C2SMART

CONNECTED CITIES WITH SMART TRANSPORTATION

A USDOT University Transportation Center

New York University

Rutgers University

University of Washington

University of Texas at El Paso

The City College of New York

Dual Rebalancing Strategies for Electric Vehicle Carsharing Operations

May 2019

Dual Rebalancing Strategies for Electric Vehicle Carsharing Operations

Joseph Y.J. Chow
New York University

Saif Jabari
New York University

Theodoros Pantelidis
New York University

Li Li
New York University

C2SMART Center is a USDOT Tier 1 University Transportation Center taking on some of today's most pressing urban mobility challenges. Using cities as living laboratories, the center examines transportation problems and field tests novel solutions that draw on unprecedented recent advances in communication and smart technologies. Its research activities are focused on three key areas: Urban Mobility and Connected Citizens; Urban Analytics for Smart Cities; and Resilient, Secure and Smart Transportation Infrastructure.

Some of the key areas C2SMART is focusing on include:

Disruptive Technologies

We are developing innovative solutions that focus on emerging disruptive technologies and their impacts on transportation systems. Our aim is to accelerate technology transfer from the research phase to the real world.

Unconventional Big Data Applications

C2SMART is working to make it possible to safely share data from field tests and non-traditional sensing technologies so that decision-makers can address a wide range of urban mobility problems with the best information available to them.

Impactful Engagement

The center aims to overcome institutional barriers to innovation and hear and meet the needs of city and state stakeholders, including government agencies, policy makers, the private sector, non-profit organizations, and entrepreneurs.

Forward-thinking Training and Development

As an academic institution, we are dedicated to training the workforce of tomorrow to deal with new mobility problems in ways that are not covered in existing transportation curricula.

Led by the New York University Tandon School of Engineering, C2SMART is a consortium of five leading research universities, including Rutgers University, University of Washington, the University of Texas at El Paso, and The City College of New York.

c2smart.engineering.nyu.edu

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Acknowledgements

Data of carshare operations in Brooklyn were shared by BMW ReachNow for this project, for which the project team is grateful. The initial model development for the electric vehicle carshare rebalancing involved collaboration with Dr. Taiyu Ma from the Luxembourg Institute of Socio-Economic Research (LISER). Milos Balac, a PhD student supervised by Professor Kay Axhausen at ETH Zurich, helped look at the BMW data when he stayed at NYU in summer of 2018. Zhexi Fu, a current PhD student at NYU, helped with heuristic algorithm development.

Executive Summary

The viability of electric vehicle car sharing operations depends on rebalancing algorithms. Earlier methods in the literature suggest a trend toward Markovian stochastic demand with server relocation with queueing constraints. It becomes quite clear that EV car sharing systems present a more complex environment in terms of mathematical modeling and decision analysis. The main goal of our study is to propose a new mathematical model for rebalancing carsharing vehicles over time that: 1) captures Markovian queueing in customer demand and 2) allows customers to pick up vehicles at different charge levels and at different locations further than their own "zone". We propose a new model formulation based on a node-charge graph structure that extends the relocation model to include transshipment relocation flows. The problem is formulated as a p-median facility location problem embedded with a capacitated commodity flow problem. The idle vehicle relocation problem in an electric carsharing system is formulated as follows. Rebalancing idle vehicles is making tradeoffs between serving customer demand at higher access cost, increasing relocation cost to reduce access cost, and increasing charging cost to cover demand for more highly charged vehicles at the expense of the vehicles' availability. From this aspect we identify two conflicting objectives: the first one is user accessibility, which is expressed by the link between demand at a node and the closest available idle vehicle, and the second is the cost of operations that arises from rebalancing vehicles to serve user demand.

Computational tests on a range of random instances with up to $|N| = 1000$ and $|H| = 4$ are solved using exact algorithms from commercial software (MATLAB). The solution time reaches up to 7724 seconds with an Intel i5-6300U CPU with 2 cores and 8GB memory. Such results show that there is a need to decrease solution times for rebalancing decisions. In order to provide a computationally efficient rebalancing system for large networks, we propose a heuristic algorithm for the p-median dynamic server relocation problem. To measure the effectiveness of our algorithm, we conduct a large test-case simulation experiment based on data obtained from the BMW ReachNow car-sharing operations in Brooklyn, New York. Our heuristic solves such instances in less than 200 seconds. Our model is then tested in a custom-built simulation environment using MATLAB to replicate the operations of a real car-sharing system for one month of operations.

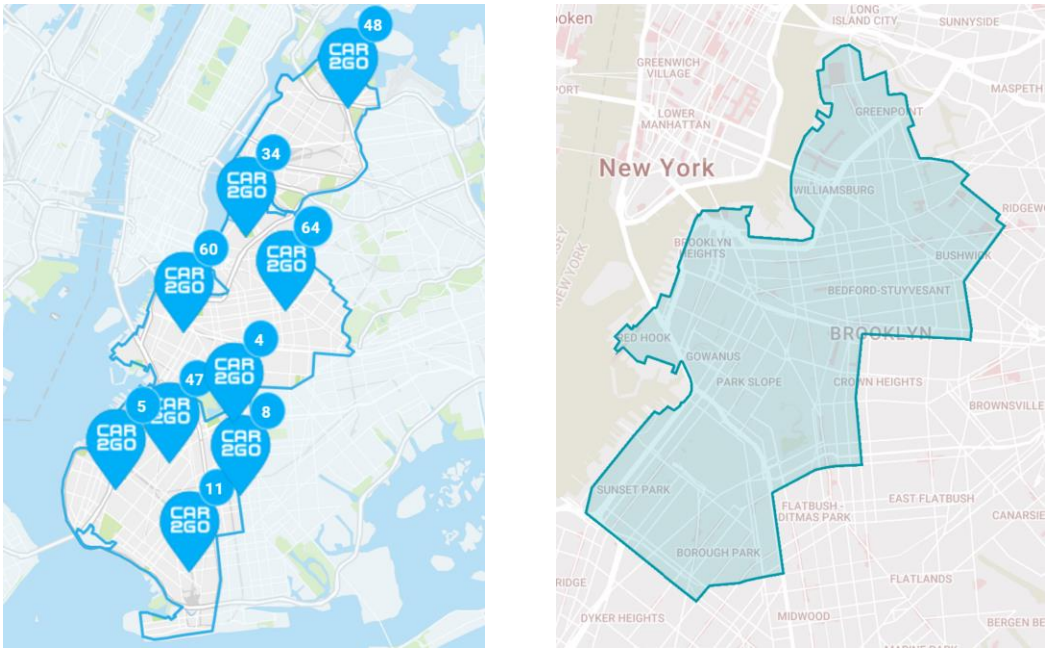
Table of Contents

List of Figures

List of Tables

Section 1: Introduction

Car sharing operations are an essential part of “smart mobility” solutions in congested large megacities. According to (Martin and Shaheen, 2016) a single carshare vehicle can replace 7 to 11 personal vehicles on the road, or between 5 to 20 vehicles by other accounts (Navigant Research, 2017). The common practice in such services is to book specific time slots and reserve a vehicle from a specific location. The return location is required to be the same for “two-way” systems but is relaxed for “one-way” systems. Within one-way systems, “station-based” systems restrict vehicles to specific parking locations while “free-floating” systems allow returns anywhere within a covered area. Examples of free-floating systems are the BMW ReachNow car sharing system in Brooklyn (until 2018) and Car2Go in New York City, with service areas in 2017 shown in Figure 1. Note that the two companies merged in 2018 (Bloomberg, 2018) and the ReachNow fleet is no longer in operation in Brooklyn. Car2go only launched in New York City in 2014, but by 2017 the number of members in NYC was more than 77,000. This makes NYC the fastest growing market in North America.



(a) **(b)**

FIGURE 1. Examples of free-floating carshare systems: (a) Car2Go (source: car2go.com) and (b) BMW ReachNow (source: reachnow.com).

In large car sharing systems, vehicle rebalancing is one of the primary challenges to ensuring efficiency and providing an adequate level of service. Potential customers may end up waiting or accessing a farther location, or even balk from using the service, if there is no available vehicle within proximity or no parking or return location available near the destination. Rebalancing involves having either the system staff or users (through incentives) drop off vehicles at locations that would better match supply to demand. Inefficient operations or underlying demand structures may cause systems to be shut down (Krok, 2016).

Car-sharing companies have further considered EV fleets to be more sustainable and to reduce gasoline consumption costs. Not only do Zipcar, Car2Go, and ReachNow all operate some EVs in their worldwide fleets, some startup carsharing businesses rely exclusively on EVs: e.g. Autolib' and Cité Lib in France, BlueIndy, DriveNow in Copenhagen, Carma in San Francisco, and Los Angeles' Low-Income Plan for EV car sharing (Lufkin, 2016). Car2go operates some electric vehicles (EV) in Europe (Stuttgart, Madrid and Amsterdam). With a total fleet of approximately 1,400 EVs, car2go is one of the largest providers in the electric vehicle carsharing sector.

Vehicle rebalancing efficiency is further hampered in an EV environment. Under such a setting, vehicles not only need to be rebalanced to meet stochastic customer demand, but they also need to be rebalanced to ensure sufficient charge level to meet future customer demand. As we will show in the literature review, methods to handle EV carshare rebalancing do not adequately address both sets of needs (demand and recharging).

Our study objective is to propose a new methodology to rebalance carsharing vehicles over time that: 1) captures Markovian queueing in customer demand and 2) allows customers to pick up vehicles at different charge levels and at different locations while explicitly considering the inconvenience of access for vehicles located further away. The model is structured as a p-median facility relocation problem with queueing constraints similar to Marianov and Serra (2002) and Sayarshad and Chow (2017), but with the following extensions: the relocation assignment component is expanded from a bipartite transportation problem into a single-commodity minimum cost flow (transshipment) assignment problem to handle recharging under a charge-space expanded network. By using this approach, we managed to reduce a multi-dimensional queuing problem into a two-dimensional location-relocation problem that can be repeatedly solved efficiently in a dynamic setting. An online heuristic algorithm is proposed to rebalance such EV carshare fleets over time. We implement an agent-based

simulation to test the algorithm against benchmark algorithms using demand data of BMW ReachNow carshare operations in Brooklyn in 2017 that the company provided.

This research provides mobility operators with an algorithm that may make it possible to operate EV carshare (or other similar Mobility-as-a-Service systems) that require both rebalancing and recharging under limited charging infrastructure. At a time when cities demand more sustainable solutions, but operators are concerned with the higher costs of operating EV fleets, this work may be a catalyst toward more sustainable smart cities.

Section 2: Literature Review

Early studies for designing car-sharing systems relied on simulation (Barth and Todd, 1999) for evaluation. More systematic mathematical models to optimize car-sharing fleets have since been proposed with rebalancing in mind. The purpose of optimal rebalancing is to make the decision considering trade-offs among a set of parameters that differ from one instance to another:

- Customer demand for vehicles over time, with distribution for pickup and drop-off locations
- Vehicle access distance for customers, which should not be restricted to a specific "zone"
- Penalty for having insufficient vehicle stock to meet demand
- Cost of relocation
- Distribution of booking duration, which may exceed one direct trip from pickup to drop-off – for example, a customer may take a vehicle out of town to run errands before returning the vehicle at another location in the coverage area
- Rental cost over time
- Distribution of remaining fuel upon return of vehicle
- For station-based one-way systems, capacity of spaces in each zone is a factor
- Reservation time requirement and availability of information to the system

In addition, for EV systems there are additional trade-offs:

- Cost of charging EVs
- Duration of EV charging
- Demand for EVs at a minimum charge level
- Distribution of charge consumed upon return of vehicle
- Capacity of charging stations, i.e. number of chargers

Many studies have been devoted to the rebalancing problem (Fan et al., 2008; Kek et al., 2009; Nair and Miller-Hooks, 2011; Nourinejad et al., 2015; Sayarshad and Chow, 2017) and related regulating challenges: station location with rebalancing (Chow and Sayarshad, 2014); pricing incentives (Clemente et al., 2014; Jorge et al., 2015; Waserhole and Jost, 2016); parking reservations (Kaspi et al., 2016); routing personnel (Bruglieri et al., 2014; Nourinejad et al., 2015); parking capacity, fleet sizing (Hu and Liu, 2016), etc.

