the EVRR feasibility problem as follows:

\[
(EVRR) \quad \sum_{p \in \phi(i)} x_p \leq 1 \quad \forall i \in N', \quad (3)
\]

\[
\sum_{p \in P} x_p = \min\{|S|, |D|\}, \quad (4)
\]

\[
x_p \in \{0, 1\} \quad \forall p \in P. \quad (5)
\]
Sets
\[ N' \] Set of permissible parking spots plus the dummy nodes
\[ N'_0 \] Set of permissible parking spots, the dummy nodes and the depots
\[ \delta^- (j) \] set of shuttle arcs entering node \( j \)
\[ \delta^+ (j) \] Set of shuttle arcs leaving node \( j \)

Variables
\[ z_{ij} \] Binary variable which equals one if a shuttle travels directly from node \( i \) to node \( j \), and zero otherwise
\[ y_i \] Number of drivers carried on a shuttle when it leaves node \( i \).
\[ \tau_i \] Integer variable representing the arrival time for shuttle at node \( i \).

Parameters
\[ K \] The number of shuttles available
\[ q \] The maximum number of drivers available for each shuttle

<table>
<thead>
<tr>
<th>Table 2 – Mathematical notation for SR Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the feasibility problem described above, the objective is to find a path for every electric vehicle. Constraint (3) makes sure that each node can only be visited once by an EV while it treads a path ( p ). Constraint (4) ensures that the total number of paths chosen must be equal to total number of EV movements. In case (</td>
</tr>
</tbody>
</table>

3.3 Modeling the Shuttle Routing (SR) Problem

The second part of our combined operational problem is a shuttle routing (SR) problem. The network for SR, \( N'_0 \), is same as the extended network of Section 3.2 with two dummy nodes for depot for beginning and termination of shuttle route added. Let \( z_{ij} \) be 1 if a shuttle travels between nodes \( i \) and \( j \) as part of its route and 0, otherwise. Similarly, let \( y_i \) be the number of drivers on a shuttle while it is leaving node \( i \).

The detailed notation for SR feasibility problem is given in Table 2 while the problem itself is described
henceforth:

\[
\text{(SR)} \quad \sum_{j \in \delta^+(0)} z_{0,j} \leq K, \quad (6)
\]
\[
\sum_{i \in \delta^-(N+1)} z_{i,N+1} \leq K, \quad (7)
\]
\[
\sum_{i \in \delta^-(j)} z_{ij} \leq 1 \quad \forall j \in \mathcal{N}', \quad (8)
\]
\[
\sum_{i \in \delta^-(j)} z_{ij} - \sum_{i \in \delta^+(j)} z_{ji} = 0 \quad \forall j \in \mathcal{N}', \quad (9)
\]
\[
z_{ij} = 1 \implies \tau_j \geq \tau_i + t_{ij} \quad \forall j \in \mathcal{N}', i \in \delta^-(j), \quad (10)
\]
\[
z_{i,N+1} = 1 \implies \tau_{N+1}^k \geq \tau_i + t_{i,N+1} \quad \forall i \in \delta^-(N+1), \quad (11)
\]
\[
z_{ij} = 1 \implies y_j = y_i - 1 \quad \forall j \in S \cup C', i \in \delta^-(j), \quad (12)
\]
\[
z_{ij} = 1 \implies y_j = y_i + 1 \quad \forall j \in C \cup D, i \in \delta^-(j), \quad (13)
\]
\[
0 \leq y_i \leq q \quad \forall i \in \mathcal{N}', \quad (14)
\]
\[
z_{ij} \in \{0, 1\} \quad \forall (i, j) \in A. \quad (15)
\]

The SR feasibility problem (6)–(15) is then to ensure that every shuttle route begins at the starting dummy node (0) and ends at the terminal dummy node (N + 1). Constraints (6)–(7) ensure that total arcs leaving depot node (0) or entering depot node (N + 1) must be less than or equal to the total shuttles K. All the other nodes must only be visited once by one of the shuttles (8). Constraint (9) is flow conservation constraint. Constraint (10) updates the arrival time of a shuttle at a node j when it visits arc (i, j) while constraint (11) finds arrival times at terminal depot node N + 1 for multiple shuttles. Constraints (12) and (13) update the number of drivers on the shuttle as it drops off and picks up the drivers at node j. Finally, constraint (14) ensures the number of drivers on each shuttle must not exceed shuttle capacity q. In the SR feasibility problem described in this section, a feasible shuttle route is any route that begins at a depot, ends at a depot while visiting any number of intermediary nodes and loading and unloading an indeterminate number of drivers. Next section synchronizes the feasibility problems described in Sections 3.2 and 3.3.

### 3.4 Synchronizing EVRR and SR Decision Models

A feasible synchronized EV and shuttle routing problem (f-SYNC) admits only those shuttle routes that include visits to the nodes supplied by the EVRR problem, as opposed to any number of nodes in a feasible SR. Similarly, an f-SYNC solution makes sure that drivers are available to drive the electric vehicles on the routes supplied by the EVRR problem, as opposed to an indeterminate number of drivers loaded and unloaded in a feasible SR solution. Finally, an f-SYNC solution only admits those shuttle routes that respect the time windows E_i of EV routes for the EVRR problem. The following constraints (16)–(19) serve the
Figure 3 – The arrival times of EV at each node $i$ on a path $p$ depend only on shuttle arrival times at Supplier node $i$ (driver drop-off) and Dummy Charger node $l$ (driver drop-off) for a particular path $p$.

Consider a path $p = i \rightarrow k \rightarrow l \rightarrow j$ where $i = S(p), k = C(p), l = C^d(p)$ and $j = D(p)$. The electric vehicle arrival time at nodes belonging to path $p$ can be uniquely determined as follows:

\begin{align*}
E_i &= \tau_i, \\
E_k &= \tau_i + t_{ik}, \\
E_l &= \tau_i + t_{ik} + w_p, \\
E_j &= \max \{ \tau_i + t_{ik} + w_p + t_{lj}, \tau_l + t_{lj} \}.
\end{align*}

