

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Modeling Shared-use Urban Mobility Systems to Increase System Performance

Permalink

<https://escholarship.org/uc/item/3gs0r0dj>

Author

Regue Grino, Robert

Publication Date

2015

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Modeling Shared-use Urban Mobility Systems to Increase System Performance

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Civil Engineering

by

Robert Regué Griñó

Dissertation Committee:
Professor Will Recker, Chair
Professor Jay Jayakrishnan
Professor Amelia Regan

2015

Portion of Chapter 2 © 2015 Elsevier
All other materials © 2015 Robert Regué Griñó

DEDICATION

To my family for all their support and encouragement.

To Alba for her unending support. I would not have completed my doctorate without you.

TABLE OF CONTENTS

LIST OF FIGURES	V
LIST OF TABLES	VI
ACKNOWLEDGMENTS	VII
CURRICULUM VITAE	VIII
ABSTRACT OF THE DISSERTATION	X
1. INTRODUCTION	1
2. A BIKESHARING REBALANCING FRAMEWORK	4
2.1. INTRODUCTION AND MOTIVATION	4
2.2. RELATED WORK	9
2.2.1. DEMAND FORECASTING	9
2.2.2. REBALANCING PROBLEM.....	12
2.3. PROPOSED FRAMEWORK	15
2.3.1. DATA DESCRIPTION.....	17
2.3.2. DEMAND FORECASTING MODEL.....	22
2.3.2.1. Gradient Boosting Machines	23
2.3.2.2. Neural Networks.....	25
2.3.2.3. Calibration	26
2.3.3. INVENTORY LEVELS MODEL	29
2.3.4. REDISTRIBUTION NEEDS MODEL.....	31
2.3.5. VEHICLE ROUTING MODEL	35
2.3.6. USER RELOCATION MODEL.....	41
2.4. PERFORMANCE MEASURES	46
2.5. EXPERIMENTAL RESULTS	47
2.5.1. DEMAND FORECASTING TECHNIQUES COMPARISON	50
2.5.2. CASE 1: PM PEAK HOUR	55
2.5.3. CASE 2: ONE WEEK SIMULATION	57
2.5.3.1. Scenario 1: Measuring the impact of the fleet size.....	58
2.5.3.2. Scenario 2: Measuring the impact of the vehicle capacity.....	59
2.5.3.3. Scenario 3: Measuring the impact of visiting buffering stations.....	60
2.5.3.4. Scenario 4: Measuring the impact of user relocation policies	61
2.6. CONCLUSIONS AND FUTURE WORK	63
3. CAR2WORK: A NEW MOBILITY CONCEPT FOR COMMUTERS	66
3.1. INTRODUCTION AND MOTIVATION	66
3.2. RELATED WORK	76
3.3. SYSTEM DEFINITION	78
3.4. MODEL FORMULATION	81
3.5. SOLUTION APPROACH	85

3.5.1. MASTER PROBLEM.....	87
3.5.2. SUB-PROBLEM	89
3.5.2.1. A note on Benders' Feasibility cut	90
3.5.3. LINK REDUCTION STRATEGY.....	90
3.6. NUMERICAL TESTS	93
3.6.1. DATA.....	93
3.6.2. RESULTS.....	95
3.7. CONCLUSIONS AND FUTURE WORK.....	99
<u>4. CONCLUSIONS.....</u>	<u>101</u>
<u>5. REFERENCES.....</u>	<u>105</u>