Less focus has been given to methods far rebalancing electric vehicle (EV) carshare systems, however. A smaller subset of studies emerged in recent years to tackle this heightened challenge. Two general methods have been adopted. The first assumes demand is sufficiently deterministic in a multiperiod setting (e.g. Xu et al., 2018). This can be problematic for systems in which most of the demand is not made for repeated commute trips and/or the fleet is spatially distributed in such a manner that the vehicle density is fairly low, which is most systems. The second group of methods assumes stochastic demand, either through stochastic programming (Brandstätter et al., 2017), simulation (Boyacı et al., 2015), or with Markovian demand (Li et al., 2016). The latter Markovian queueing models appear promising, but earlier EV studies either assume a simplistic relocation policy (Li *et al.*, 2016) or, in the case of queueing networks (Waserhole and Jost, 2016), do not allow customers to pick up vehicles at nearby locations. Discrete network approaches have not been used with queueing models for EV charging settings either. It is quite clear that EV car sharing systems present a more complex environment in terms of mathematical modeling and decision analysis.

The evolution of car-sharing rebalancing models can be broken down into four general stages with increasing ability to address key trade-offs: deterministic demand assignment, stochastic demand, queueing network, and queueing-based facility location. Simulation methods are left out as we are interested in analytical solutions. We summarize some example representative studies (non-EV and EV) as well as the trade-offs that they handle well or poorly. Note that they do not necessarily follow a chronological order as different methods from different stages have been used for different applications.

Deterministic demand assignment: these are characterized by network models, sometimes with multiple time periods to capture capacity costs in terms of delay costs for customers.

- Non-EV studies: Di Febbraro et al. (2012), Sayarshad and Chow (2017), Jorge et al. (2015), Kaspi et al. (2016)
- EV studies: Boyacı et al. (2015), Xu et al. (2018)
- Weaknesses: does not handle random demand or its consequences like stockout/balking penalties, information availability to the system, and impacts of charging station capacities

Stochastic demand: these models consider stochastic representations of the demand, in some cases involving Markovian demand with one-step look-ahead via two-stage stochastic programming or chance constraints

- Non-EV studies: Nair and Miller-Hooks (2011)
- EV studies: Brandstätter et al. (2017)
- Weaknesses: while demand is random, the decisions are generally myopic (ignores dependency of future decisions on current decisions) and reliant on simulation-based optimization for capturing all the trade-offs

Queueing network: these models consider the steady state impacts of rebalancing decisions under a stochastic environment, which are more dynamically stable

- Non-EV studies: Waserhole and Jost (2016), Zhang and Pavone (2016)
- EV studies: none published yet
- Weaknesses: the restriction to have demand be served only at the specified origin/destination zones assumes customers cannot be served by vehicles at nearby zones, which is not realistic for one-way systems, particularly free-floating carshare systems; for EV-based systems the queueing dynamics are more complex, which is why no methods have been proposed yet

Queueing-based facility location: these are characterized by facility location models that incorporate queueing criteria in the objectives or constraints, and allow for interzonal access costs

- Non-EV studies: Sayarshad and Chow (2017)
- EV studies: none published yet
- Weaknesses: see below

Facility location can consider queue delay by defining each service node as a queue with a number of s servers. In this case, however, even a simple assumption of an M/M/s stochastic queue results in a nonlinear objective. Since nonlinear integer programming problems are undesirable, researchers have proposed alternative methods to handle the queueing. One such way is the Q-MALP model from Marianov and ReVelle (1996), who showed that the queue delay objective can instead be cast as a set of piecewise linear constraints for the intensity to be within a specified reliability level η. Because the intensity parameter can be preprocessed for different numbers of servers, it is possible to solve a facility location with desired queueing-based service reliability as a mixed integer linear programming problem. The model has since been modified to handle maximal coverage (Marianov and Serra, 1998), server allocation (Marianov and Serra, 2002), and p-median coverage with relocation costs (Sayarshad and Chow, 2017).

Queueing-based facility location models handle everything that the "Queueing network" models can, and on top of that they allow for inter-zonal matching of vehicles to demand. However, the relocation component in Sayarshad and Chow (2017) is based on a bipartite transportation problem of moving excess servers to locations in demand of servers. This is fine for a non-EV carsharing system, but for EV charging the mechanics are more complex (see Jung et al., 2014). The model does not distinguish demand for a minimum charge level that can be less than 100%. For example, a customer should be allowed to demand a vehicle with at least 60% charge, and this same customer should be allowed to pick-up a vehicle with 80% charge.

We address this issue by first proposing a node-charge expanded network representation of the location problem for EV carshare fleets. Under this representation, a network is replicated into multiple charge levels and movement from one charge level up to another represents recharging activity. Under this representation, we further propose the first queueing-based facility relocation problem with minimum cost flow relocation, where a vehicle may start at a lower charge level, be repositioned at a charging station, recharge up several charge levels, and then be matched to customers at nearby zones. This model is formulated as a mixed integer programming problem. For larger cases we propose a novel heuristic algorithm to solve the MILP that ensures three dimensions of feasibility of intermediate solutions: coverage, queueing intensity, and capacity.

Section 3: Proposed Model

3.1 Node-graph Structure

Before diving into the model formulation of the EV rebalancing problem, we need to modify the conventional graph structure such that the trade-offs can all be addressed. A network $G(N, A)$ of zones N is connected by links A. A subset of these zones is designated as a set of charging stations $J \subset N$ with finite numbers of chargers, $u_j, j \in J$. At the start of every time interval t, there is a set of idle vehicles F that have not been reserved by any customer. These vehicles may either be sitting somewhere in a zone unused, charging at a charging station, or being relocated to another zone or charging station. The locations and charge levels of the vehicles are known. Customers arrive randomly within that time interval according to a stationary (within that interval) Poisson process. When they book the vehicles for use, the vehicles are effectively "serving" the customers for a period of time that is assumed to follow an exponential distribution. Customer arrivals and vehicle return zone locations are assumed to follow discrete distributions.

We can graphically illustrate this as a one-dimensional network without loss of generality. Consider a 5-node network lined up in sequence as shown in Figure 2(a) where node 1 and node 3 are charging stations (denoted by gray nodes).

The graph is converted into a node-charge graph representation in Figure 2(b) where the y-axis is a discrete charge interval (let's say there are four intervals: 20%+, 40%+, 60%+, 80%+). Each layer represents the same zones at a certain charge level. Unidirectional links exist at the charging stations to represent charging with link costs representing charging cost and time. The charging links are also capacitated. A vehicle positioned at a node covers ALL nodes underneath it with lower charge as well; access costs for demand is based only on the spatial link costs and not the charging link costs. For example, a vehicle at node 4 with charge 40+ (node-charge 9) can serve nodes 1 to 5 at 40%+ and also at 20%+, as illustrated by the purple arrows in Figure 2(b). The access cost of demand at (node 2, charge 20+) for the vehicle at (node 4, charge 40+) is just the cost from node 2 to node 4.

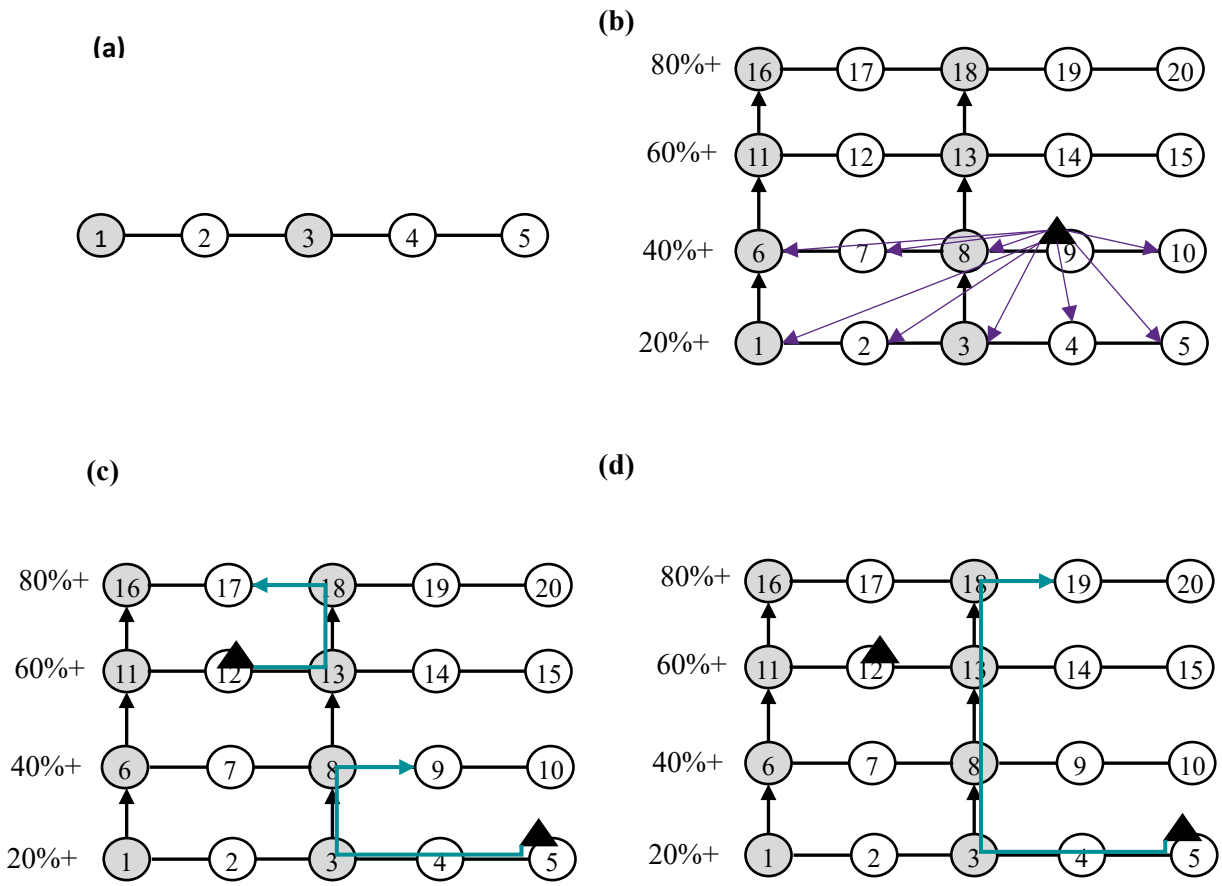


FIGURE 2. (a) initial graph; (b) expansion to a node-charge graph with coverage illustration in blue arrows; (c) one possible rebalancing solution in red arrow; and (d) a second rebalancing solution.

Suppose there are two idle vehicles, one with 20% charge at node 5 and one with 60% charge at node 2. Two feasible rebalancing solutions are shown in Figure 2(c) and 2(d). In Figure 2(c), one vehicle is directed toward station 3 to recharge 20% before being relocated back to node 2, while the second vehicle is sent to station 3 to recharge 20% before being relocated to node 4. In Figure 2(d), only one vehicle is relocated to station 3 to be recharged up to 80%+ and would then be relocated to node 4. The two solutions lead to different coverage results and different charging capacity requirements (Figure 2(c) requires a capacity of two chargers).

The optimality of these solutions depends on a mix of factors. For example, Figure 2(c) may be best if there is high demand near nodes 1, 2, and 3 up to 80%+ charge and there is enough capacity at station 3 to allow two vehicles to charge at the same time. The charging cost might be very high relative to the spatial relocation and/or access costs/penalties, leading to two

short charges instead of one longer charge. Figure 2(d) may be best if the high charge demand is located closer to nodes 4 and 5, and/or perhaps there is only enough capacity for 1 vehicle charging at station 3 and the relocation cost to station 1 does not warrant the additional charging.