For a direct path $p = i \rightarrow j$ where $i = S(p)$ and $j = D(p)$:

\begin{align*}
E_i &= \tau_i, \\
E_j &= \tau_i + t_{ij}.
\end{align*}

The set of equations can be used to determine the path dependent arrival times of each electric vehicle once the arrival times of shuttle at node $i$ and node $l$ are known. As a repositioning shuttle completes its tour, it drops and picks the drivers, who in turn relocate the EVs between nodes. Given a shuttle route, at different nodes, we may have a shuttle, an EV or a driver waiting for some time. Equation (20) states that arrival of EV at supplier node $i$ only occurs after a driver is dropped off by the shuttle at time $\tau_i$. Therefore, the EV wait time is simply equal to the arrival time of the shuttle at the node, i.e., $EV_{\text{wait}} = E_i$. Given $\tau_i$, the charging process for EV begins at $\tau_i + t_{ik}$ and ends at $\tau_i + t_{ik} + w_p$. At charger node $k$, a driver may wait
for a shuttle after dropping the EV, i.e., \( D_{\text{wait}} = \tau_k - E_k \). The charging process for EV begins at \( \tau_i + t_{ik} \), therefore, it does not wait on the node. EV arrival on dummy charger node \( l \) occurs at the end of charging process at time \( \tau_i + t_{ik} + w_p \). At each dummy charger node, if the charging process is over before shuttle arrives, the EV must wait for the driver, i.e, \( EV_{\text{wait}} = \tau_l - E_l \). In this case, EV arrival time at demander \( j \) is \( \tau_l + t_{lj} \). Conversely, if the charging process is not over when the shuttle arrives, the driver must wait for the EV, i.e., \( D_{\text{wait}} = E_l - \tau_l \). the EV arrival at demander \( j \) occurs at \( E_l + t_{lj} \). At demander node \( j \), if the driver has already dropped the EV before shuttle’s arrival, the driver must wait, i.e., \( D_{\text{wait}} = \tau_j - E_j \). In some cases, however, a shuttle may arrive at a demander before the driver brings the EV, i.e., \( SH_{\text{wait}} = E_j - \tau_j \); hence, the shuttle must wait for driver’s arrival.

3.5 Final Formulation for the Synchronized Approach

Thus far, we have presented three disjoint feasibility problems. The EVRR problem finds feasible EV routes and only admits the EV routes that fulfill the demand and charging requirements of the system. Similarly, the SR problem finds feasible shuttle routes, and the f-SYNC admits only those feasible shuttle routes that are synchronized with the requirements outlined in the EVRR problem. In this section, we connect all three disjoint feasibility problems into a combined MIP formulation with an overarching objective function.

\[
\text{(SYNC)} \quad \min \sum_{k \in K} \tau_{N+1}^k, 
\]

s.t. \( \text{EV Relocation and Recharging (EVRR)} \) [3–5], \( \text{Shuttle Routing (SR)} \) [6–15], \( \text{Synchronizing EV and Shuttle Routing (f-SYNC)} \) [16–19].

The objective function (26) of the combined problem minimizes the travel time or makespan of shuttle routes. Thus the combined MIP problem makes sure that only those electric vehicle relocations are carried out that are promising for the overall system objective of minimizing the physical cost of carrying out that relocation using shuttles and drivers.

3.6 Formulation for the Standalone EVRR Problem

EVRR feasibility problem in Section 3.2 describes a relocation problem complete with demand and charging requirements. This relocation model can be used in a standalone manner by the decision makers, interested in relocation alone, using an appropriate objective function.

\[
\text{(P1)} \quad \min \sum_{p \in P} l_p x_p, 
\]

\[
\text{(P2)} \quad \min \max_{p \in P} (l_p x_p). 
\]
We recommend two different objective functions for relocation problem as given by (27) and (28). The former minimizes the total length of the paths selected for EVs while the latter minimizes the maximum length of paths selected. The second objective can be linearized by introducing an auxiliary variable $m$ and adding an extra constraint.

\[
\text{(P3)} \quad \min m \\
\text{s.t.} \quad m \geq l p x_p \quad \forall p \in P
\]

It is easy to show that (P3) with constraint (30) is equivalent to (P2). The relocation problem gives a set of EV routes $\pi$ as an optimal solution. On its own, the solution of EVRR problem for the full city-wide network provides decision makers with optimal relocation decision. The problem will also be used in the heuristic approaches for solving the synchronized EV relocation and shuttle routing problem.

### 3.7 Formulation for the Sequential Approach

We can use the standalone relocation model presented in Section 3.6 and a subsequent shuttle routing model to formulate the sequential approach.

\[
\text{(SEQ-A)} \quad \min \sum_{p \in P} l p x_p, \\
\text{s.t.} \quad (3)–(5), \quad \text{[EV Relocation and Recharging (EVRR)]}
\]

Let $\bar{x}$ be an optimal solution for (SEQ-A). Given $\bar{x}$, solve (SEQ-B).

\[
\text{(SEQ-B)} \quad \min \sum_{k \in K} \tau^k_{N+1}, \\
\text{s.t.} \quad (6)–(15), \quad \text{[Shuttle Routing (SR)]} \quad (16)–(19). \quad \text{[Synchronizing EV and Shuttle Routing (f-SYNC)]}
\]

The sequential approach uses same modeling components as synchronized approach. It is easy to see that the sequential approach gives us solutions which are feasible for synchronized approach but are sub-optimal. Therefore, solutions for sequential approach can be used as initial solutions for synchronized approach.

### 4 Computational Methods

Owing to large size of our problems for the full city-wide network, we devise heuristic approaches to solve the sequential (SEQ) and the synchronized (SYNC) problems. The heuristic approaches for SEQ and SYNC rely on cluster-relocate-route approach to solve the EV relocation and shuttle routing problems. The algorithmic components for the two approaches are described in this section. In Section 4.1, we describe the
heuristic approach for solving the sequential problem. The best solution from sequential approach serves as an initial solution for our main algorithm, the Exchange-Based Neighborhood-Search Method (EBNSM), for solving the SYNC problem as described in Section 4.2.

4.1 Setting the Benchmark: the Sequential Approach

To show the system wide benefits of the synchronized (SYNC) approach to EV relocation and shuttle routing problem, we compare it with the sequential (SEQ) approach. Since we want to compare the two approaches in terms of their relocation decisions, we use a similar cluster-relocate-route approach for the SEQ problem. The steps to solve the SEQ problem are described as Algorithm 2.

**Algorithm 2: The Sequential Approach**

**Input:** $N'_0, K, q$

**Output:** Best EV paths: $X_k$, Best shuttle route: $z$

1. Given number of shuttles $K$, find $K$-centers in the network using greedy approximation approach mentioned in Section 4.1.1.
2. Solve the MIP problem (NCA) described in Section 4.1.2 for creating $K$ clusters of nodes from $K$-centers. Put the objective coefficient $d_{ik} = t_{ik}, \forall i \in N, k \in K$. Let $N_k, S_k, C_k, D_k,$ and $P_k$ be the sets of nodes, suppliers, chargers, demanders and paths for each cluster $k$, respectively;
3. **foreach** $1 \leq k \leq K$ **do**
   4. Given $N_k, S_k, C_k, D_k,$ and $P_k$, solve (SEQ-A). Let $X_k$ be the set of best EV paths;
   5. Given $X_k$, obtain an initial route $r_k$ for the shuttle using greedy approach;
   6. Given an initial shuttle route ($r_k$), run a customized 2-interchange algorithm to get the best shuttle route. Let $ub^k \leftarrow \text{obj(2-int)}$;
7. bestSoln $\leftarrow \sum_{k \in K} ub^k$;
8. bestRoutes $\leftarrow \bigcup_{k \in K} r_k$;

The clusters for the SEQ approach are formed using minimum cost assignment problem NCA with objective coefficient $d_{ik} = t_{ik}$ while the relocation decision is achieved by solving minimum path cost EV relocation and recharging problem (SEQ-A). The MIP formulation SEQ-B can be used to get optimal shuttle routes for moderate sized instances but fails for large instances owing to precedence constraints and loose time windows. Therefore, we use a greedy algorithm to construct an initial shuttle route from optimal EV path set and use the customized 2-interchange algorithm to improve the route.