LIST OF FIGURES

FIGURE 1: SPATIAL IMBALANCE AT BICING, BARCELONA AT 11PM (LEFT). TEMPORAL IMBALANCE AT HUBWAY, BOSTON (RIGHT)	6
FIGURE 2: DIFFERENT REPOSITIONING MECHANISMS. RICKSHAWS (LEFT), VANS (RIGHT).....	7
FIGURE 3: BIKESHARING REBALANCING FRAMEWORK	16
FIGURE 4: THE HUBWAY BIKESHARING SYSTEM (2012)	18
FIGURE 5: MEAN NUMBER OF BIKES WITH 1 STANDARD DEVIATION FOR STATION 14 AND STATION 15	22
FIGURE 6: NETWORK DIAGRAM FOR A TWO-LAYER NEURAL NETWORKS (BISHOP 2006).	26
FIGURE 7. INVENTORY LEVELS FOR STATION 14 AND STATION 38	31
FIGURE 8. REDISTRIBUTION NEEDS EXAMPLE	33
FIGURE 9: VEHICLE ROUTING ILLUSTRATIVE EXAMPLE	41
FIGURE 10: USER RELOCATION MODEL OUTCOMES	45
FIGURE 11: SOFTWARE USER INTERFACE.....	50
FIGURE 12: ERROR MEASURES FOR THE TEST AND VALIDATION DATASETS	51
FIGURE 13: MEAN VARIABLE IMPORTANCE FOR THE ENTIRE SYSTEM AT DIFFERENT TIME WINDOWS FOR THE NON-CALIBRATED GBM MODELS	55
FIGURE 14: PERFORMANCE MEASURES TRENDS BASED ON FLEET SIZE.....	59
FIGURE 15: PERFORMANCE MEASURES TRENDS BASED ON VEHICLE CAPACITY	60
FIGURE 16: PERFORMANCE MEASURES TRENDS BASED ON Γ COEFFICIENT.	61
FIGURE 17: PERFORMANCE MEASURES TRENDS BASED WITH USER RELOCATION POLICIES AND NUMBER OF VEHICLES.....	62
FIGURE 18: NORTH AMERICA CAR-SHARING TRENDS. ADAPTED FROM (SHAHEEN 2013).....	67
FIGURE 19: DIRECTIONS FROM UNIVERSITY OF CALIFORNIA, IRVINE TO IRVINE TRANSPORTATION CENTER	75
FIGURE 20: DIRECTIONS FROM UNIVERSITY OF CALIFORNIA, IRVINE TO LOS ANGELES UNION STATION	75
FIGURE 21: CAR2WORK EXAMPLE.....	81
FIGURE 22: AGGREGATION/DISAGGREGATION ITERATIVE ALGORITHM STEPS	86
FIGURE 23: TIME WINDOWS DEFINITION	91
FIGURE 24: FIRST SCENARIO (LEFT). SECOND SCENARIO (RIGHT)	94
FIGURE 25: SC1-1 VEHICLE ROUTES FOR THE AM PEAK.	99

LIST OF TABLES

TABLE 1: VARIABLE DEFINITION. NCDC: NATIONAL CLIMATIC DATA CENTER.....	20
TABLE 2: INITIAL INVENTORY LEVELS AND EVENT TYPES.....	30
TABLE 3: SIMULATION PARAMETERS	49
TABLE 4: CALIBRATION PARAMETERS RANGES AND TYPES.....	51
TABLE 5: RMSE VALUES FOR GBM MODELS WITHOUT VALIDATION DATASET.....	54
TABLE 6: PM PEAK PERIOD SIMULATION RESULTS	57
TABLE 7: CAR2WORK PARAMETERS SETTINGS.....	95
TABLE 8: EXPERIMENTAL RESULTS	98

ACKNOWLEDGMENTS

This research has been supported, in part, by grants from the University of California Transportation Center, the Balsells-Generalitat de Catalunya Fellowship and the Fundación Caja Madrid Fellowship. Their support is gratefully acknowledged.

Some sections of this dissertation are a reprint of the material as it appears in Transportation Research Part - E: Logistics and Transportation Review, entitled Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. The co-author listed in this publication directed and supervised research which forms the basis for the dissertation.

The author also acknowledges the support from his family, the Balsells fellows, the Institute of Transportation Studies at University of California, Irvine, and Dr. Recker for their support, guidance and encouragement.

CURRICULUM VITAE

7524 Palo Verde Road · Irvine, CA 92617 · (949) 214-9287 · robert.regue@gmail.com · www.robertregue.com

EDUCATION

- PhD, Civil Engineering** Sep. 2015
University of California, Irvine, United States. GPA: 3.98
Dissertation: Modeling Shared-use Urban Mobility Systems to Increase System Performance.
- MSc, Transport Policy & Business Management** Sep. 2010
Loughborough University, United Kingdom. Grade: Honors
Dissertation: An Investigation of the Feasibility of Implementing a Freight Tram Scheme in Barcelona.
- B.S, Industrial Engineering, Mechanical Branch.** Jul. 2009
Universitat Politècnica de Catalunya (UPC), Spain. Grade: 8.55/10
Dissertation: Design of a freight Light Rail Vehicle based on the ALSTOM CITADIS platform.

SKILLS

Optimization, Machine Learning, Predictive Analytics, Data Analysis, Web Development, MATLAB, Python, HTML, Javascript, Django, Cplex, Gurobi.