This graph structure is highly flexible. For example, it can be set to 100 discrete charge levels. Duplicates of locations can be used (e.g. a zone 3 with a zone "3a" for charging stations at zone 3) to model the decisions to remove a charged vehicle from a station without having to rebalance it to another zone. The bottom charge level can represent minimum charge needed to get from any zone to any other zone without depleting fully.

Under this graph structure, we model the relocation decision such that demand at one or more charge levels are assigned to each zone, where the service at the zone acts as a stochastic queue with one or more servers.

3.2 Model Formulation

The following notation leaves out the time interval t for convenience since they all apply to the same interval.

N: set of nodes

H: set of charging levels, $h = 1,2, \ldots$

A: set of directed arcs in the node-charge graph, $A = \{((i,g),(j,h)) | \forall i,j \in N, g,h \in H\}$

B: total number of idle vehicles at the start of a rebalancing time interval

y_{ig}: number of idle vehicles at the node-charge (i,g) at the start of a rebalancing time interval

$J \subset N$: subset of nodes that are charging stations

θ_{ij}: parameter for cost of charge to get from node i to node j (rounded to nearest $h\varphi$)

$u_j, j \in J$: capacity of charging facilities available

c_{igjh}: cost on an arc of node-charge graph from (i, g) to (j, h)

O: set of origins of idle vehicles on the node-charge graph

λ_{ih}: arrival rate of customers at node i with demand for SOC h intervals or higher; assume that customers use up exactly h intervals during their trip;

C: max number of servers at a node-charge

F: set of idle vehicles at the beginning of idle vehicle relocation epoch

M: large positive penalty constant

A_{ig}^{+}: set of outgoing arcs originating at a node-charge $(i, g), i \in N, g \in H$

A_{ig}^{-}: set of incoming arcs destinated to a node-charge $(i, g), i \in N, g \in H$

W_{jgjh}: rebalancing EV flow on arc $((i, g), (j, h)) \in A$,

$\boldsymbol{Y_{jhm}}$: if the m-th vehicle are located at node j with charge h

$\boldsymbol{X_{igjh}}$: vehicle at node j with charge level h serves customer demanding $g \in \{1, \dots, |H|\}$ charge or more at node i if $X_{igjh} = 1$

Parameters should be defined such that $\lambda_{idh} = 0$ if $c_{id} > h\varphi$, where φ is the amount of charge in one interval. The objective of the problem is to minimize total customer's cost and the operator's weighted rebalancing cost under customer stochastic demand and charging station capacity constraints. One particular challenge is that the capacity is not link-level but node level (u_j). It doesn't matter which level the charging takes place; the capacity is on the number of vehicles ascending any part of the charging station.

The problem is an extension of a multiple server relocation problem under stochastic demand (Sayarshad and Chow, 2017) in the context of electric vehicle charging. We consider the problem on a digraph with multi-levels (multilayers) that converts the EV-rebalancing problem into a facility location embedded with a single commodity minimum cost flow problem to rebalance idle vehicles to meet customer demand for different charging levels.

Let $G'(N', A')$ be a directed graph with N' being a set of node-charge $(i, h), \forall i \in N, h \in H$, and A' a set of directed arcs, $A' = \{((i, g), (j, h)) | \forall i, j \in N, \forall g, h \in H\}$. A node-charge (i, h) is characterized by demand's spatial location $i \in N$ and request charging level (battery level) $g \in H$. We discretize customer's charging demand into a set of charging levels *H*. All demand between two levels $h - 1$ and h sum up to level h. Arcs only cross from one charging level up to another at charging station nodes J and there is a capacity $u_j > 0, j \in J$ applied to all flows going through that station node regardless of charge level. Charging arcs belong to a subset of arcs defined as $A_{\hbar} = \{((j, g), (j, h)) | \forall j \in J, \forall h = g + 1, g = 1, \dots |H - 1|\}, A_{\hbar} \subset A$. The assigned flow on arcs is an integer decision variable. The cost of assigning flow on arcs is the multiplication of arc flows by its unitary cost, measured as rebalancing time/cost or charging time/cost on arcs.

Rebalancing idle vehicles involves making tradeoffs between serving customer demand at higher access cost, increasing relocation cost to reduce access cost, and increasing charging cost to cover demand for higher charged vehicles at the expense of another vehicle being allowed to charge. The problem is formulated as a p-median facility location problem embedded with a route-capacitated minimum cost flow problem. The idle vehicle relocation problem in an electric carsharing system is formulated as follows.

$$\min \mathrm{Z} = \sum_{i \in N} \sum_{j \in N} \sum_{h \in H} \sum_{g \in H} \lambda_{ig} t_{ij} X_{igjh} + \theta \sum_{((i,g),(j,h)) \in A} c_{igjh} W_{igjh} \tag{1}$$

s.t.

$$\sum_{j \in N} \sum_{h \in H, h \geq g} X_{igjh} = 1, \qquad \forall i \in N, g \in H \tag{2}$$

$$\sum_{j \in N} \sum_{h \in H, h < g} X_{igjh} = 0, \qquad \forall i \in N, g \in H \tag{3}$$

$$Y_{jhm} \leq Y_{jh,m-1}, \qquad \forall j \in N, h \in H, m = 2,3, \dots, C \tag{4}$$

$$\sum_{i\in N}\sum_{g\in H}\lambda_{ig}X_{igjh} \le \mu_{jh}\left[Y_{jh1}\rho_{\eta jh1} + \sum_{m=2}^{C} Y_{jhm}\left(\rho_{\eta jhm} - \rho_{\eta jh,m-1}\right)\right], \forall j \in N, h \in H \tag{5}$$

$$\sum_{j\in N}\sum_{h\in H}\sum_{m=1}^{C} Y_{jhm} = B \tag{6}$$

$$X_{igjh} \le Y_{jh1}\,, \qquad \forall i,j \in N, g,h \in H \tag{7}$$

$$\sum_{(j,h)\in A_{ig}^{-}} W_{igjh} - \sum_{(j,h)\in A_{ig}^{+}} W_{igjh} \le MY_{ig1}\,, \forall (i,g) \in N \times H \backslash O \tag{8}$$

$$-\left(\sum_{(j,h)\in A_{ig}^{-}} W_{igjh} - \sum_{(j,h)\in A_{ig}^{+}} W_{igjh}\right) \le MY_{ig1}\,, \forall (i,g) \in N \times H \backslash O \tag{9}$$

$$\sum_{(i,g)\in A_{jh}^{-}} W_{igjh} - \sum_{(j,h)\in A_{ig}^{+}} W_{igjh} + y_{ig} = \sum_{m=1}^{C} Y_{jhm}\,, \qquad \forall j \in N, h \in H \tag{10}$$

$$\sum_{g'\le g}\sum_{h'\ge h} p_{jg'jh'} = W_{igjh}\, \forall j \in J, g \in H, \forall ((j,g),(j,h)) \in A \tag{11}$$

$$\sum_{g=1}^{|H|-1}\sum_{h'=g+1}^{H} p_{jgjh'} \le u_j, \forall j \in J, g \in H, \forall ((j,g),(j,h)) \in A \tag{12}$$

$$X_{igjh} \in \{0,1\}, \forall i,j \in N, g,h \in H \tag{13}$$

$$Y_{jhm} \in \{0,1\}, \forall j \in N, h \in H, m = 1,2,3,\dots,C \tag{14}$$

$$W_{igjh} \in 0 \cup \mathrm{Z}^{+}, \forall i,j \in N, g,h \in H \tag{15}$$

$$p_{jgjh} \geq 0, \forall((j,g),(j,h)) \in P_j \quad (16)$$

The objective function minimizes the total access cost of customers to servers (idle vehicles) and total routing cost of idle vehicles (i.e. travel time/cost from the current locations of idle vehicles to charging stations, charging time/cost and travel time/cost to its respective destinations) on the node-charge graph (network). The rebalancing operations are run at each predefined time interval in order to serve customer's demand and minimize queueing delay and operating cost.

Constraints (2) and (3) require that rebalanced idle vehicles to serve randomly arriving customers must have sufficient charge to match the demanded amount. Constraint (4) is an order constraint stating a m-th server can be present only if there is already a (m-1)-th server at the same location.

Constraint (5) is the piecewise linear queueing constraint from Marianov and ReVelle (1996) queueing constraint representing that when a customer arrives at an idle vehicle, there will be no more than b other customers waiting on a line with a probability more than service reliability η. The intensity is setup as a recursive cumulative value based on the number of servers assigned to the location.

Constraint (6) states the total number of servers is equal to the total number of available idle vehicles. Constraint (7) assures that only a location with servers can cover demand nodes. Constraints (8-10) are the flow conservation constraints of the minimum cost flow problem. For the charging station capacity, we need to ensure that the assigned flow on charging arcs does not exceed the limit of chargers available on a charging station. This constraint should ensure that, for example, Figure 2(c) should not occur if $u_2 = 1$ because technically both vehicle flows are concurrently using that charging station. To address this, the link flows W_{igjh} are matched to enumerated path flows in constraint (11). There is one set of path flows for each charging station, so there are not many – for 4 charging levels there are 6 variables per charging station: e.g. {1-2, 1-3, 1-4, 2-3, 2-4, 3-4} where charge level 1 is lower than charge level 2. The path flows are used to ensure that path flow capacity is met in constraint (12). Lastly, X_{igjh} and Y_{jhm} are binary decision variables. Sayarshad and Chow (2017) showed that Y_{jhm} can generally be relaxed to a continuous variable between [0,1] since the piecewise linear constraint will generally be satisfied, which leads to a much more computationally efficient model. Arc flow

W_{igjh} is a non-negative integer decision variable of vehicle flow, and the path flows are continuous non-negative variables.

Note that in Eq. (5), $\rho_{\eta jm}$ is the coefficient of the utilization rate constraint, given a user-defined reliability rate η, m idle vehicles (servers) and *b* customers in a queue. The value of $\rho_{\eta jm}$ is obtained exogenously by solving the following Equation (17):

$$\sum_{k=0}^{m-1}((m-k)m!\,m^b/k!)\,(1/\rho^{m+b+1-k}) \geq 1/(1-\eta) \tag{17}$$

If the queueing constraint (Eq. (5)) is considered, the model represents a non-myopic case in which the relocation decisions are designed to minimize steady state demand access costs which include wait time due to unavailability of nearby vehicles. Otherwise, the model is a myopic case without anticipating a future queueing state in the system.

As a p-median problem, the model is NP-complete (see Owen and Daskin, 1998). Existing heuristics for p-median problems like Teitz and Bart (1968) are not directly applicable because they violate queueing intensity and capacity feasibility. We propose a new heuristic to solve this problem.

Section 4: Proposed Solution Algorithm

In order to provide a computationally efficient rebalancing system for large networks, we propose a heuristic algorithm for the p-median dynamic server relocation problem with route-capacitated minimum cost flow relocations. We are interested in developing a rebalancing system that can scale up to a fleet like the BMW ReachNow one in Brooklyn, NY. This network has 304 zone centroids and considers up to 8 charge levels. Computational tests on a range of random instances with up to $|N| = 1000$ and $|H| = 4$ are solved using exact algorithms from commercial software (MATLAB). The solution time reaches up to 7724 seconds (> 2 hours) with an Intel i5-6300U CPU with 2 cores and 8GB memory; this takes too long for practical implementation in an online setting.

We propose a heuristic that solves such instances in less than 200 seconds. The core of the algorithm is based on the greedy heuristic from Teitz and Bart (1968) but modified to maintain feasibility with respect to: (a) queueing constraints, (b) capacity, (c) minimum cost flow relocation, and (d) demand access savings when accounting for multiple servers. In a p-median problem without queuing constraints, each server can satisfy the entire demand from any location. In our model, however, a server cannot satisfy demand at higher layers, while demand at layers lower or equal to the server's location can be served up to the RHS amount of equation (5). For this reason, a node that has an idle car already can still yield potential gains of adding an additional server to it. A summary is provided in Algorithm 1.