4.1.1 Finding $K$-centers

As first step for sequential method, we solve a $K$-center problem to find $K$ centers in the network. The $K$-center problem is a well known NP-hard problem [Megiddo and Supowit 1984]. We use a 2-opt greedy approximation algorithm [Plesnik 1987] to solve the problem. We let $t_{ij}$ be the traversal time between nodes $i$ and $j$ and assume the time matrix to be symmetric.
4.1.2 Creating $K$ Clusters

Once $K$-centers have been found, we can assign to each center $k$ a set of nodes which form the $k$-th cluster. The assignment must fulfill certain high level requirements.

1. The number of nodes within each cluster should be approximately equal to form evenly sized clusters.

2. For each cluster, the number of demanders and chargers in the cluster should be greater than or equal to the minimum of the number of suppliers and demanders in the cluster to ensure that each electric vehicle has an available path within the cluster.

Given $K$ clusters, these requirements are equivalent to fulfilling the following conditions: $|S_k| \geq \min \left\{ \frac{|S|}{K}, \frac{|D|}{K} \right\}$, $|D_k| \geq |S_k|$, $|C_k| \geq |S_k|$. The assignment of nodes to clusters is done using an assignment problem that minimizes the total distance from all nodes to the assigned center. Let $r_{ik}$ be a binary variable which is 1 if node $i$ is assigned to center $k$ and 0, otherwise. Similarly let $d_{ik} = t_{ik}$ be the measure of distance from node $i$ to center $k$. The node to center assignment problems (NCA) can be written as follows:

\[
\begin{align*}
\text{(NCA)} & \quad \min \sum_{i \in N} \sum_{k \in K} d_{ik} r_{ik}, \\
\text{s.t.} & \quad \sum_{k \in K} r_{ik} = 1 \quad \forall i \in N \quad (34) \\
& \quad \sum_{i \in S} r_{ik} \geq \min \left\{ \frac{|S|}{K}, \frac{|D|}{K} \right\} \quad \forall k \in K, \quad (35) \\
& \quad \sum_{i \in D} r_{ik} \geq \sum_{i \in S} r_{ik} \quad \forall k \in K, \quad (36) \\
& \quad \sum_{i \in C} r_{ik} \geq \sum_{i \in S} r_{ik} \quad \forall k \in K, \quad (37) \\
& \quad r_{ik} \in \{0, 1\} \quad \forall i \in N, k \in K. \quad (38)
\end{align*}
\]

4.1.3 Finding Optimal EV Routes

Once $K$ clusters have been formed, optimal EV paths in each cluster can be found by simply solving the minimum path cost EV relocation and recharging problem (SEQ-A). The output of the problem is an optimal set of paths for each cluster. The set of paths is subsequently used to find best shuttle routes for each cluster.

4.1.4 Finding Optimal Shuttle Routes

We present a customised 2-interchange algorithm which draws from a 2-interchange procedure for Dial-A-Ride Problem (DARP) presented in Psaraftis (1983). A DARP involves a vehicle picking up and dropping off multiple customers. In our problem, the EVs are equivalent to customers in DARP.

Contrary to a customer in DARP, each EV moves multiple times through the nodes on its route. Therefore, a shuttle must satisfy multiple precedence constraints for each EV. However, given a set of EV paths, a
Algorithm 3: A customized 2-interchange procedure for finding 2-optimal shuttle route

**Input:** \( q \); set of EV paths \( X \); \( \text{Iter} \) = number of 2-interchange iterations

**Output:** \( z \)

Given \( X \), use GreedyProcedure(\( X \)) to obtain an initial route \( r \) as an array \( r[1], r[2], \ldots, r[\text{end}] \):

1. **for** \( k \leftarrow 1 \) to \( \text{Iter} \) **do**
   2. \( \Delta \leftarrow 0; \)
   3. **for** \( i \leftarrow 1 \) to length(\( r \)) - 3 **do**
      4. **for** \( j \leftarrow i + 2 \) to length(\( r \)) - 1 **do**
         5. // Swapping nodes \( i + 1 \) and \( j \) and reversing nodes between them
         6. \( r_{\text{new}} \leftarrow \text{copy}(r); \)
         7. \( r_{\text{new}}[i + 1 : j] \leftarrow r_{\text{new}}[j : i + 1]; \)
         8. **if** \( \neg\text{PrecedenceFeasibilityCheck}(r_{\text{new}}) \) then // for synchronization
            9. **continue**;
         10. **if** \( \neg\text{CapacityFeasibilityCheck}(r_{\text{new}}) \) then // for driver availability
            11. **continue**;
         12. \( \Delta_{ij} \leftarrow \text{RouteImprovement}(r, r_{\text{new}}); \)
         13. **if** \( \Delta_{ij} > \Delta \) then
            14. \( \Delta \leftarrow \Delta_{ij}, r_{\text{best}} \leftarrow r_{\text{new}}; \)
   15. **end**
   16. **end**

Given route \( r \), generate a vector \( z \).

Shuttle route \( r \) can be constructed trivially using a greedy procedure whereby the suppliers are visited first, followed by chargers, dummy chargers and demanders, in this order, while maintaining capacity feasibility for drivers. Given \( r \), a new route \( r_{\text{new}} \) can be constructed by swapping two arcs \((i, i + 1)\) and \((j, j + 1)\) with two new arcs \((i, j)\) and \((i + 1, j + 1)\). Since direction of segment \((i + 1 \rightarrow \cdots \rightarrow j)\) is now reversed, it is necessary to check precedence feasibility and ensure driver availability on the shuttle. The proposed 2-interchange procedure is presented as Algorithm 3. The feasibility and improvement checks are customised for our problem and their algorithmic descriptions are presented, in relative detail, in Section \( \text{A} \) in the appendix.

4.2 Solving the SYNC Problem using EBNSM

In this section, we describe in relative details the steps of EBNSM procedure for finding solutions to synchronized EV relocation and shuttle routing problem (SYNC). EBNSM is an iterative procedure, described as Algorithm 4 for solving EV relocation and shuttle routing problem synchronously. It relies on solution for SEQ method and improves it by iteratively adding neighborhood paths and updating the shuttle routes. Here, we expand on the individual steps.