RESEARCH EXPERIENCE

- Graduate Student Researcher** Sep. 2010 - Present
Dept. of Civil and Environmental Engineering, University of California, Irvine.
- Developed a framework to solve the dynamic rebalancing problem in bike-sharing systems combining machine learning and optimization techniques.
 - Car2work - New shared-use mobility concept for the last mile connection of Transit Stations and Workplaces.
 - Developed a calibration methodology for the Household Activity Pattern Problem to optimally schedule household activities along the day.
 - Developed course material for an online introductory graduate course to activity based modeling.
 - Study of thru traffic at University of California, Irvine campus using Bluetooth re-identification.
 - Modeling of corridor systems: coordination of on-ramp metering and arterial signalization.
 - VISLABE: Visualization of lane changing maneuvers from vehicle trajectories extracted from aerial photography.

WORK EXPERIENCE

- Engineering Research Intern** Jun. 2015 - Sep. 2015
Scoop Technologies, Inc
- Co-Founder. CTO** Nov. 2013 - Present
Beacon Assisted Technologies LLC. (www.kaleri.com), United States.
- Developed the backend and content management system in Django (python-based web framework).
 - Implemented REST API.
 - Developed first prototype of iOS mobile App in PhoneGap.
 - Developed website frontend.
 - Coordinated with a team of overseas developers.
 - Experience with the iBeacon ecosystem.
- Engineering Intern** Jun. 2008 - Jul. 2009
ALSTOM TRANSPORTE, SA. Sourcing Department, Spain.
- Assisted site sourcing manager in managing supplier panel.
 - Developed a tool to forecast supplier workload according to projected tenders.

AWARDS

- UCCONNECT Dissertation Grant - UC Center on Economic Competitiveness in Transportation 2015
- Railway Association of Southern California Scholarship 2014
- 1st Place Butterworth Product Development Competition, 2013 - UCI 2013
- Fundación Caja Madrid Fellowship to pursue Higher Education Studies in US 2011 – 2012

Balsells-Generalitat de Catalunya Fellowship to pursue Higher Education Studies at UCI	2010
Transportation Planning Associates Prize for most outstanding Master Dissertation	2010
Universia-Fernado Alonso Scholarship to pursue an MSc in United Kingdom	2009
Number one student distinction, Graduated on 2004-2009 at Universitat Politècnica de Catalunya (UPC), Terrassa School (ETSEIAT)	2009

PUBLICATIONS

Journals

- Regué, R.**, Allahviranloo, M. and Recker R., (Under Review). A goal programming model of activity participation and scheduling. *Transportation Research Part B*.
- Allahviranloo, M., **Regué, R.** and Recker, R., (Under Review). Pattern Clustering and Activity Inference. *Transportation*.
- Regué, R.** and W. Recker (2014). Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transportation Research Part E - Logistics and Transportation Review*, 72, pp. 192-209.
- Regué, R.** and A.L Bristow (2013). Appraising Freight Tram Schemes: A Case Study of Barcelona. *European Journal of Transport and Infrastructure Research (EJTIR)*, 13(1), pp. 56-78

Conferences

- Regué, R.**, Masoud, N. and Recker, R., (2015, Under Review). Car2work: A shared mobility concept to connect commuters with workplaces. 95th Annual Meeting of Transportation Research Board, Washington, D.C.,US.
- Regué, R.**, Allahviranloo, M and Recker, R., (2015, accepted). Understanding Household Priorities when Scheduling Activities. 14th International Conference on Travel Behavior Research, London, UK.
- Regué, R.**, Allahviranloo, M and Recker, R., (2015). Understanding Household Priorities when Scheduling Activities. 94th Annual Meeting of Transportation Research Board, Washington, D.C.,US.
- Regué, R.** and Recker, W., (2014). Proactive vehicle routing with inferred demand to solve the bike sharing rebalancing problem. Presented at INFORMS Annual Meeting 2014 San Francisco, CA, US.
- Regué, R.** and Recker, W., (2014). Using Gradient Boosting Machines to Predict Bikesharing Station States. 93rd Annual Meeting of Transportation Research Board, Washington, D.C., US.
- Allahviranloo, M., **Regué, R.** and Recker, R., (2014). Pattern Clustering and Activity Inference. 93rd Annual Meeting of Transportation Research Board, Washington, D.C., US.
- Regué, R.**, (2013). An Integrated Approach to Improve Bike-Sharing Rebalancing Operations. Presented at 3rd International Sub-Forum of Transportation Science and Engineering, Beihang University, Beijing, China.
- Regué, R.** and Recker, W., (2013). An Integrated Approach to Improve Bike-Sharing Rebalancing Operations. Presented at INFORMS Annual Meeting 2013 Minneapolis, MN, US.
- Yang, M., **Regué, R.**, Luong, T. and Recker, W. (2013) Visualizing and Understanding Lane Changing Maneuvers in the Vicinity of Freeway-to-Freeway Connectors. Presented at 7th International Symposium on Visualization in Transportation, Irvine, California, US.
- Hernandez, S. and **Regué, R.**, (2013). Using Signature-Based Vehicle Re-identification to measure lane-changing maneuvers. Presented at the 92nd Annual Meeting of Transportation Research Board, Washington, D.C., US.
- Regué, R.** and A.L. Bristow (2012). Exploring Scenarios for the Introduction of Freight Trams in Barcelona. Presented at the 91st Annual Meeting of Transportation Research Board, Washington, D.C., US.
- Regué, R.**, and A. L. Bristow (2011). An exploration of the potential feasibility of a freight tram scheme in Barcelona. In 2011 European Transport Conference Proceedings, Glasgow, UK.
- Regué, R.**, and F. Astals (2010). Tranvía para mercancías basado en una plataforma convencional de transporte para viajeros. Presented at Congreso de Ingeniería del Transporte. "IX Congreso de Ingeniería del Transporte". Instituto de Investigación Aplicada del Automóvil, Madrid, Spain