Algorithm 1: Proposed heuristic

---

1. Initiate by solving the 1-median location problem and place all idle servers at the solution point.
2. Compute the number of servers that exceed capacity in the current solution.
3. While $excess > 0$ consider rebalancing only to the subset of nodes: (j, h): $h = g \in H$, where g is the layer of the server's initial location.
4. Let $k = k + 1$. Compute the objective value of adding one new potential facility to each node j based on maximizing savings from the initial solution.

5. If $\sum_{i\in N}\sum_{g\in H}\lambda_{ig} > \sum_{j\in N}\sum_{h\in H}\mu_{jh}\left[Y_{jh1}\rho_{\eta jh1} + \sum_{m=2}^{C} Y_{jhm}\left(\rho_{\eta jhm} - \rho_{\eta jh,m-1}\right)\right]$ then: $savings(j) = -\infty$
6. If a node is selected for rebalancing the node savings will be calculated as:

$savings(j)^{k+1} = \frac{gains(j)}{d(j)} savings(j)^{k}$, $where\ d(j) = \sum_{K} d(k)$ is the demand attracted to this node (closest)

$$and\ gains(j) = \begin{cases} \left(\rho_{jh,m+1} - \rho_{jh,m}\right)\mu_{jh} & if\ d(j) - \rho_{jh,m}\mu_{jh} > \left(\rho_{jh,m+1} - \rho_{jh,m}\right)\mu_{jh} \\ d(j) - \rho_{jh,m}\mu_{jh} & elsewhere \end{cases}$$

7. When $k = |F|$ (number of idle servers), stop.

Section 5: Numerical Study

Evaluation of the proposed model is conducted in two sets of replicable experiments. The first is with an illustrative example that serves to verify the model and demonstrate the capabilities to evaluate certain trade-offs. The second is with a set of 7 generated instances ranging in size from 10 to 1000 nodes, and 4 charging levels (up to 4,000 node-charges) to test the scalability of the model using commercial solvers. For context on what performance to expect, Zhao et al. (2018) developed a Lagrangian relaxation approach that could solve an electric vehicle routing problem about 5 times faster than exact methods with an 8% optimality gap for medium and large test case instances.

5.1 Illustrative Example

To illustrate the model, we consider a small network of 24 node-charges with 6 aligned nodes, extended up to 4 charging demand levels (20%, 40%, 60%, 80%) (see Figure 3). The goal is to test whether the proposed model can effectively rebalance the idle vehicles to meet all customer demand under available charging capacity constraints. The travel time between nodes is denoted as t_{ij} and the charging time (vertical travel distance) as c_{ij}. Three idle vehicles with respective remaining charge levels are located at node-charge 3, 7, and 14. We consider 2 charging stations (nodes 2 and 6) with the same capacity u per station. We test the model under three different capacities: $u = \{1,2,3\}$. Customer arrival rates are arbitrarily generated over all node-charges and fixed for the three scenarios. These rates are shown in the numbers over each node-charge (e.g. 3.8 customers/hr arrive at node-charge 15, which represents node 3 at charge level 60%+). It is worth mentioning that demand varies between different charging levels because the range required for specific trips can differ among travelers.

The model is implemented in MATLAB using a Dell Latitude E5470 laptop with win64 OS, Intel i5-6300U CPU, 2 Cores and 8GB memory. The MATLAB mixed-integer linear programming solver (intlinprog) is used to solve the optimization problems for this study. The test instances are publicly available on zenodo.

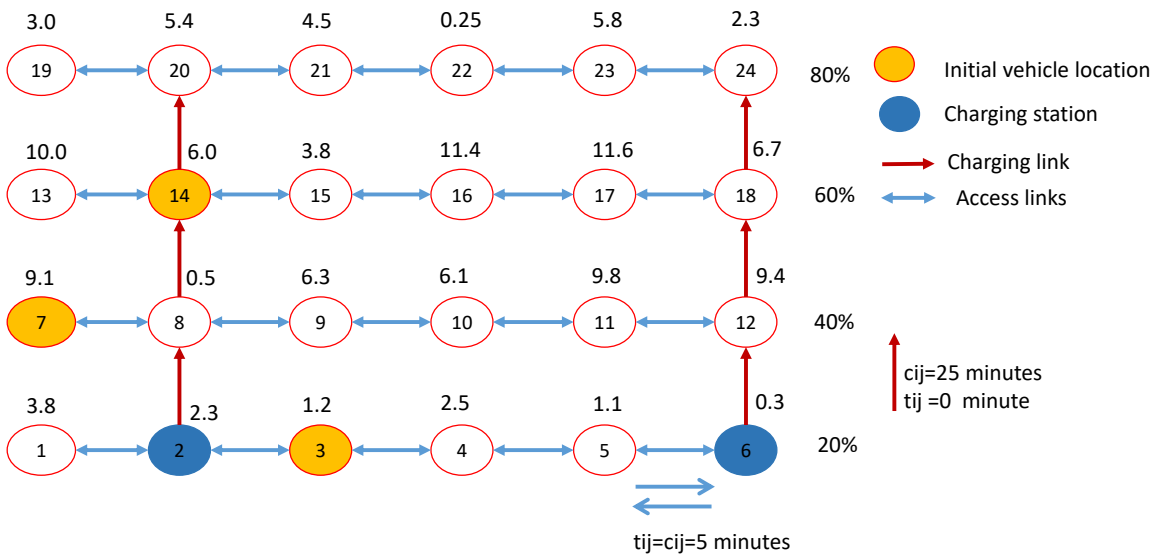


FIGURE 3. Illustration of a node-charge graph.

Figure 4 presents the computational result of rebalancing EV flows for the case of capacity $u = 3$. We see that all vehicles are rebalanced to the highest level 4 at different nodes which minimize total access cost of customers. The vehicles are assigned to use the nearest charging links to their destinations. The optimal objective value is $Z^* = 281.5$.

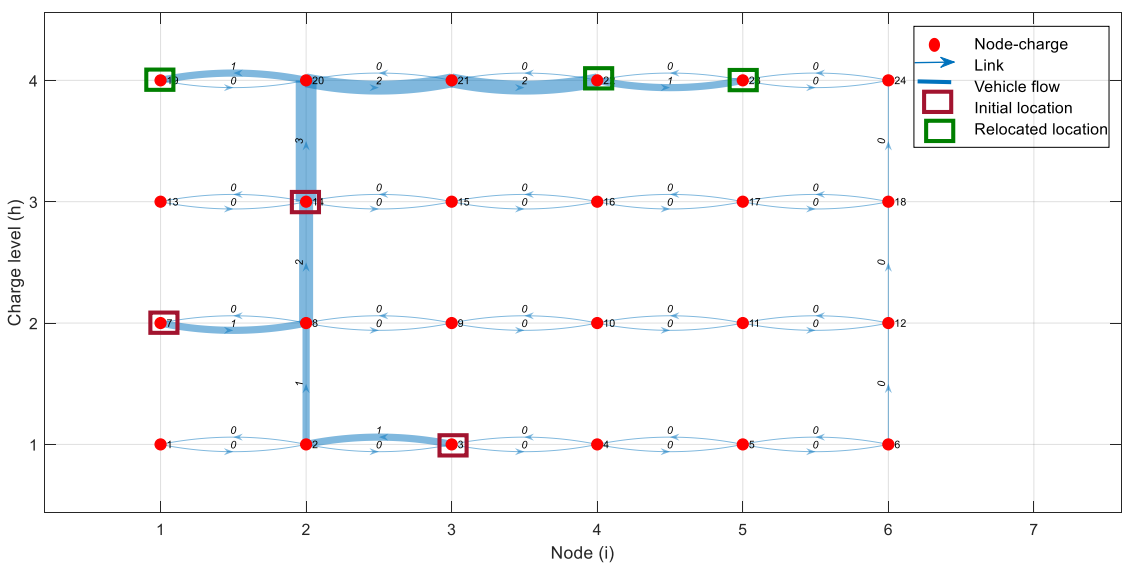
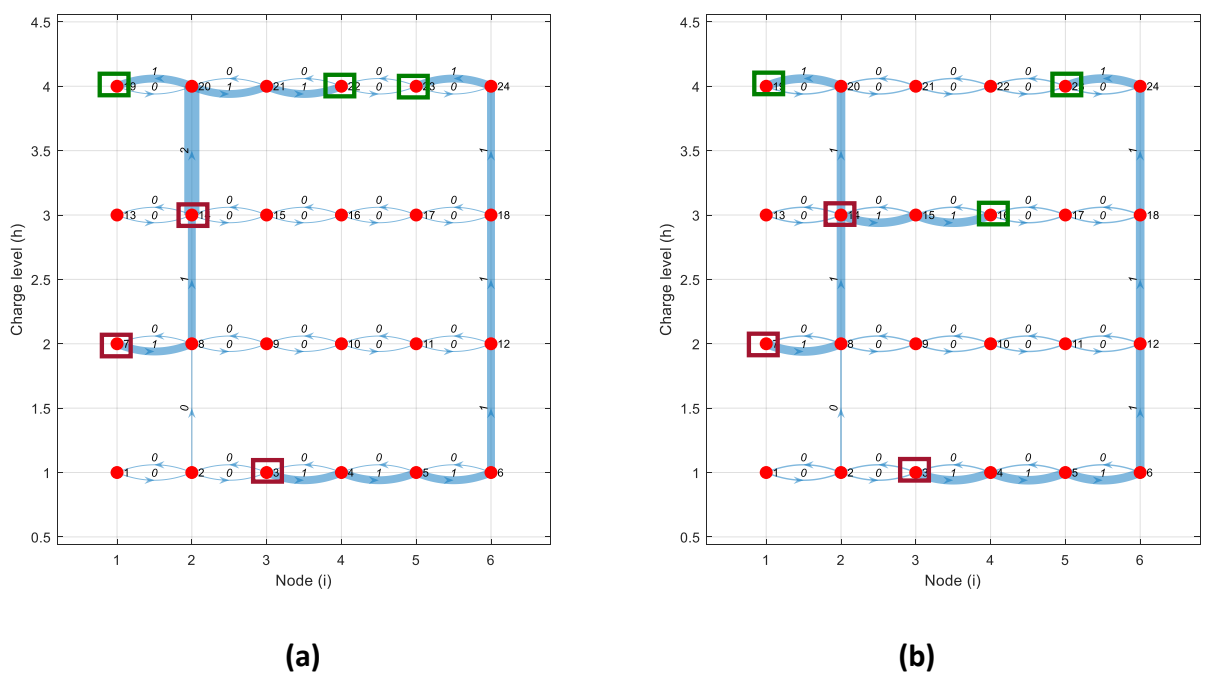


FIGURE 4. Rebalancing idle vehicle flows and its locations under charging station capacity = 3.

When reducing the capacity by 1 unit, we see the first charging station (i.e. node 2) is fully capacitated by vehicles at node 7 and 14. The vehicle at node 3 moves farther to charge at node 6 and comes up to its destination at node 23 (see Figure 5(a)). The numbers over links represent the assigned flow on links. The obtained $Z^* = 281.5$ is the same as the preceding case.

When further reducing the capacity to $u = 1$ per station, there are only two chargers available in the system. The obtained solution shows the vehicles at node 7 and 3 coming up to level 4 and the vehicle at node 14 is rebalanced on the same level to node 16. All charging capacity is used with least system cost (see Figure 5(b)). The obtained objective values is now $Z^* = 300.25$.

These results verify that the model formulation works.



(a) **(b)**

FIGURE 5. Rebalancing idle vehicle flows under charging station capacity (a) $u = 2$ and (b) $u = 1$.

5.2 Large Case Test Instances from 10 to 1000 Nodes

To test the performance of the proposed model in large networks, we provide 7 test instances ranging from 40 node-charges to 4000 node-charges (4 charging demand levels: 20%, 40%, 60%, and 80%). Customer demand at each node-charge is randomly generated from [0, 1]. The available idle vehicles are set as 10, and 4 charging stations with a capacity of 4 vehicles per

station are considered. The parameter setting is shown in Table 1. The sizes of the test problems are shown in Table 2. We see the numbers of decision variables and of constraints (Eq. (2) - (11)) increase exponentially (162k decision variables for the 400 node-charge instance and 16 million constraints for the 4000 node-charge case).