EBNSM improves the solution from the sequential method by iteratively adding new EV paths and updating the shuttle route. The exchange procedures, ExchangeSuppliers() and ExchangeChargers(), are used to replace a pair of old paths in path set \( X \) with a pair of new paths by exchanging their supplier.
Algorithm 4: Exchange-Based Neighborhood-Search Method (EBNSM)

**Input:** $N_0', S, C, C^d, D, K = \text{Number of Shuttles} = \text{Number of Clusters}, q = \text{Number of Drivers per Shuttle}, \text{SNSIt: Number of iterations for Small Neighborhood Search}

**Output:** Best shuttle routes: $z$

1. Given number of shuttles $K$, use Algorithm 2 to find initial paths $X_k$ and initial shuttle routes $r_k$ for each cluster $k$;
2. for $k \leftarrow 1$ to $K$ do
3.   for $i \leftarrow 1$ to $|X_k|$ do
4.     $\Delta \leftarrow 0$
5.     for $j \leftarrow i + 1$ to $|X_k|$ do
6.       $\bar{X}_k^i, \bar{X}_k^j \leftarrow \text{ExchangeSuppliers}(X_k^i, X_k^j)$ \hspace{1cm} // $\bar{X}_k$ is the updated path set
7.       $r_{k\text{new}} \leftarrow \text{UpdateRoutes}(\bar{X}_k)$;
8.       $\Delta_{ij} \leftarrow \text{RouteImprovement}(r_k, r_{k\text{new}})$;
9.       if $\Delta_{ij} > \Delta$ then
10.          $\Delta \leftarrow \Delta_{ij}$;
11.     if $\Delta = 0$ then
12.        break;
13.     $r_k \leftarrow r_{k\text{new}}, X_k \leftarrow \bar{X}_k$;
14.   for $i \leftarrow 1$ to $|X_k|$ do
15.     $\Delta \leftarrow 0$
16.     for $j \leftarrow i + 1$ to $|X_k|$ do
17.       $\bar{X}_k^i, \bar{X}_k^j \leftarrow \text{ExchangeChargers}(X_k^i, X_k^j)$ \hspace{1cm} // $\bar{X}_k$ is the updated path set
18.       $r_{k\text{new}} \leftarrow \text{UpdateRoutes}(\bar{X}_k, r_k)$;
19.       $\Delta_{ij} \leftarrow \text{RouteImprovement}(r_k, r_{k\text{new}})$;
20.       if $\Delta_{ij} > \Delta$ then
21.         $\Delta \leftarrow \Delta_{ij}$;
22.     if $\Delta = 0$ then
23.        break;
24.     $r_k \leftarrow r_{k\text{new}}, X_k \leftarrow \bar{X}_k$;
25.   $\text{ub}_k = \text{length}(r_k)$
26. Given route $r_k$, generate a vector $z_k$;
27. $\text{ub} \leftarrow \sum_{k \in K} \text{ub}_k$;
28. bestSoln $\leftarrow$ ub, bestRoutes $\leftarrow z_k$;
and charger/dummy charger nodes, respectively. No new nodes are added in the process. However, the visiting order of exchanged nodes must be changed in the shuttle route to ensure that precedence feasibility is maintained. The route update step in EBNSM, $\text{UpdateRoute}(\cdot)$, swaps the positions of the pair of exchanged nodes in the shuttle route according to the updated path set $\mathcal{X}_k$. In doing so, since a supplier is swapped with a supplier, and a charger with a charger, the capacity feasibility of shuttle route is also maintained. Finally, the route improvement check for the new route $r_{\text{new}}$ is done using the same procedure $\text{RouteImprovement}(\cdot)$ as used in Algorithm 3 and described in Section A in the appendix.

5 Case Study: car2go in Amsterdam

We apply our framework to a fully operational system of car2go in Amsterdam, the Netherlands, where the FFEVS service is provided using more than 300 EVs. From the actual data, we take the initial and target locations of EVs that need to be relocated and test the performance of our computational method. We use actual municipal boundaries of smallest size as neighborhoods as shown in Figure 4a. For the computational experiment, we first construct a full network containing 339 neighborhoods, 829 nodes, 332 dummy nodes and 2 nodes representing the depot. We use rules in Section 3.1 to reduce the size of the network. Final network has 155 suppliers, 270 chargers, 270 dummy chargers and 155 demanders. The total number of possible EV paths is $|\mathcal{P}| = 6,267,580$ out of which, 155 paths need to be selected to relocate the 155 EVs. The nodes are depicted in Figure 4b which clearly shows that the current EV locations (suppliers) and the desired locations (demanders) are concentrated in different areas, hence necessitating relocation operations.
5.1 Dataset and Parameters

In our numerical experiments, we use both neighborhood level and spot level data. Based on the two boolean characteristics, i.e., occupancy and availability of charging infrastructure at a node, each node can belong to either of the four node types described in Section 3.2. Each node is also located in a certain neighborhood. All the neighborhoods and the nodes therein have locations described through their coordinates. The locations are used to generate their inter node distances and travel times. We calculate Euclidean distances, and multiply them with a detour index of 1.4 to estimate driving distances \cite{Boscoe2012}. We use city speeds of 30 miles per hour to estimate travel times in minutes. The parameter value of $\beta_1$, the vehicle’s charge depletion rate, is set to 2.5 minutes per unit of charge depletion. Similarly, $\beta_2$ representing a vehicle’s charging rate and given in percentage gain in charging per minute is set at 0.4. The values for $\beta_1$ and $\beta_2$ were derived from actual system data. Barring a major technological shift, these values will likely stay fixed over the medium term. The starting battery level for all vehicles parked at the supplier nodes given by $c^0$ is found from the system status data at 11:59 am on May 8th, 2016. The desired charging level $c^f$ for all nodes located in a particular neighborhood is found as an average of the initial battery levels for all the trips originating in that neighborhood. In our experiments, we allow for full recharging whereby all the EVs are recharged to 100%.

5.2 The Exact Approach

In this Section, we present the results for solving the standalone EV relocation and recharging problem (Section 3.6), sequential EV relocation and shuttle routing problem (Section 3.7) and synchronized EV relocation and shuttle routing problem (Section 3.5) using exact solution approaches. All the experiments were done on a machine with 3.6 GHz CPU clock speed, 16 GB RAM and 64-bit Windows 8 operating system using Java API of CPLEX 12.9.0.

For sequential approach, the relocation and recharging problem (SEQ-A) was solved to optimality for the full network with 829 nodes within 118 seconds. However, the subsequent shuttle routing problem (SEQ-B) is a DARP variant with each EV (customer) being served up to four times by a repositioning shuttle. The problem also involves multiple precedence constraints and loose time windows which depend on shuttle arrivals at the preceding nodes. CPLEX was only successful in solving SEQ-B exactly for up to 10 EVs and 50 nodes. For synchronized EV relocation and shuttle routing problem described in Section 3.5 CPLEX failed to solve the MIP model for the system sized instance discussed in this paper. For this instance, we could not even get a feasible solution for the model. In general, CPLEX was successful in solving the MIP model exactly when the problem size was limited to 15 neighborhoods and 50 total nodes. Even so, in many cases, we were only able to get feasible solutions without a proof of optimality even when we ran the model for one hour.