LANGUAGES

English: Professional Working Experience

Spanish: Native

Catalan: Native

ABSTRACT OF THE DISSERTATION

Modeling Shared-use Urban Mobility Systems to Increase System Performance

By

Robert Regué Griñó

Doctor of Philosophy in Civil Engineering

University of California, Irvine, 2015

Professor Will Recker, Chair

Shared-use mobility systems, which enable users to have short-term access to transportation modes on an on-demand basis, have experienced tremendous growth over the last decade. However, most of the existing systems suffer from two confounding issues: the lack of modeling tools to understand, simulate and predict their behavior and the lack of integration with the existing transit network. To address those issues, this dissertation focuses on investigating the operational challenges of bikesharing systems, with an emphasis on the rebalancing operations and the modeling of a new mobility concept, Car2work, which builds upon existing carsharing ideas and successfully integrates with existing transit networks. A methodological framework to solve the bikesharing rebalancing problem is proposed. The novelties of the approach are that it is proactive instead of reactive, as the bike redistribution occurs before inefficiencies are observed, and uses the outputs of a demand-forecasting technique to decompose the inventory and the routing problem. The decomposition makes the problem scalable, responsive to operator inputs, and able to accommodate user-specific models. Simulation results based on data from the Hubway bikesharing system show that system performance improvements of 7% in the afternoon peak could be achieved.

Car2work main goal is to connect commuters with workplaces while leveraging the line-haul capabilities of existing public transit systems and guaranteeing a trip back home, efficiently tackling the “last mile” problem that is a limiting characteristic of public transit. It differs from the traditional dynamic-ridesharing approaches because it is designed for recurrent commuting trips where commuters announce their (multiple) trips in advanced and an automated all-or-nothing matching strategy is performed, guaranteeing a ride home. The problem is formulated as a pure binary problem that is solved using an aggregation/disaggregation algorithm that renders optimal solutions. The solution approach is based on decomposing the problem into a master problem and a sub-problem, reducing the number of decision variables and constraints. As a result, various instances of the problem can be solved in reasonable amount of time, even when considering the transit network. The model can be used to simulate a widespread implementation of the concept.

1. INTRODUCTION

Over the last decade there has been a surge in shared-use mobility concepts. As defined in Shaheen & Chan (2015), shared-use mobility is “an innovative transportation solution that enables users to have short-term access to transportation modes on an as-needed basis.” Included under this broad definition of shared-use mobility, one can consider any means of transportation that serves the above purpose, e.g., carsharing, peer-to-peer carsharing, bikesharing, ridesharing, ridesourcing, among others, as a type of shared-mobility service.

The underlying reasons for this growth can be sought on the need for a more sustainable and efficient transportation system that responds to pressing demands from the users, requesting cheaper, convenient and flexible transportation services, from the society, as the environmental concerns gain awareness, and because of the advent of technological developments that have enabled and simplified communication and management of such systems. Among those developments, the smartphone and electronic and wireless communications have had a major impact, reducing the barriers and the pain points for users to massively accept shared-use mobility as an alternative.