First, we solve the myopic EV rebalancing problem by relaxing the queueing constraint (Eq. 5). The goal is to explore the computational performance of existing solver under a general personal computer. We see our model formulation allows finding the optimal solution for the 400 node-charges case in 10 seconds, around 40 minutes for 1600 node-charge case, and 2.14 hours for a large network with 4000 node-charges. The result shows the performance of the model is quite fast and could be easily accelerated by using parallel computing on multithread computers for real-time operations.

TABLE 1 Reference parameter settings

Parameter	Value
|N|	10-1000
|H|	4
B and |O|	10
λ_{ig}	Random number drawn from [0,1] for each node-charge
θ	0.2
C	3
ρ_{jm}	(0.2236, 0.6416, 1.1576)
|J|	4
u_j	4
M	10000

TABLE 2 Problem instance and computational times to solve myopic EV idle vehicle rebalancing problem

N	H	$N \times H$	Num. of decision variables	Num. of equations	CPU time (sec.)
10	4	40	1 804	1 901	<1
50	4	200	41 004	41 581	2
100	4	400	162 004	163 181	10
200	4	800	644 004	646 381	196
400	4	1600	2 568 004	2 572 781	2610
800	4	3200	10 256 004	10 265 581	7222
1000	4	4000	16 020 004	16 031 981	7724

For the non-myopic case, the computational time depends on the complexity of Eq. (5), characterized by possible combinations of Y_{jhm} and its relation between λ and μ. To illustrate this point, we explore the computational time on the case with 200 node-charges by varying the service rate μ_{jh} from $0.1N$ to N with an incremental $0.1N$ while keeping the same parameter setting. The result shows the computational time could be 2420 times higher when introducing the queueing constraint. Over the 10 tested μ_{jh}, only the case with $\mu_{jh} = N$ obtains an optimal solution. No feasible solutions were found for the other cases. The results demonstrate why further research is necessary in studying fast heuristics to find approximate solutions for a large node-charge network.

TABLE 3 Influence of queueing constraints (eq. (5)) on computational time

Case	$N \times H$	$\sum_{ig} \lambda_{ig}$	μ_{jh}	cputime (sec.)	Z*
1	200	102.81	5	2	NA
2	200	102.81	10	2306	NA
3	200	102.81	15	2	NA
4	200	102.81	20	2	NA
5	200	102.81	25	6	NA
6	200	102.81	30	1816	NA
7	200	102.81	35	2600	NA
8	200	102.81	40	809	NA
9	200	102.81	45	4840	NA
10	200	102.81	50	22	831.206

Remark: NA means no feasible solutions. The reported CPU times are the average of three runs for each case.

5.3 Greedy Heuristic Non-myopic Algorithm

To measure the efficiency of the heuristic we developed, we measure the time difference that it takes for the algorithm to reach a solution compared to the MILP commercial solver and the optimality gap in order to evaluate the quality of the solution.

The experiments were performed by generating random locations for a number of vehicles equal to 40% of the nodes of the network. The service time parameter is chosen as: $\mu = 1.2 * \sum_{i \in N} \sum_{g \in H} \lambda_{ig}$. The reason for choosing these parameters is to produce solutions where (1) the number of vehicles $|F|$ denotes how many times the 1-median problem is solved in the heuristic and increases proportionally with the network size (percentage of total nodes) and (2) we ensure that constraint (5) is binding. The results of the computational experiments are shown in Table 4 and suggest that the computational savings improve as the problem size increases.

The results for networks up to 1000 node-charges are promising. For large networks (over 400 nodes) the optimality is only around 22%, while the computational time is 100-500 times faster than commercial solvers. These performance measures indicate that the algorithm is suitable for deployment as an online rebalancing system.

TABLE 4 Computational times and optimality gap of the proposed heuristic

# of Nodes	MIP comp. time (sec)	Optimality Gap (%)	Heuristic comp. time (sec) (% MIP time)
24	1.9	3	0.4 (21%)
100	4.5	20	1.14 (25%)
200	5	5	2 (40%)
300	8	15	4 (50%)
400	30	23	4 (13%)
600	117	24	10 (8.5%)
700	195	22	22 (11%)
800	380	23	17 (4.5%)
1000	1034	22	23 (2.2%)

Section 6: Simulation Development

We developed a simulator using MATLAB for testing the rebalance strategies. The simulator is agent-based and includes two kinds of agents: customer agents and vehicle agents. The vehicle-to-customer assignment rules determine how the customers book the vehicles (in other words, how the vehicles are assigned to customers), and the rebalancing strategies determine how vehicles are rebalanced among different nodes.

6.1 Customer Agents

Customer agents enter the system, report their location and request a vehicle with a desired charge level. Each customer c in the set of customers $\mathcal{C}$ has an attribute tuple $\mathcal{A}_c \equiv (a_c, s_c, l_c, e_c, w_c)$, where a_c is the arrival time of c, $s_c \in \{0, 1, 2\}$ is the state of c, 0 representing waiting, 1 representing a state of having been served (i.e., been assigned a vehicle), and 2 representing a state of having abandoned the system without being served. The attribute, l_c, is the customer's location in the network. The attribute e_c is the required energy (charge level) for them to complete their trip and w_c is the maximal waiting time c is willing to endure.

At the beginning of each time step t, new customers join the waiting customers list $\mathcal{C}^w \subset \mathcal{C}$; their arrival time is set to $a_c = t$, and their state is set to $s_c = 0$. Then, in descending order of arrival time, each customer in $\mathcal{C}^w$ searches for the nearest available (idle) vehicle within a pre-specified *pick-up distance*. If no available vehicle is found, the customer remains in $\mathcal{C}^w$. Otherwise, the customer books a vehicle in accordance with the assignment rules (elaborated further in 6.3 below), their state becomes $s_c = 1$ and they are removed from the set $\mathcal{C}^w$. If no available vehicle is found by time step $t + w_c$, c abandons the system, their state is set to $s_c = 2$ and they are removed from the set $\mathcal{C}^w$. The updating rules for customer agents at each time step t are shown in Figure 6.

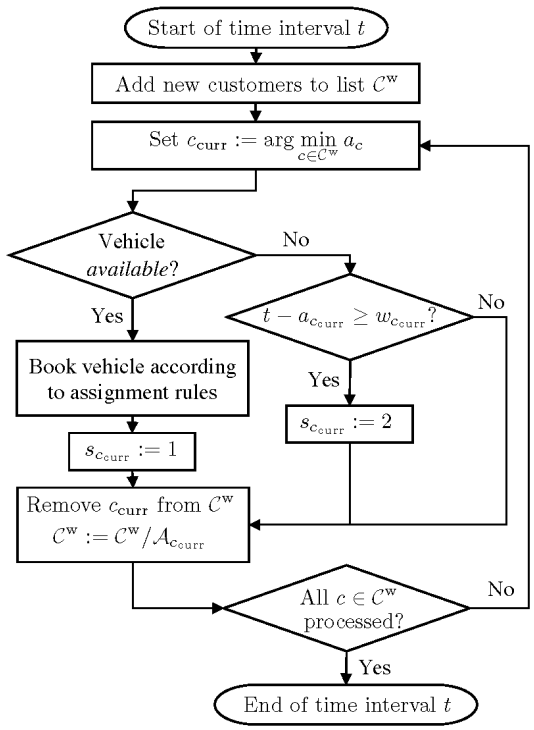


FIGURE 6. Flowchart of customer agent dynamics.

6.2 Vehicle Agents

Let $\mathcal{V}$ denote the set of vehicle s in the system. Each vehicle $v \in \mathcal{V}$ has an attribute tuple $\mathcal{A}_v \equiv \left(S_v, L_v, E_v, E_v^{\mathrm{d}}, L_v^{\mathrm{d}}, L_v^{\mathrm{ch}}\right)$, where $S_v \in \{0,1,2\}$ is v's state, 0 representing being idle, 1 representing being booked, and 2 represents being relocated. Attribute L_v is v's location in the network, E_v is v's charge level, E_v^{d} is v's desired charge level, L_v^{d} is v's destination location (when booked or is being rebalanced), and L_v^{ch} is the charging station location.

The state, S_v, changes from 0 to 1 when an idle vehicle is booked, and from 0 to 2 when an idle vehicle is being relocated (for charging purposes). Once a booked vehicle is returned by a customer, its state changes from 1 to 0. Once a relocating vehicle arrives to its destination, its state S_v changes from 2 to 0. Attributes L_v and E_v of v are updated in every time step t; L_v depends on v's state, E_v increases during charging and decreases if being used. When v is booked or is being relocated for rebalancing purposes L_v^{d} changes to the location of the new destination. If charging is required during rebalancing, two things happen: (i) L_v^{ch} holds the location of the scheduled charging station ($L_v^{\mathrm{ch}} = \emptyset$ if v is charged to the desired level) and (ii)

E_v^{d} changes to the desired/required charge level. A first-come-first-served rule is applied at the charging stations. Idle vehicles automatically re-charge themselves if they are located at a station that offers charging. The updating rules for vehicle agents at each time step t are illustrated in Figure 7.

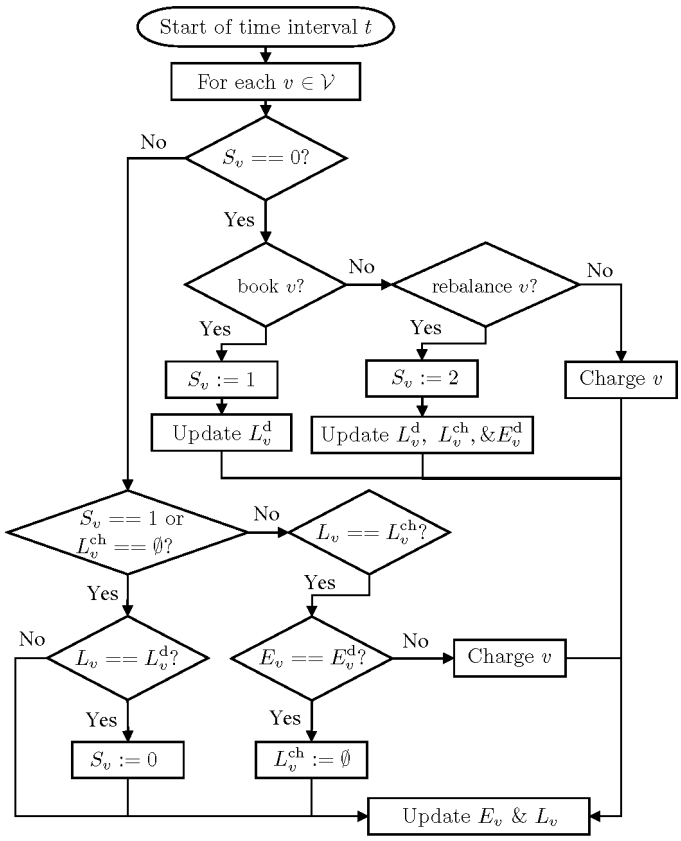


FIGURE 7. Flowchart of vehicle agent dynamics.

6.3 Vehicle-to-Customer Assignment Rules

Once a vehicle is booked by a customer, it is assigned to the customer. As mentioned in 6.1, customers are only allowed to book an idle qualified vehicle that is within a certain pick-up distance, e.g. 1 km. This distance is an input parameter for the model. When searching for available vehicles, only vehicles with $E_v \geq e_c + e_{v,c}$ are considered qualified, where $e_{v,c}$ is the additional energy needed for vehicle v to pick up customer c and then drive to the nearest charging station after dropping the customer off.