The results comparing the exact approaches with the heuristic approaches for solving SEQ and SYNC
| $|N'|$ | $|H|$ | $|X'|$ | Value | CPLEX | EBNSM | CPLEX | HEURISTIC |
|---|---|---|---|---|---|---|---|
| 15 | 6 | 2 | Objective | 123.8 | 123.8 | 123.8 | 123.8 |
| | | | CPU Time | 10.65 | 2.23 | 1.80 | 2.07 |
| 25 | 10 | 4 | Objective | 156.7 | 155.3 | 155.3 | 155.3 |
| | | | CPU Time | 3600 | 2.45 | 470.40 | 2.28 |
| 45 | 15 | 9 | Objective | 184.6 | 184.8 | 184.8 | 184.8 |
| | | | CPU Time | 3600 | 2.76 | 3600 | 2.69 |

**Table 3** – Value of Objective Function for Different Network Sizes with a) Exact Method for SYNC b) EBNSM c) Exact Method for SEQ and d) Heuristic Approach for SEQ for the 15, 25, and 45 Node Networks

...problems for small instances are provided in Table 3. Generally, for the smaller sized problems, the difference between synchronized and sequential approaches was not substantial owing to the fact that even in sequential approach, the relocation objective is to minimize the total EV path length of all EVs. In smaller networks, this is a good proxy for minimizing the total shuttle route. The results obtained using heuristic approaches were comparable to those with exact approaches for small instances while taking considerably smaller time. Therefore, for solving the system sized instance of the sequential and synchronized problems, we use the heuristic approaches presented in Section 4.

### 5.3 Computational Performance of EBNSM

We run EBNSM for instances of the SYNC problem for various numbers of shuttles $K$ and number of drivers $q$. Each shuttle services one cluster. We solve SEQ-A to get an initial path set $X_k$ and use the 2-interchange procedure to get shuttle route $r_k$. Since we do not have a lower bound for the shuttle route length, we use number of iterations to be the termination criterion. Moreover, if $K$ is small, each shuttle will have larger route length. Therefore, the number of 2-interchange iterations are inversely proportional to $K$. We use $1000/K$. The numbers are empirically chosen as when the 2-interchange procedure has stopped improving the objective. Given the size of the instances, we also want to limit the run time of the algorithm to 20 minutes. Since the size of clusters varies in inverse proportion to number of shuttles/clusters $K$, we run $500/K$ iterations of the neighborhood procedure for finding the “best” EV relocation decision. We use a depth-first strategy and for each iteration select the exchange with highest improvement. For each iteration of the neighborhood search, we update the shuttle route while maintaining precedence and capacity feasibility to get the best shuttle route given the EV paths selected. When the iterations have stopped giving improvement, we terminate the exchange procedure. Each instance of EBNSM for SYNC problem takes less than 10 minutes to terminate. Moreover, the number of iterations can be modified according to the size...
of the problem.

5.4 Value of the SYNC Approach

We compare the results by the SYNC and SEQ approaches. Given values of \( K \) and \( q \), the output of each instance of algorithms is in terms of total route length in minutes denoted as \( L \). We can also calculate following parameters: total number of personnel \( (P = Kq + K) \) and average route time per shuttle in hours \( (T = \frac{L}{60 \times K}) \). We also calculate total wait times for shuttles \( (SH_{\text{wait}}) \), EVs \( (EV_{\text{wait}}) \) and drivers \( (D_{\text{wait}}) \) for the best route. We compare the performance of SYNC and SEQ approaches for 100% recharging situation. For full recharging, the SYNC approach outperforms the SEQ approach in terms of total route length for all instances of the problem. On average, the route length for SYNC approach is 15% shorter as compared to the SEQ approach. The difference in objective function value varies considerably, in 5–28% range, across instances. Generally, the difference grows larger when the number of drivers on board each shuttle increases as can be seen in Figure [5]. However, if the number of drivers is increased so much that some drivers are not utilized at all, the difference between the two approaches shrinks. As with the exact approach, when service area is very small, i.e., when the number of clusters/shuttles is very large, the difference between the SYNC and SEQ approaches decreases.

We also compare the cumulative wait times for all EVs \((155)\), all shuttles \((K)\) and all drivers \((K \times q)\) for SYNC and SEQ approaches. The SYNC approach outperforms the SEQ approach in terms of EV wait times. On average, the average improvement for full recharging is 21%. The improvement for driver wait times is also 24% on average. However, the difference decreases as number of drivers per shuttle increases. In case of shuttles, the wait times are negligibly small for small \( K \) values. As \( K \) increases, the wait times oscillate and SEQ approach has, on average, smaller shuttle wait times in case of full recharging.

6 Resource Allocation and Operational Efficiency

An important managerial decision in the operation of EV relocation for FFEVS systems is how operational resource allocation decisions impact the efficiency of operation. In this section, we study the relationship between the operational cost of EV relocation and the size of the shuttles and number of drivers. Specifically, using our computational method and car2go data, we conduct an extensive numerical experiment by calculating the relocation cost for different number of shuttles (clusters) and number of drivers on board each shuttle. We find that besides the per unit shuttle and personnel costs, the size of service area and the initial battery levels of EV fleet are important determining factors for an efficient operation of EV relocation in FFEVS systems.
6.1 Cost Parameters

Let $\Gamma_D$ and $\Gamma_{SH}$ be the hourly costs for personnel and shuttles, respectively. The total cost of relocation operation can be calculated as $C = P \times T \times \Gamma_D + K \times T \times \Gamma_{SH}$. It is the sum of personnel cost and the shuttle operating cost. In our experiments, we assume the labor cost to be $40/hr, and the per hour operating cost for a shuttle to be $24. We also limit the available time for relocation operation to 7 hours. Therefore, we only consider the instances which achieve the relocation operation within the time limit.

6.2 Sensitivity Analysis for the Number of Shuttles and Drivers

We consider the sensitivity to the number of repositioning shuttles $K$ and the number of drivers $q$ on board each shuttle. We vary the value of $K$ from 1 to 30 and $q$ from 1 to 12. By varying $K$ and $q$, we not only vary the extent of clustering, but also find out the best way to distribute manpower and shuttle resources to carry out system-wide relocation of EVs. Given a number of total personnel $P$, a manager may be interested in the best resource allocation in terms of shuttles and drivers. For instance if $P = 6$, the possible resource allocation combinations could be $(K, q) = \{(1, 5), (2, 2), (3, 1)\}$, since each shuttle also requires a driver. In Figure 6, we map the cost of all such combinations for $P$ ranging from 2 to 50 with $K$ ranging from

![Figure 5](https://ssrn.com/abstract=3480725)
1 to 30 and \( q \) ranging from 1 to 12 when 155 EVs are to be relocated. We observe that in most cases the combination \( (K, q) \) with larger number of shuttles is also more cost effective. However, as shown in boxes in Figure 6 in some cases, for given \( P \), increasing \( K \) actually increases the cost of operation and combinations with smaller number of shuttles and larger number of drivers per shuttle are more cost effective. When we repeat the same experiment with a smaller network of 10 EVs varying \( K \) from 1 to 5 and \( q \) from 1 to 12, we obtain similar results, shown in Figure B.1 in the appendix, where choosing the highest \( K \) is not the best option. Details of the instances for which \( (K, q) \) combinations with smaller number of shuttles are more cost effective are given in Table 4.