Several benefits have been reported to result from such a systems, including: car-ownership and vehicle usage reduction, increased network connectivity and encouraging multi-modality. Reduced emissions, traffic congestion are often reported as benefits, however the effects of those benefits are yet to be quantified (Chan & Shaheen 2012). For example, Shaheen & Chan (2015) report that one carsharing vehicle replaces between 9 to 13 vehicles, and that 25% of the people that joined a carsharing program sold their private automobile. In terms of bikesharing, on the same report it is stated that 50% of bikesharing users reduced private

automobile usage and bus usage increased overall, mostly due to the increased accessibility to the transit network offered by the bikesharing system.

At the user level, participants of such systems benefit from shared costs of travel, travel time-savings and preferential access to carpool lanes or parking. However, there are still several behavioral barriers that need to be overcome to bring shared mobility to mainstream use.

To overcome those barriers (accessibility, convenience, lack of flexibility, privacy or safety) providers are taking different approaches to shared-use mobility alternatives.

On the ridesourcing space, perhaps the most successful experience is Uber, which has suffered an exponential growth disrupting the taxi industry completely offering on-demand mobility through a mobile application. In addition, even though Uber and Lyft started by offering on-demand mobility, they quickly launched Uberpool and Lyft Line to let their customers share rides, services referred to as ride-splitting. According to the Shared Use Mobility Center (2015), ridesharing and ridesourcing are one of the most viable and rapidly growing areas of shared mobility, which can bring shared-use mobility to mainstream. Other alternatives in similar spaces exist, mostly led by startups that are exploring with innovative solutions. For example, Leap, Chariot, Bridj and Via offer an alternative to transit by providing on-demand, flexible private bus lines. ZipCar provides short-term car rentals; Carma, Zimride, and BlaBlaCar allow users to log their trips so that other users can find matches, and Scoop automatically creates carpools on a per-trip basis focusing on regular commuters. There are, of course also traditional vanpool and carpool services that co-workers arrange by themselves.

The bikesharing space has also evolved quickly since the first bikesharing systems appeared on the early 60's in Netherlands (Chan & Shaheen 2012). For example, in 2005 there were only 74 bikesharing systems across the world and today, as of 2015, there are more 770

(Shared Use Mobility Center 2015). A traditional modern bikesharing system operates as follows: a member can pick up a bike from any of the stations available in the system and must return it before a predefined time period to any other station that has empty docks available. Stations have a fixed capacity and a time limit is imposed to ensure high bike usage and bike rotation. Newer alternatives are also being tested, promoting dockless systems, for example.

Shared-use mobility systems need to gain enough scale and successfully integrate with the existing transportation network to be appealing to users, however, its scalability and multi-modality lead to operational challenges that need to be addressed. The focus of this dissertation is on addressing operational challenges on bikesharing systems generated by the need to reposition bikes across the system and the modeling of a new mobility concept that builds upon existing carsharing ideas and that integrates with existing transit networks. The dissertation, therefore, has been divided into two main sections.

Section 2 addresses the bikesharing-rebalancing problem, covering the history, trends and evolution of bikesharing systems and why a rebalancing system is required. The goal is to define a framework that is modular in design, scalable, flexible enough to accommodate different bikesharing operator needs, that can be used as a simulation tool to aid in operational decision making and that is able to respond proactively to different types of events.

Section 3 focuses on the new shared-use mobility system that is proposed, Car2work. Car2work integrates with the existing transit network and specifically targets commuters. The goal is to conceptually design the system, model it as a mathematical problem and propose a methodology to solve it efficiently.

2. A BIKESHARING REBALANCING FRAMEWORK

2.1. Introduction and motivation

Bikesharing is a sustainable and environmentally friendly transportation mode that offers bikes “on-demand” to improve daily urban mobility. Since its inception, bikesharing systems have evolved considerably. In its origins, bikesharing systems did not have locking stations and the bikes were spread around a given area and free to use. These types of systems are referred to in the literature as the first generation bikesharing. The second generation introduced docking stations to resolve vandalism and theft. Bikes had to be borrowed and returned to the stations to recover a deposit. However, vandalism and theft was not eradicated due to the anonymity of the users. The third generation introduced information technologies into bikesharing systems. RFID, electronic payments and wireless tracking technologies are used to improve user accountability and to monitor the system. The third generation is the most widely implemented nowadays. Finally, fourth generation bikesharing systems include demand-responsive rebalancing, dynamic pricing, and integration with other modes of transportation. This generation is still in early development (Shaheen et al. 2014).