After a customer completes their booking, they cannot change their booking, even when a closer vehicle becomes available before the booked vehicle arrives to the customer. Hence, when choosing a pick-up distance parameter, one needs to consider such trade-offs: there is a higher chance of finding an available vehicle instantly with larger pick-up distances but higher risks of finding a more suitable vehicle after booking. This rule differs significantly from some previous work which allows customers to book the nearest available vehicle without a limit on pick-up distance. Since vehicles are expected to be returned anywhere anytime in the network, absence of a pick-up distance makes it possible to match a vehicle and customer that are far away from each other, while the customer could have chosen another much closer vehicle had they waited a little.

6.4 Rebalancing Strategies

Rebalancing strategies are used to determine how to rebalance vehicles among different nodes. Note that a rebalanced vehicle cannot be booked until it reaches its destination and changes its state to idle. The assignment rule ensures that customers cannot book a vehicle that is outside the pick-up distance. The relocation strategy, however, allows vehicles anywhere in the network to be rebalanced to any other part of the network. The simulator is open to any rebalance strategies, which means we could test different kind of rebalancing strategies in this car-sharing system.

The input for the rebalancing strategy could vary depending on what rebalancing strategy we are using. For example, for the proposed model in this study, we need to feed the rebalancing algorithm the average arrival rate and service rate for each node, the location and charge level of all available vehicles, the driving distance and driving time between each pair of nodes, the location of all charging stations, and the number of chargers at each charging stations. For the other simple rebalancing strategy we used for comparison in Section 7, which is called the ChargerChasing strategy, all the input we need is the location and charge level of all available vehicles, and the location of all charging stations.

Note that for different rebalancing strategies, the output is the same. It should specify the rebalancing route for all available vehicles, which is a sequence of nodes that the rebalanced vehicle should visit. If a vehicle is not assigned for rebalancing, then its route is just its current node.

6.5 Simulation Tool

We have uploaded the developed simulator in zenodo. Open the main.m in MATLAB, choose the corresponding rebalance strategy, and click "run". One can also adjust the parameters in the parameterSetup.m file. The adjustable parameters include: total simulation time, simulation step, fleet size, maximum charge level, charge time, vehicle range, customer maximum waiting time, and pick-up distance. Although this simulator is tailored for an electric vehicle car-sharing system, one could easily simulate non-electric vehicles by adjusting the vehicle range to a large number and the charge time to a small number. A screenshot of the simulator output is shown in Figure 8. The + is the position of the TAZs, while the size of the green square represents the number of available vehicles within this TAZ. We do not trace the real time position of the occupied or rebalanced vehicles.

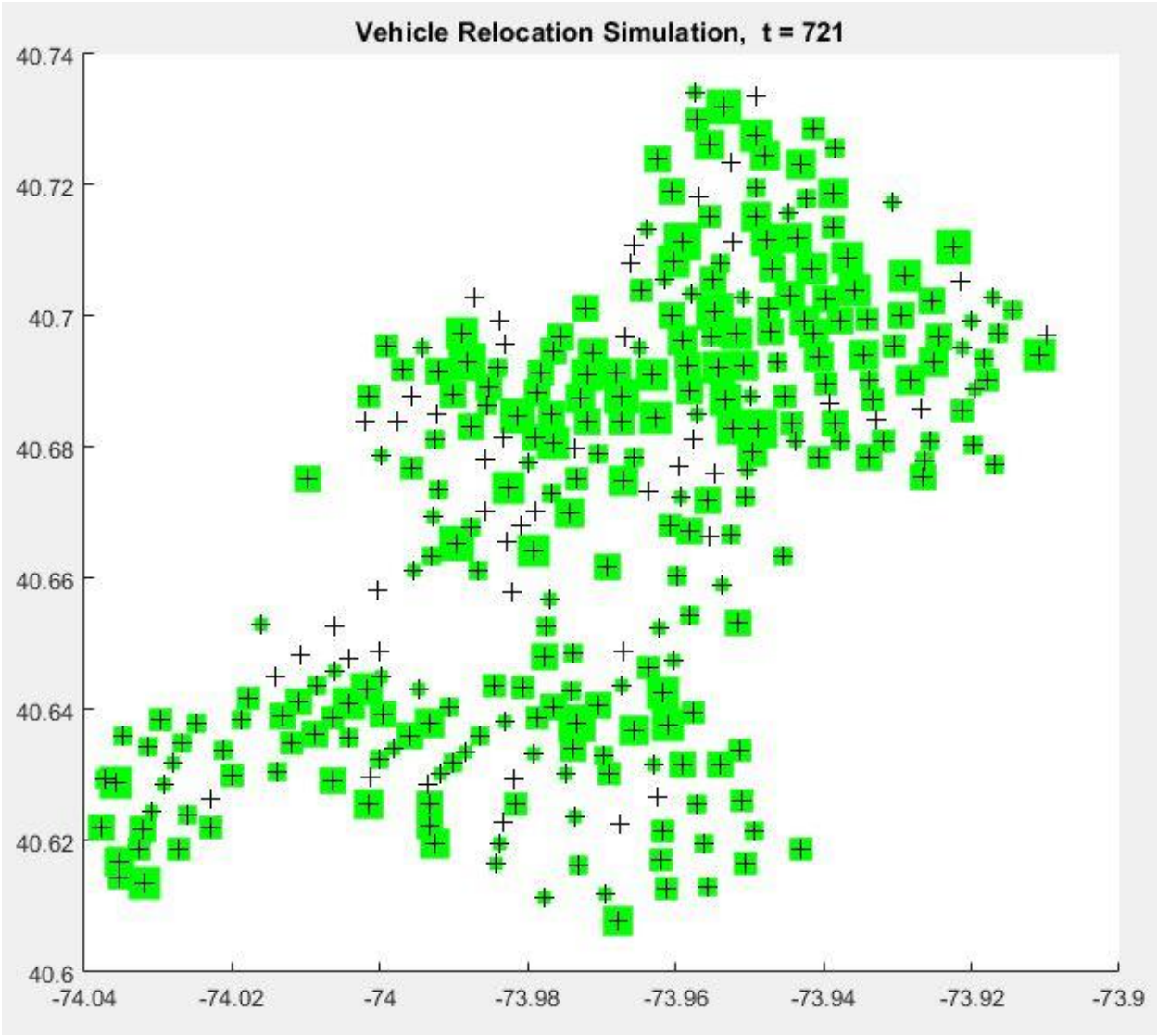


FIGURE 8. Screenshot of the simulator.

Section 7: Simulation Experiments

7.1 Small Network Simulation Trial

Using a custom-created simulation environment in MATLAB, we generated customer arrivals for the network shown in Figure 3. Rebalancing decisions are made for idle vehicles at the start of every time interval, which is every hour. For the small network we solve the MILP to

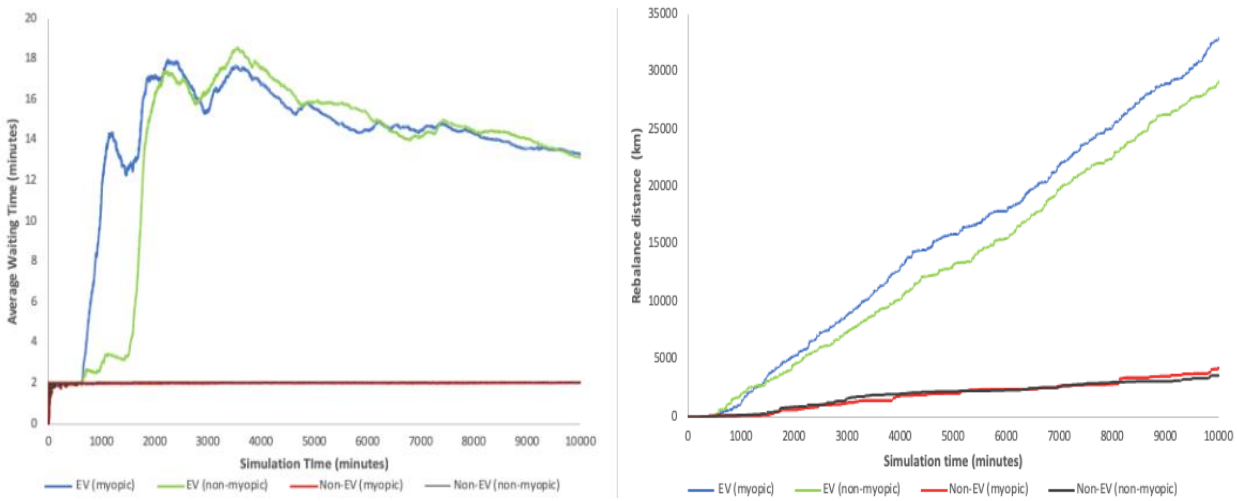
obtain exact solutions for each time step t. Details about the simulation input parameters are shown in Table 5.

TABLE 5. Input parameters for small network computational experiment

Parameters	Value
Number of Nodes	6
Layers	5
Fleet	25
Station Node Locations	[2,6]
Station Capacities	[4,4]
Service constraint μ_{ij} (non-myopic case)	144
Vehicle Average Speed	60 kph
Simulation Duration	10000 $minutes$

Running the simulation with the proposed model on the simple network is promising, as shown in Figure 9. Figure 9(a) shows the average wait time of passengers, which inevitably increases when EV charging is required because vehicles will tend to be unavailable longer. The accumulated rebalancing cost is captured in Figure 9(b), which shows that the presence of the EV charging constraint increases rebalance costs, but those costs can be mitigated by

approximately 15% in this example using the non-myopic rebalancing model that considers Markovian customer demand and queue delays due to vehicle unavailability.



(a) **(b)**

FIGURE 9: (a) Average waiting time and **(b)** Accumulated rebalance distance over time.

7.2 Brooklyn Network Experiment

We tested the algorithm in the Brooklyn TAZ network, shown in Figure 10. We use data obtained from the BMW ReachNow car-sharing operations in 2017 to simulate passenger arrivals and destinations. Our dataset includes all trips for the month of September and an average of 231 car pickups per day. Demand density is shown in Figure 11. The average reservation time is 6 hours and the average mileage is 25.8. From the operator's side, the per-passenger net revenue collected from operating the system is $22.6, while the after-tax cost for the average customer rises to $27.2.