Therefore, it is important for a decision maker to understand when it is advantageous to increase drivers as opposed to shuttles and vice versa, for given number of available personnel. Systems with relatively high labor cost should consider single person mobility options like bicycles, scooters or mopeds so a cluster of EVs could be assigned to an individual driver without the shuttles moving him/her around. Changing the per unit cost parameters may also impact these results and the optimal strategy may shift to two drivers per shuttle rather than one driver. However, increasing the number of drivers per shuttle beyond two drivers invariably gives diminishing returns due to large driver wait times on the shuttle.

### 6.3 Analyzing the Shuttle, EV, and Driver Wait Times

For given \( \Gamma_D \) and \( \Gamma_{SH} \), the average route time per shuttle given by \( T \) also impacts the total cost of operation. \( T \) is in turn influenced by shuttle wait times. As \( K \) is increased, each shuttle is responsible for relocating smaller number of EVs. For \( K = 1 \), a shuttle must relocate all 155 EVs and for \( K = 14 \), the number drops to less than 12 EVs. As the number of EVs per shuttle decreases, the corresponding shuttle wait times start increasing as can be seen in Figures 8 and 9. This is owing to the fact that for smaller number of EVs, shuttles must wait for EVs to get recharged before picking up the final batch of drivers from the demander nodes. Therefore, after some point, adding more shuttles only increases the wait times of the shuttles, and it becomes beneficial to increase the number of drivers instead, as shown for a small instance in Figure 7.

### 6.4 Analyzing the Impact of Initial Battery Levels

The difference between initial battery levels and desired charging levels is also an important factor in relocation operations. A larger difference signifies longer recharging processes and larger wait times for shuttles and drivers. This impact is considerable especially for small-scale networks where an EV’s charging process may delay the trip completion. For low initial battery levels, the charging process takes longer and adding extra shuttles only contributes to shuttle wait times. However, if the initial battery levels get higher (so the charging requirements become lower), increasing the number of shuttles can still be beneficial. As shown in Figure 10, we increase the initial battery levels from \( c^0 \) to \( \alpha c^0 \) uniformly. As \( \alpha \) increases, the charging requirement decreases, and the cost of operation with larger number of shuttles (green asterisks) becomes more favorable.
Figure 6 – For a Large Network with 155 EVs and given a certain number of personnel and per unit costs, it is advantageous to cluster more and increase the number of shuttles

7 Conclusion

This paper presents a mathematical model to solve the problem of repositioning and recharging of electric vehicles in a FFEVS system. Since in many such systems, EVs are moved by a crew of shuttles and drivers, we propose that the relocation decision be made in synchronization with shuttle routing decision to minimize the cost of relocation operation. In contrast, most current approaches make the relocation and shuttle routing decisions sequentially. Our unique path based formulation can solve problems of moderate size. For system sized instances, we propose an exchange based neighborhood search method which draws from the mathematical model and solves all the instances within 10 minutes. Comparison of synchronized approach with sequential approach shows that the former improves the total length of shuttle routes and in turn the cost of relocation operation by 15% on average. FFEVS systems require an elaborate relocation operation and improving the cost of such an operation improves the bottom line of these systems. Moreover, our model achieves complete system wide repositioning and recharging, therefore improving the distribution of EVs and directly addressing the issues of demand imbalance and range anxiety in FFEVS systems.
Figure 7 – Possible relocation operation alternatives when number of personnel = 6, number of EVs = 10. For \((K, q) = (3, 1)\), shuttles wait 52% of the time; for \((K, q) = (2, 2)\), shuttles wait 37% of the time; for \((K, q) = (1, 5)\) shuttles wait 1.8% of the time.

Figure 8 – Shuttle wait times increase as the number of shuttles increases

Figure 9 – Shuttle wait times increase as the number of shuttles increases
Table 4 – Comparison of Shuttle Route Length and Operating Costs for For Various \((K, q)\) Combinations Given Number of Personnel for a Small Network with Number of EVs = 10: The instances in the table, marked red in Figure [B.1] in the appendix, correspond to situations when increasing the number of drivers is advantageous.

The data for our experiments comes from a real life FFCS operator and the instances used in this paper represent the complexities of an actual relocation operation. This paper presents EBNSM which uses cluster-relocate-route approach to solve the system-sized relocation instances for relocating and recharging. For the largest instance solved in this paper, we relocate 155 EVs while increasing the system wide average charging levels from 42 % to 90 %. The model is also flexible to changes in system status, initial battery levels, and desired demand configuration. It allows for partial recharging of EVs and their relocation close to the actual demand points.

We conduct a variety of experiments with different numbers of shuttles and drivers per shuttle to find out the most cost effective human resource allocation. The results suggest that given a certain number of personnel, it is more cost effective to increase the number of shuttles rather than number of drivers on each shuttle, especially when the service area is large. This trend may be reversed for small service areas and small number of EV relocations per shuttle. In these cases, adding extra shuttles only adds to wait times and increasing the number of drivers may be beneficial. This implies that although increasing the number of drivers on shuttles improves the route length, the improvement is not justified due to extra cost. Therefore,
Different Initial Charging Levels ( = \( ac^2 \))

![Figure 10](image)

**Figure 10** – As initial battery levels increase, it becomes beneficial to increase the number of shuttles systems with relatively high labor cost should consider single person mobility options like bicycles, scooters or mopeds as this will relieve the cost of an extra shuttle and driver.

**References**


de Almeida Correia, G. H., A. P. Antunes. 2012. Optimization approach to depot location and trip selection


Appendix for
Optimizing the Relocation Operations of Free-Floating Electric Vehicle Sharing Systems

A Two-Interchange Algorithm for Multiple Precedence Constraints

We present a customised 2-interchange algorithm which draws from a 2-interchange procedure for Dial-A-Ride Problem (DARP) presented in Psaraftis (1983). A DARP involves a vehicle picking up and dropping off multiple customers. A DARP tour, unlike that of a Traveling Salesman Problem (TSP) (Lin, 1965), must satisfy precedence constraints since origin of each customer must precede his/her destination on the route. In our problem, the EVs are equivalent to customers in DARP. Contrary to a customer in DARP, each EV moves multiple times through the nodes on its route. Therefore, a shuttle must satisfy multiple precedence constraints for each EV. We borrow and extend the notation used in Psaraftis (1983).