For the purpose of this work, we assume that a typical current bikesharing system operates as follows: a member can pick up a bike from any of the stations available in the system and must return it before a predefined time period to any other station that has empty docks available. Stations have a fixed capacity and a time limit is imposed to ensure high bike usage and bike rotation.

In the United States, in 2012 there were 15 IT-based bikesharing programs (Shaheen et al. 2012) and by December 2014 the number increased to 68 (Shaheen & Chan 2015), including

major cities like New York, Chicago or San Francisco. Similar trends are observed around the world (Meddin & DeMaio 2007; Shaheen et al. 2012; Shaheen et al. 2014).

As reported by Shaheen et. al (2014) bikesharing systems have the potential to reduce fuel consumption and traffic congestion in urban areas, increase mobility and network connectivity, lower transportation costs and promote healthier lifestyles. The combination of benefits and the technological developments that have reduced the friction in bikesharing systems explain their rapid growth across the globe.

However, although bikesharing systems potentially offer a viable alternative for enhancing urban mobility, they suffer from the effects of fluctuating spatial and temporal demand that inherently lead to severe system inefficiencies; e.g. having empty or full stations for long periods of time. Figure 1 (left) shows the spatial imbalance for the Bicing bikesharing system in Barcelona on a given day at 11PM. Observe the clusters of stations in red representing full stations. Figure 1 (right) shows the mean number of bikes at station 14 for The Hubway bikesharing system in Boston for a given time period during 3 months. Note the different trends during week days and weekends, and the remarkably similar patterns during weekdays.

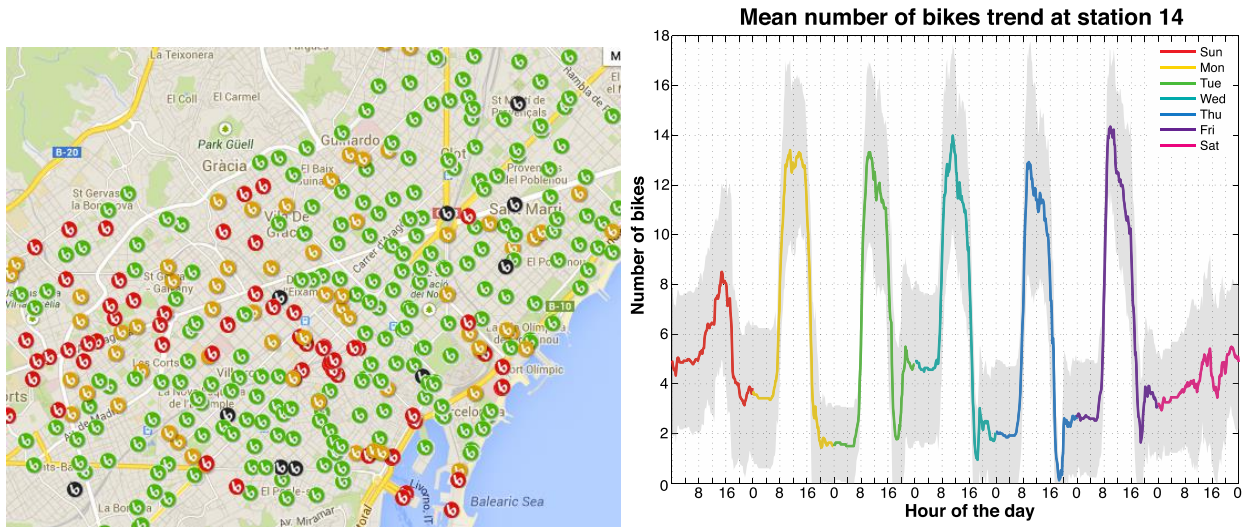


Figure 1: Spatial imbalance at Bicing, Barcelona at 11PM (left). Temporal imbalance at Hubway, Boston (right)

The inefficiencies are embedded in the fabric of bikesharing because one-way trips are allowed and the operator has little control over the behavior of the users. As a result, some stations are empty and some others are full, impeding potential users to either pick up or drop off bikes at their desired stations, degrading the level of service, system performance and causing disappointment that may result in loss of users. To resolve these inefficiencies, bikesharing operators are compelled to reposition bikes dynamically to avoid the system from collapsing (Fricker et al. 2012). For example, in The Hubway bikesharing system in Boston, during the early stages of the system when it had 61 stations and 600 bikes, 420 bikes were relocated daily on average, with peaks of more than 1000 bikes. Bikesharing operators use different methods to reposition bikes. In New York, under some heavily congested routes it is more efficient to use rickshaws that can move up to three bikes at a time (Figure 2 (left)). Other commonly used vehicles are small vans that can carry more bikes as seen in Figure 2 (right).