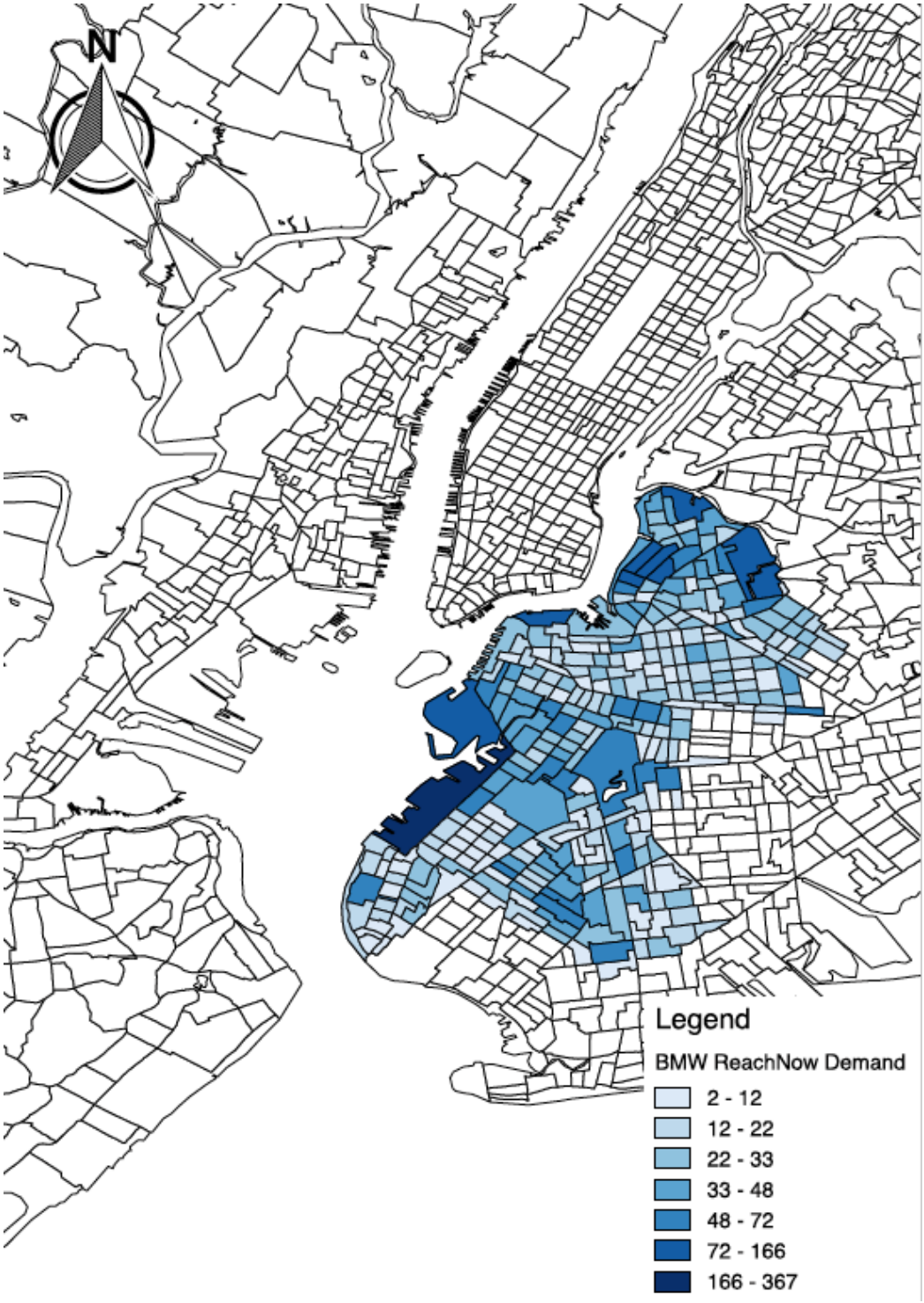


FIGURE 10. Brooklyn traffic analysis zones ((NYMTC, 2010) with BMW ReachNow dataset O-D pairs)

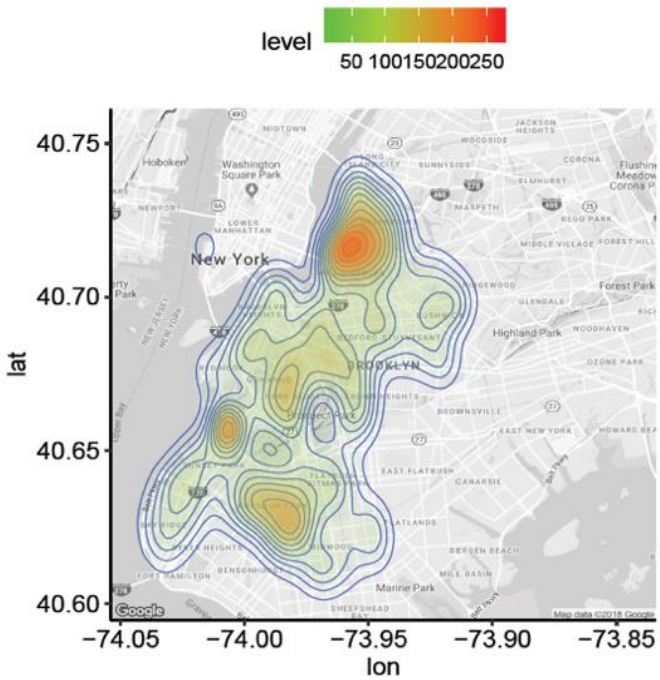


FIGURE 10. Demand density from BMW ReachNow dataset.

To test the heuristic, we assume a fleet of 100 electric vehicles with charging stations as shown in Figure 12. The charging facilities already exist to serve privately owned electric vehicles. We further assume that passengers can tolerate up to 30 mins of waiting time before they choose to balk. We run the simulation over a period of one month.

TABLE 6. Summary of aggregate measures from each scenario

Scenarios	EV_Rebalance_ Non Myopic	EV_Rebalance_Myopic	NonEV_No rebalance	EV_ChargerChasing
Number of lost customers	1277	1245	949	4433
Number of customers	6766	6766	6766	6766

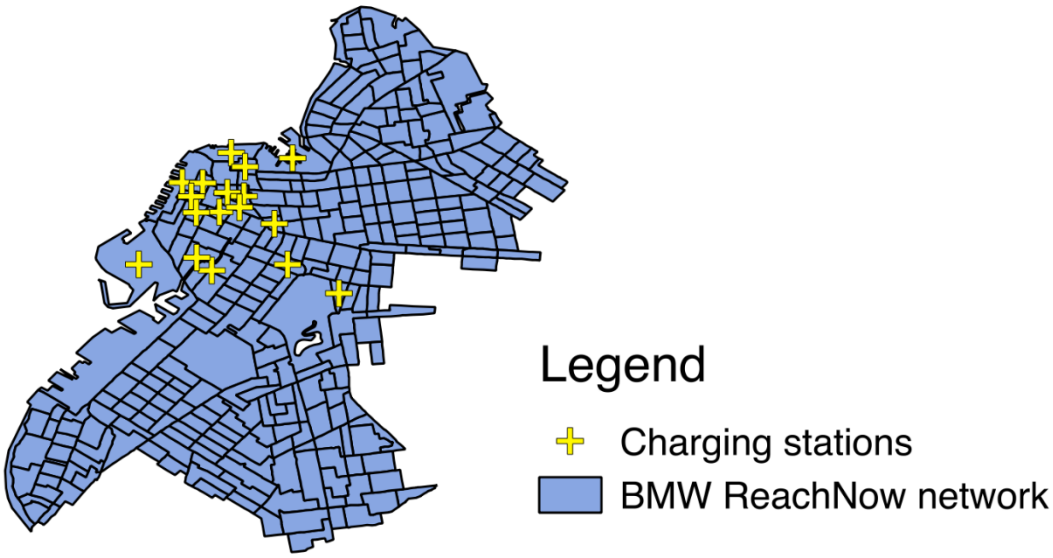


FIGURE 12. Existing charging stations used for the analysis of the EV fleet operations.

To illustrate the benefits of our non-myopic algorithm, we simulate four different scenarios with the same demand realizations but different rebalancing policies and/or vehicle technology:

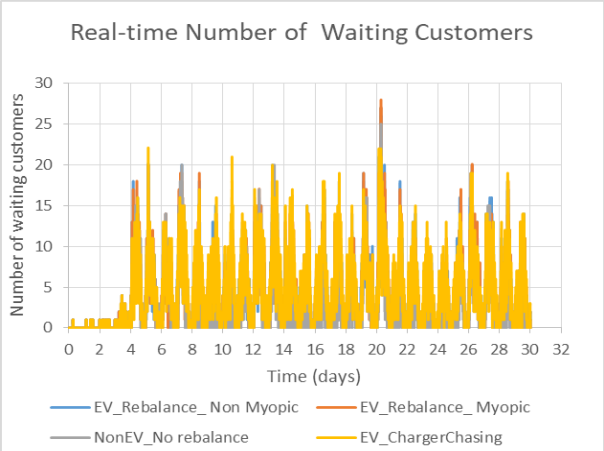
a) A non-rebalancing car-sharing operation with gas-fueled vehicles which simulates the current BMW operations in Brooklyn.
b) Myopic rebalancing by using our model with the constraint (5) relaxed
c) Non-myopic rebalancing strategy
d) We introduce a minimum rebalance distance scenario named "ChargerChasing" that assigns vehicles to the closest charging facility right after they drop off a passenger, while giving priority to lowest charged vehicles to use charging ports.

The parameters used in each scenario are listed in Table 7.

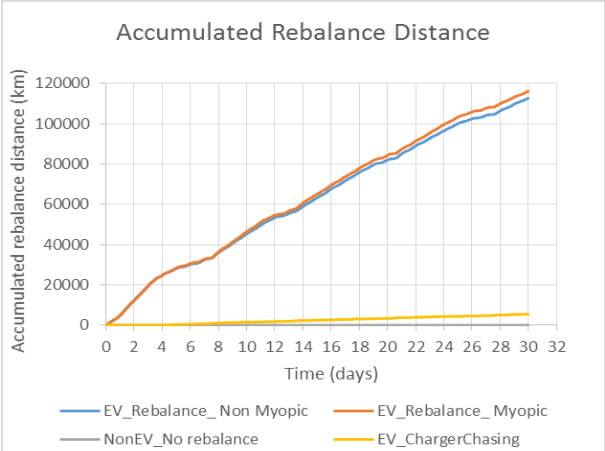
TABLE 7 Parameters used in the 4 tested scenarios for Brooklyn network

Parameters	NonEV_No rebalance	EV_Non Myopic	EV_Myopic	EV_ChargerChasing
Simulation horizon	30 days	30 days	30 days	30 days
Simulation step	1 min	1 min	1 min	1 min
Fleet size	100 vehs	100 vehs	100 vehs	100 vehs
Vehicle range	**600 km**	200 km	200 km	200 km
Max charge level	5	5	5	5
Charge time	**0 min**	100 min	100 min	100 min
Customer tolerance time	30 min	30 min	30 min	30 min
Pick-up distance	3 km	3 km	3 km	3 km

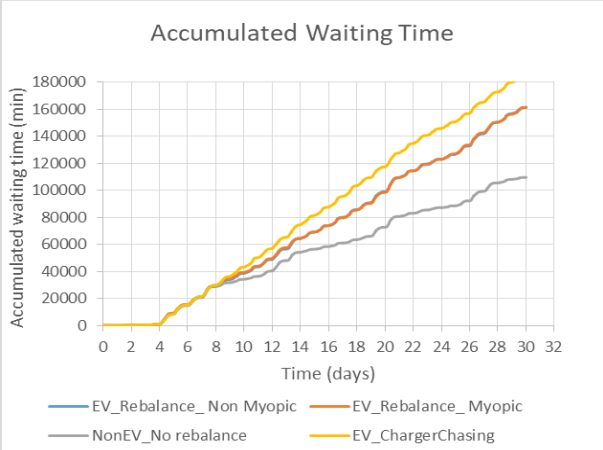
For these instances, we compare the rebalancing kilometers, average waiting time per customer, total delay and customers on the queue at each time period. The results are shown in Figure 13. The detailed data are publicly available on Zenodo.



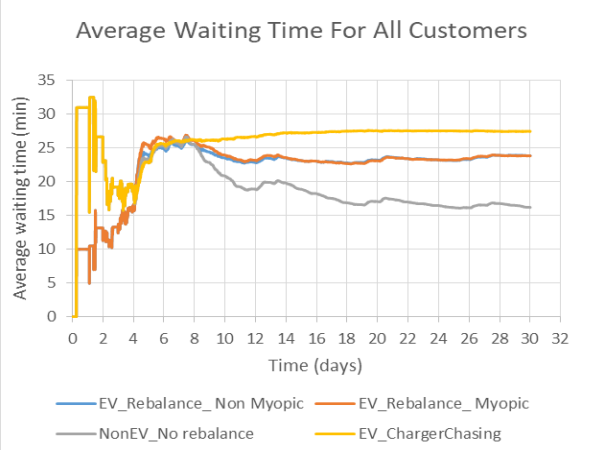
(a): Customers on queue



(b): Total rebalance kilometers



(c): Total system delay



(d): Average waiting time per customer

FIGURE 13. Comparison between different scenarios from simulation

When comparing the performance of the non-myopic algorithm against the myopic version, we find that both versions achieve similar customer waiting times.

The non-myopic algorithm reduces the cost of rebalancing (or rebalance kilometers) for the operator by **5% - 15%** when: $\mu = 10\ customer\ per\ hour$. The efficiency of this approach is also observable when comparing the performance measures of the "ChargerChasing" approach. The average waiting time is **15% more** when this strategy is used instead of the heuristic algorithm.

Another interesting finding is the overall efficiency of gas-fueled vehicles even without a rebalancing strategy. However, the operating costs involve fuel consumption which is not calculated in this study.