A given shuttle route $r$ of length $M = \text{length}(r)$ can be described as a sequence $\left(\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_i, \ldots, \hat{S}_M\right)$ where $i$ represents the $i$-th stop of the route and $\hat{S}_i$ is defined using the following symbolic values:

$$
\hat{S}_i = \begin{cases} 
0 & \text{if } i = 1 \text{ or } i = M \text{ (Depots)} \\
n^+ & \text{if shuttle visits supplier node of EV } n \text{ at node } i \\
n^\geq & \text{if shuttle visits charger node of EV } n \text{ at node } i \\
n^< & \text{if shuttle visits dummy charger node of EV } n \text{ at node } i \\
n^- & \text{if shuttle visits demander node of EV } n \text{ at node } i
\end{cases}
$$

for all $i = 1, 2, \ldots, M$. Alternatively, the shuttle route can be represented through matrix $\left[ m(n, i) \right]$ where $m(n, i)$ is the status of EV $n$ at the $i$-th stop of the shuttle tour ($n = \text{EV number}, \ i = 1, 2, \ldots, M$).

$$
m(n, i) = \begin{cases} 
5 & \text{if supplier for EV } n \text{ has not been visited so far.} \\
4 & \text{if charger for EV } n \text{ has not been visited so far.} \\
3 & \text{if dummy charger for EV } n \text{ has not been visited so far.} \\
2 & \text{if demander for EV } n \text{ has not been visited so far.} \\
1 & \text{if route for EV } n \text{ has been completed.}
\end{cases}
$$

Given an initial route $r$, a 2-interchange swapping algorithm works by interchanging two links in the route with two other links. For a given route $r$ and a proposed swap $(i, j)$, a new route $r_{\text{new}}$ can be constructed by substituting two links. Link $i \rightarrow i + 1$ is substituted with $i \rightarrow j$, while $j \rightarrow j + 1$ is substituted.
Algorithm 5: GreedyProcedure(\mathcal{X}) : Greedy algorithm for finding initial shuttle route

Input: \mathcal{X} = \text{set of EV routes}, q, y = \text{current number of drivers}, \mathcal{V} \leftarrow \{\}

Output: r = [r[1], r[2], \cdots, r[\text{end}]]

1. j \leftarrow 1;
2. r[j] \leftarrow 0, y \leftarrow q;
3. j \leftarrow j + 1;
4. foreach \ x \in \mathcal{X} do
   5. while y \geq 1 do
      6. \hspace{1em} r[j] \leftarrow \{S(x)\};
      7. \hspace{1em} y \leftarrow y - 1;
      8. \hspace{1em} j \leftarrow j + 1;
      9. while y \leq q - 1 do
         10. \hspace{1em} r[j] \leftarrow \{C(x)\};
         11. \hspace{1em} y \leftarrow y + 1;
         12. \hspace{1em} j \leftarrow j + 1;
   13. foreach \ x \in \mathcal{X} do
      14. while y \geq 1 do
         15. \hspace{1em} r[j] \leftarrow \{C^d(x)\};
         16. \hspace{1em} y \leftarrow y - 1;
         17. \hspace{1em} j \leftarrow j + 1;
      18. while y \leq q - 1 do
         19. \hspace{1em} r[j] \leftarrow \{D(x)\};
         20. \hspace{1em} y \leftarrow y + 1;
         21. \hspace{1em} j \leftarrow j + 1;
   22. end \leftarrow j;
   23. r[\text{end}] \leftarrow 0;

with \(i + 1 \to j + 1\). Since direction of segment \((i + 1 \to \cdots \to j)\) is now reversed, it is necessary to check precedent feasibility. The shuttle also picks up and drops off the drivers at each node it visits. Therefore, a proposed 2-interchange must also be feasible in terms of number of drivers on the shuttle. Furthermore, a proposed 2-interchange must also improve (decrease) the length of the shuttle route. Given a route \(r\) and a proposed interchange \((i, j)\), we next describe the steps for PrecedenceFeasibilityCheck(\(r_{\text{new}}\)), CapacityFeasibilityCheck(\(r_{\text{new}}\) and RouteImprovement(\(r, r_{\text{new}}\)). Let us also consider a small example to illustrate the steps of the 2-interchange procedure. We consider 4 EVs with the following paths: 11 \to 2, 40 \to 15, 23 \to 16 \to 47 \to 4 and 24 \to 13 \to 46 \to 12. EVs 1 and 2 have direct paths while EVs 3 and 4 visit intermediate charging stations.

GreedyProcedure(\mathcal{X}):

Given a set of EV paths, we use the procedure described in Algorithm[5] to construct an initial route \(r\). The initial shuttle route for the example with 4 EV routes is shown in Table[A.1]
PrecedenceFeasibilityCheck($r_{new}$):

For a given shuttle route to be precedence feasible, the supplier of each EV route must be visited before the charger, which in turn must be visited before the dummy charger and finally the dummy charger must be visited before the demander. Therefore, for each EV $n$, there are three precedence constraints. We can use the matrix $m(n,i)$ to ensure the feasibility of all three precedence constraints. Given a proposed 2-interchange $(i,j)$, assume that for an EV $n$, $m(n,i) = 5$ and $m(n,j) = 3$. This implies that the supplier for EV $n$ has not been visited at node $i$ but the charger has already been visited before node $j$. It follows then that the supplier and charger nodes for EV $n$ lie within the shuttle route segment $(i + 1 \rightarrow \cdots \rightarrow j)$. Since performing the 2-interchange $(i,j)$ will reverse the direction of traversal on segment $(i + 1 \rightarrow \cdots \rightarrow j)$, the charger node will be visited before the supplier node. This will violate the first precedence constraint for EV $n$. If $m(n,i) = 4$ and $m(n,j) = 2$, the proposed 2-interchange $(i,j)$ will result in the dummy charger node being visited before the charger node of an EV $n$, violating the second precedence feasibility constraint.

Normally, performing the feasibility check will require $O(N^3)$ time since we must perform $O(N)$ checks for each 2-interchange. However, the computational complexity can be reduced to $O(N^2)$ by performing a customized version of screening procedure described in [Psaraftis, 1983]. For a given tour $r$ and a given stop $i$ ($1 \leq i \leq M - 2$), we define FIRSTSTOP($i$) to be the position of the first stop (charger) beyond node $(i+1)$ for which the corresponding EV supplier has not been visited including and up to node $i$. If no such stop exists, FIRSTSTOP($i$) = $M$. Similarly, we define SECONDSTOP($i$) to be the position of the second stop (dummy charger) beyond node $(i+1)$ for which the corresponding EV charger has not been visited including and up to node $i$. If no such stop exists, SECONDSTOP($i$) = $M$. Finally, we define THIRDSTOP($i$) to be the position of the third stop (demander) beyond node $(i+1)$ for which the corresponding EV dummy-charger has not been visited including and up to node $i$. If no such stop exists, THIRDSTOP($i$) = $M$. Mathematically, FIRSTSTOP($i$) = $h$ if $h$ is the smallest number above $(i+1)$ for which there exists an EV $n$ so that $m(n,i) = 5$ and $m(n,h) = 3$. If no such EV exists, the $h = M$. Similarly, SECONDSTOP($i$) = $h$ if $h$ is the smallest number above $(i+1)$ for which there exists an EV $n$ so that $m(n,i) = 4$ and $m(n,h) = 2$. If no such EV exists, the $h = M$. Finally, THIRDSTOP($i$) = $h$ if $h$ is the smallest number above $(i+1)$ for which there exists an EV $n$ so that $m(n,i) = 3$ and $m(n,h) = 1$. If no such EV exists, the $h = M$. Theorem A.1 describes the test for precedence feasibility check for a proposed 2-interchange $(i,j)$.

\[1\]

**Theorem A.1.** The exchange $(i,j)$ is precedence feasible if and only if $j < \text{FIRSTSTOP}(i)$ and $j < \text{SECONDSTOP}(i)$ and $j < \text{THIRDSTOP}(i)$.