Figure 2: Different repositioning mechanisms. Rickshaws (left), vans (right).

Generally speaking, removing the inefficiencies from the system is the goal of the bikesharing rebalancing problem, which tries to answer the three following questions:

- 1) Which stations have or will suffer from inefficiencies?
- 2) How many bikes need to be added or removed to any given station and when?
- 3) Which is the optimal route of the repositioning vehicle?

The research presented here outlines a comprehensive framework to solve the dynamic bikesharing rebalancing problem — finding the optimal routes and inventory levels to keep the bikesharing system balanced while it is in operation (Contardo et al. 2012; Caggiani & Ottomanelli 2012; Rainer-Harbach et al. 2013; Raviv & Kolka 2013; Schuijbroek et al. 2013).

The framework is based on five core models:

- 1) Demand-forecasting model at the station level
- 2) Station inventory model
- 3) Redistribution needs model
- 4) A user-rebalancing model
- 5) A vehicle routing model

The dynamic bikesharing rebalancing problem, from a mathematical perspective, can be seen as variation of the One-Commodity Pickup-and-Delivery Vehicle Routing Problem (1-PDTSP) (Hernandez-Perez & Salazar-Gonzalez 2004) with the added complexity that the inventory at the stations is flexible (Schuijbroek et al. 2013). The proposed methodology is seen as an heuristic to solve such a problem that on its core uses anticipated future demands to decouple the inventory and the routing problem, reducing the complexity, making it scalable, proactive instead of reactive and allowing for real time decision-making. Vehicle routes are being built dynamically based on current and expected events in a proactive manner, as inefficiencies are resolved before they actually occur, increasing customer satisfaction. As routes are being built periodically, operator interaction is permitted, overriding current routing decisions if necessary.

The routing problem maximizes the utility gained by removing inefficiencies from the system, it is selective (not all stations are visited), keeps track of the vehicle inventory, can handle a non-homogenous fleet and allows for pick ups or drop offs at buffering stations—stations from which some bikes can be removed or added without causing future inefficiencies—solving the issue of having an empty or full vehicle that is not able to respond to existing inefficiencies.

The proposed predictive model has the potential to help determine more efficient user-based relocation policies by means of incentives or dynamic pricing policies. It is also self-adaptive, as it is regularly retrained as new system data are being acquired. Further enhancements to the predictive module can be made if bikesharing system users data were available—for example, offering discounts to users that express the need of a bike at a given

station using a mobile application. Doing so can improve the predictive model and lead to better routing decisions.

The framework has been tested under various simulation scenarios with variable time steps using data from The Hubway bikesharing system in Boston (Hubway 2011). The simulation results show that level of service can be improved compared to the “do nothing” scenario, especially in reducing the observed number of full and empty events. Managerial decisions are also simulated, testing for the impact of the number and the capacity of the vehicles used for rebalancing operations.

2.2. Related Work

Bikesharing-related literature has been growing as more and more systems are being implemented. Relative to issues addressed, there are two main areas of interest: forecasting future demands in shared-use systems, and approaches to formulate and solve the dynamic bikesharing rebalancing problem.

2.2.1. Demand Forecasting

Concerning forecasting future demands, a variety of techniques have been explored. Initial insights can be found in the carsharing literature, which has a longer history of investigation. Barth & Todd (1999) show under a simulation framework that for a one-way carsharing system the knowledge of future demands significantly impacts performance measures. Four different predictive techniques are tested on real data from the Honda Intelligent Community Vehicle System (ICVS) (Kek et al. 2005): Neural Networks (NN), regression, selective moving average and Holt’s model. The results indicate that NN has the best performance. Based on this research, Cheu et al. (2006) ran tests on an expanded dataset of ICVS

comparing NNs and Support Vector Machines (SVM). Their results also show that NNs lead to better performance and it is argued that they can better capture nonlinearities in the system. These results motivated the later implementation of a decision support system to optimize operator-based relocation operations in carsharing systems (Kek et al. 2009), which is modeled as a variation of a pick-up and delivery problem.