Overall the non-myopic algorithm provides fast and efficient rebalancing for electric one-way car sharing online systems. To achieve maximum cost savings, we must choose an appropriate μ for constraint (5) for each node in the network.

Section 8: Key Findings

In this study, we propose a non-myopic idle vehicle rebalancing model in an electric carsharing system by considering customer stochastic charging demand and capacitated charging station constraints. To the best of our knowledge, this is the first facility relocation model formulation that considers queueing constraints applicable to EV charging. We formulate the problem as a p-median problem embedded with a capacitated minimum cost flow network problem on a node-charge graph to jointly determine the relocation and routing decisions of idle vehicles under available charging capacity. The formulation on a two-dimensional node-charge graph allows us to explicitly consider a customer's charging demand profile and optimize rebalancing operations of idle vehicles accordingly. An illustrative example on a small network shows the assigned vehicle flow and rebalanced positions of idle vehicles minimize total designed objective while optimally utilizing the available charging capacity in the system.

To address the computational complexities of real car-sharing networks, we propose a greedy heuristic algorithm that incorporates queuing constraints and solves the relocation problem about 20x times faster than the MILP. We further test the performance of our algorithm on a set of large test networks. The results show 8-20% optimality gap for various instances and up to 1000 nodes and 400 idle vehicles. Our new algorithm demonstrated significant cost savings and computational efficiency for operating large electric rebalancing systems. Through our simulation procedure we notice consistency in the results between the realistic Brooklyn network and the smaller illustrative network as presented in Section 3.

Section 9: Technology Transfer, Dissemination, and Broader Impacts

9.1 Technology Transfer

In this section we provide links to all the completed products we developed during this project as shown in Table 8. Each of the items in the table served on its own to transfer "new knowledge" or contributed in developing the product that did.

TABLE 8: Delivered products and source links

Output	Purpose
Original MILP formulation	Comparison against more computationally efficient methods/ teaching tool for graduate students
Teitz & Bart Heuristic	Benchmark tool for new methods
Markovian Heuristic	Proposed online rebalancing system to be implemented for a real EV car-sharing system
Heuristic results	Verify and display the advantages of our heuristic/ measure the effectiveness of the solution
Simulation input data (BMW ReachNow)	Tool for providing realistic simulation conditions
Simulation	Measure the effectiveness of the new algorithm/ evaluate and compare different strategies

9.2 Dissemination

The initial MILP formulation was presented at the TRB 98th Annual Meeting 2019 for small and large test case instances. The purpose of the presentation was to introduce a new rebalancing system for electric vehicles and demonstrated the need for computationally faster algorithms.[1]

This presentation is intended to introduce an online heuristic algorithm to rebalance electric vehicles in large car-sharing operations. The presentation uses real data obtained from the BMW ReachNow project in Brooklyn, New York.[2]

Another rebalancing strategy was introduced in the IEEE conference proceedings using the BMW ReachNow data. This strategy is evaluated using agent-based simulation for shared automated electric vehicle fleets.[3]

A paper is in preparation for submission to a journal for publication now.

9.3 Broader Impacts

In addition to the direct dissemination and technology transfer, this research has led to a number of broader impacts.

Student training and involvement: In addition to the main research team, we participated in the ARISE program to expose K-12 STEM students to this research and other projects from our lab. PI Chow helped advise a Vertically Integrated Project team in the area of smart cities to work with smart grid charging. PI Chow also served as a judge for the Forbes Idea Incubator Challenge, which empowered women students to develop innovative ideas in sustainable vehicle technologies and smart mobility[4].

---

[1] Ma, T.Y., Pantelidis, T., Chow, Y.J. Optimal queuing-based rebalancing for one-way electric carsharing systems with stochastic demand: TRB 98th Annual conference proceedings, 2019.
[2] Pantelidis, T., Li, L., Ma, T.Y., Chow, J. Y.J., Jabari, S.E.: Doubly-constrained rebalancing for one-way electric carsharing systems with capacitated charging stations: INFORMS Transportation Society & Logistics Society workshop, 2019
[3] Li, L., Lin, D., Pantelidis, T., Jabari, S.E., Chow, J.Y.J.: An Agent-Based Simulation for Shared Automated Electric Vehicles with Vehicle Relocation: IEEE ITS conference proceedings, 2019
[4] http://c2smart.engineering.nyu.edu/2019/03/20/c2smart-faculty-judge-forbes-idea-incubator-challenge/

Public engagement: The team presented our work at the NYU Tandon Research Expo[5], which exposes our project to the local community as well as to other students at NYU Tandon.

Industry engagement: PI Chow presented this work at the Building Energy Smart Technologies (BEST) Workshop hosted by the NSF Industry-University Collaborative Research Center[6]. PI Chow also served as a judge on the NYCx Climate Action Challenge organized by the NYC Mayor's Office of the Chief Technology Officer[7].

[5] https://engineering.nyu.edu/events/2019/05/03/2019-research-expo
[6] https://best-iucrc.net/wp-content/uploads/2019/04/BEST_Agenda_4-5-2019_updated.pdf
[7] http://c2smart.engineering.nyu.edu/2018/08/15/deputy-director-serves-as-judge-in-nycx-competition/

References

Barth, M. and Todd, M. (1999) 'Simulation model performance analysis of a multiple station shared vehicle system', *Transportation Research Part C: Emerging Technologies*. Pergamon, 7(4), pp. 237–259.

Bloomberg (2018). *Daimler BMW Reach a Deal to Merge Car-Sharing Units - Bloomberg* (no date). Available at: https://www.bloomberg.com/news/articles/2018-03-28/daimler-bmw-are-said-to-reach-deal-to-merge-car-sharing-units (Accessed: 15 May 2019).

Boyacı, B., Zografos, K. G. and Geroliminis, N. (2015) 'An optimization framework for the development of efficient one-way car-sharing systems', *European Journal of Operational Research*, 240(3), pp. 718–733.

Brandstätter, G., Kahr, M. and Leitner, M. (2017) 'Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand', *Transportation Research Part B: Methodological*. Pergamon, 104, pp. 17–35.

Bruglieri, M., Colorni, A. and Luè, A. (2014) 'The Vehicle Relocation Problem for the One-way Electric Vehicle Sharing: An Application to the Milan Case', *Procedia - Social and Behavioral Sciences*. Elsevier, 111, pp. 18–27.

Chow, J. Y. J. and Sayarshad, H. R. (2014) 'Symbiotic network design strategies in the presence of coexisting transportation networks', *Transportation Research Part B: Methodological*. Pergamon, 62, pp. 13–34.

Clemente, M., Fanti, M. P. and Ukovich, W. (2014) 'Smart Management of Electric Vehicles Charging Operations: the Vehicle-to-Charging Station Assignment Problem', *IFAC Proceedings Volumes*, 47(3), pp. 918–923.

Fan, W. (David), Machemehl, R. B. and Lownes, N. E. (2008) 'Carsharing', *Transportation Research Record: Journal of the Transportation Research Board*. Transportation Research Board of the National Academies , 2063(1), pp. 97–104.

Di Febbraro, A., Sacco, N. and Saeednia, M. (2012) 'One-Way Carsharing: Solving the Relocation Problem', *Transp. Res. Rec. %@ 0361-1981 %U https://doi.org/10.3141/2319-13*, 2319(1), pp. 113–120.

Hu, L. and Liu, Y. (2016) 'Joint design of parking capacities and fleet size for one-way station-based carsharing systems with road congestion constraints', *Transportation Research Part B: Methodological*. Pergamon, 93, pp. 268–299.

Jorge, D., Molnar, G. and de Almeida Correia, G. H. (2015) 'Trip pricing of one-way station-based carsharing networks with zone and time of day price variations', *Transportation Research Part B: Methodological*. Pergamon, 81, pp. 461–482.

Jung, J., Chow, J.Y.J., Jayakrishnan, R., Park, J. (2014) 'Stochastic dynamic itinerary interception refueling location problem with queue delay for electric taxi charging stations', *Transportation Research Part C: Emerging Technologies*. Elsevier Limited, 40, pp. 123–142.

Kaspi, M. *et al.* (2016) 'Regulating vehicle sharing systems through parking reservation policies: Analysis and performance bounds', *European Journal of Operational Research*. North-Holland, 251(3), pp. 969–987.

Kek, A. G. H. *et al.* (2009) 'A decision support system for vehicle relocation operations in carsharing systems', *Transportation Research Part E: Logistics and Transportation Review*, 45(1), pp. 149–158.

Krok, A. (2016). Car2Stop: Car2Go shuts down services in San Diego. CNET, Nov 21.

Li, X. *et al.* (2016) 'Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model', *Transportation Research Part B: Methodological*. Pergamon, 88, pp. 21–45.

Lufkin, B. (2016) '6 Electric Car-Sharing Programs Better Than a Billion Teslas on the Road', *Gizmodo %U https://gizmodo.com/6-electric-car-sharing-programs-better-than-a-billion-t-1766904615*.

Marianov, V. and ReVelle, C. (1996) 'The Queueing Maximal availability location problem: A model for the siting of emergency vehicles', *European Journal of Operational Research*. North-Holland, 93(1), pp. 110–120.

Marianov, V. and Serra, D. (1998) 'Probabilistic, Maximal Covering Location—Allocation Models forCongested Systems', *Journal of Regional Science*. John Wiley & Sons, Ltd (10.1111), 38(3), pp. 401–424.

Marianov, V. and Serra, D. (2002) 'Location–Allocation of Multiple-Server Service Centers with Constrained Queues or Waiting Times', *Annals of Operations Research*. Kluwer Academic Publishers, 111(1/4), pp. 35–50.

Martin, E. and Shaheen, S. (2016) 'Impacts of Car2go on Vehicle Ownership,Modal Shift, Vehicle Miles Travelled, and Greenhouse Gas Emissions: An Analysis of Five North American Citiies', *Transportation Sustainability Research Center, UC Berkeley, California*.

Nair, R. and Miller-Hooks, E. (2011) 'Fleet Management for Vehicle Sharing Operations | Transportation Science %U https://pubsonline.informs.org/doi/abs/10.1287/trsc.1100.0347'.

Navigant Research (2017). Mobility as a Service – The future of moving people: carsharing, ridehailing, micro transit, automated mobility, and P2P rental services. Retrieved from https://www.navigantresearch.com/reports/mobility-as-a-service

Nourinejad, M. *et al.* (2015) 'Vehicle relocation and staff rebalancing in one-way carsharing systems', *Transportation Research Part E: Logistics and Transportation Review*, 81, pp. 98–113.

Owen, S. H., & Daskin, M. S. (1998). Strategic facility location: A review. *European journal of operational research*, *111*(3), 423-447.

Sayarshad, H. R. and Chow, J. Y. J. (2017) 'Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem', *Transportation Research Part E: Logistics and Transportation Review*, 106, pp. 60–77.

Teitz, M. B. and Bart, P. (1968) 'Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph', *Operations Research*. INFORMS, 16(5), pp. 955–961.

Waserhole, A. and Jost, V. (2016) 'Pricing in vehicle sharing systems: optimization in queuing networks with product forms', *EURO J. Transp. Logist.* 5(3), pp. 293–320.

Xu, M., Meng, Q. and Liu, Z. (2018) 'Electric vehicle fleet size and trip pricing for one-way carsharing services considering vehicle relocation and personnel assignment', *Transportation Research Part B: Methodological*. Pergamon, 111, pp. 60–82.

Zhang, R. and Pavone, M. (2016) 'Control of robotic mobility-on-demand systems: A queueing-theoretical perspective', *The International Journal of Robotics Research*. SAGE Publications Sage UK: London, England, 35(1–3), pp. 186–203.

Zhao, M. *et al.* (2018) 'An integrated framework for electric vehicle rebalancing and staff relocation in one-way carsharing systems: Model formulation and Lagrangian relaxation-based solution approach', *Transportation Research Part B: Methodological*. Pergamon, 117, pp. 542–572.