The screening part of the 2-interchange algorithm can be summarized in the following algorithm. The values for $m(n,i), \hat{S}_i, \text{FIRSTSTOP}(i), \text{SECONDSTOP}(i), \text{THIRDSTOP}(i)$, and the matrix $T_F(i,j)$ for the small example are calculated in Table A.1

\[1\] This theorem is similar to the result of [Psaraftis, 1983]. The proof is obvious and hence omitted.
Algorithm 6: Screening Procedure for Prerecedence Feasibility Check for a Proposed 2-Interchange

| Input: FIRSTSTOP(i), SECONDDSTOP(i), THIRDDSTOP(i) |
| Output: TF(i,j) |
| for i ← 1 to M − 2 do |
| for j ← i + 1 to M do |
| if j < FIRSTSTOP(i) and j < SECONDDSTOP(i) and j < THIRDDSTOP(i) then |
| TF(i,j) ← ‘true’ ; |
| else |
| TF(i,j) ← ‘false’ ; |

We use a depth first procedure to select the best interchange. The TF(i,j) matrix checks for precedence feasibility for all possible (i,j) swaps given an initial route r. The precedence feasible swaps are shown as T in the table. We next apply capacity feasibility check and improvement check on this set of routes.

CapacityFeasibilityCheck(rnew):

The capacity feasibility check is done to ensure that the proposed 2-interchange does not cause the number of drivers to drop below zero or jump above the capacity q. A given shuttle route r begins at the depot with q drivers. A driver is dropped at each supplier and dummy charger node and one is picked from each charger and demander node. Given a proposed 2-interchange (i,j), and the new route rnew, the capacity only changes for the route segment (j → ⋯ → i + 1). Given yold, i.e., the number of drivers at node i on the current shuttle route r, one can easily determine ynew, i.e., the number of drivers at node i for new shuttle route rnew. If for a given node i on shuttle route, ynew falls below 0 or above capacity q, the proposed interchange (i,j) is deemed infeasible. For example, the proposed interchange (7, 10) is capacity infeasible as shown in Table A.2

RouteImprovement(r, rnew):

A proposed two interchange (i, j) involves substitution of two links (i, i + 1) and (j, j + 1) with two new links (i, j) and (i + 1, j + 1). In traditional TSP and DARP problems, an interchange is considered to be improving or favorable if \( t_{i,i+1} + t_{j,j+1} > t_{i,j} + t_{i+1,j+1} \). The improvement for an interchange (i, j) can simply be calculated as \( t_{i,i+1} + t_{j,j+1} - (t_{i,j} + t_{i+1,j+1}) \). The updated arrival times \( \tau_j \) for the new route can be calculated by iteratively adding the link traversal times, i.e., \( \tau_1 = 0, \tau_j = \tau_i + t_{ij}, \forall (i,j) \). In case of the synchronized EV relocation and shuttle routing problem presented in this paper, the improvement check is computationally more challenging. The arrival time of a shuttle at a demander node on an EV path p ∈ X depends on both the EV path length \( l_p \) and the shuttle arrival time on corresponding supplier node. This dependency is given by Equations (17) and (18). Therefore for a shuttle moving on link (i, j) the arrival
<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>**</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>C</td>
<td>C</td>
<td>DC</td>
<td>DC</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>**</td>
</tr>
<tr>
<td>Node, r[i]</td>
<td>0</td>
<td>11</td>
<td>40</td>
<td>23</td>
<td>24</td>
<td>16</td>
<td>13</td>
<td>47</td>
<td>46</td>
<td>2</td>
<td>15</td>
<td>4</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{S}_i$</td>
<td>0</td>
<td>1$^+$</td>
<td>2$^+$</td>
<td>3$^+$</td>
<td>4$^+$</td>
<td>3$&gt;$</td>
<td>4$&gt;$</td>
<td>3$&gt;$</td>
<td>4$&lt;$</td>
<td>1$^-$</td>
<td>2$^-$</td>
<td>3$^-$</td>
<td>4$^-$</td>
<td>0</td>
</tr>
<tr>
<td>$m(1, i)$</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$m(2, i)$</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$m(3, i)$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$m(4, i)$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FIRSTSTOP[i]</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>SECONDSTOP[i]</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>THIRDSTOP[i]</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>12</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Table A.1 – An Example Illustrating the Screening Procedure for Precedence Feasibility Check

Time $\tau_j$ at node $j$ is calculated as:

$$\tau_j = \begin{cases} 
\max(\tau_i + t_{ij}, \tau_l + l_p) & \text{if } p \in \mathcal{X}, l = S(p), j = D(p) \\
\tau_i + t_{ij} & \text{otherwise}.
\end{cases}$$

Since a proposed interchange $(i, j)$ reverses the shuttle traversal on segment $(i + 1 \rightarrow \cdots \rightarrow j)$, a demander node contained in the segment may have its arrival time changed from $\tau_{ij} + t_{ij}$ to $\tau_{ij} + l_p$ or vice
versa, owing to the change in its position. Similarly, a supplier node \( l' \) contained in the segment may have its arrival time changed due to change in its position on the route. If the supplier node belongs to path \( p \in \mathcal{X} \), this change in \( \tau'_l \) can also nonlinearly impact the shuttle arrival at the downstream demander node \( j' = D(p) \) and at any nodes after \( j' \) on the route. Therefore, improvement after a proposed 2-interchange can only be checked by fully calculating the shuttle arrival times at all nodes using the expression for \( \tau_j \) given above. Let \( \tau_{old}^M \) be the total route time for a given route \( r \). Let \( \tau_{new}^M \) be the total route time for the route \( r_{new} \). The improvement \( \Delta_{ij} \) can be calculated as: \( \Delta_{ij} = \tau_{new}^M - \tau_{old}^M \).

**B Additional Figures**

![Figure B.1](https://ssrn.com/abstract=3480725)

Figure B.1 – For a Small Network with 10 EVs and given a certain number of personnel and per unit costs, it is advantageous to increase the number of drivers.