In the bikesharing literature, Froehlich et al. (2009) and Kaltenbrunner et al. (2010) use data from the *Bicing*, the bikesharing system in Barcelona (Spain). Froehlich et al. (2009) test four different predictive techniques: last value, historic mean, historic trend and a Bayesian Network (BN). The best results are obtained with the BN, with an average error of 8%, averaged over all days and prediction windows used (10, 20, 30, 60, 90 and 120 minutes). As expected, prediction errors increase with the prediction window. Kaltenbrunner et al. (2010) implement an Auto-Regressive Moving Average (ARMA) model with an FIR low-pass filter to predict station states. Mean absolute errors in a 60-minute prediction window of 1.39 bikes with a maximum error of 6 bikes are reported.

Borgnat et al. (2011) use data from Vélo'v, Lyon's (France) bikesharing system to predict hourly rentals. The problem is decomposed into first predicting the non-stationary amplitude in a day and then adding hourly fluctuations. The amplitude is modeled as a linear regression using external factors to account for time and season. Hourly fluctuations are modeled as an autoregressive process of order 1 with exogenous inputs. Apparently, they are the first to include explanatory variables other than those related to historical system data in their models.

Caggiani & Ottomanelli (2012) use NN to forecast arrivals and departures for their decision support system that solves the bikesharing rebalancing problem. NNs are selected on the

grounds that previous studies (Kek et al. 2005; Cheu et al. 2006) demonstrated its applicability. The independent variables set includes weather conditions.

More recently, Henderson & Fishman (2013) use Poisson regressions to predict arrivals and departures for the next 60 minutes and estimate the probability of stations being empty or full in the future. The model is tested on Divvy Bikes, the bikesharing system from Chicago.

The above approaches to predict bikesharing station states do not include external factors as independent variables. Only two studies, Borgnat et al. (2011) and Caggiani & Ottomanelli (2013) test for weather and holidays, showing improvements of more than 50% in predictive power when those factors are included. Gebhart and Noland (2013) study the impact of weather on trips made at Capital Bikeshare, in Washington DC, considering the precise weather conditions when the trip is made and conclude that fewer trips are made in rainy, humid, windy and cold conditions.

Another common assumption is that the same explanatory variables work for all stations, meaning that the same model can be used to make forecasts for individual stations. However, the behavior of bikesharing stations is extremely dynamic and that behavior need not be correlated among them. As a result, either a variable selection step specific to each station should be introduced or the predictive technique should internally account for that.

Additionally, most of the current predictive models do not attempt to be used as inputs to the bikesharing rebalancing problem, but rather attempt either to explain bikesharing usage and riders' behavior (Froehlich et al. 2009; Borgnat et al. 2011) or to provide information to the users and operators about the system (Kaltenbrunner et al. 2010) such that better design and planning policies can be implemented.

2.2.2. Rebalancing Problem

The bikesharing-rebalancing problem is typically decomposed into the static and the dynamic rebalancing and can be seen as a variation of the pickup and delivery problem (PDP). Static rebalancing occurs during periods where demand is negligible (overnight) and the main goal is to set the stations to the optimal inventory level such that dynamic repositioning is minimized when the system is in operation. Most of the focus in the literature has been on formulating and solving the static rebalancing problem (Rainer-Harbach et al. 2013; Nair et al. 2013; Chemla et al. 2012).

In terms of finding optimal inventory levels, Raviv & Kolka (2013) introduce a dissatisfaction measure based on the number of renters and returners that abandon the system. The measure is a function of the initial inventory level at the beginning of each study period and the goal is to find the optimal inventory level such that the dissatisfaction is minimized. The dynamics of the station inventory are modeled as a continuous time Markov chain. The estimated inventory levels are tested on the Tel-O-Fun bikesharing system in Tel Aviv (Israel) by computing the initial inventory of each station at the beginning of the day. The performance of the proposed inventory levels is measured based on the number of shortages observed during the day and compared to current practices. Results indicate a 17% reduction of observed shortages during the day, reducing the cost of the dynamic repositioning.

A different approach to determine optimal inventory levels is presented in Lu (2013) . A bikesharing system is represented using a time-space network flow model and the goal is to minimize the system total cost, including holding cost at station, bike supply costs, redistribution costs and the cost of losing customers due to unmet demand. The solution to the problem is the optimal bike allocation and the redistribution flows. To account for demand uncertainty, robust

optimization techniques are introduced. Numerical tests are carried out using the bikesharing system in Banciao District, in New Taipei City, Taiwan.

Schuijbroek et al. (2013) combine both the inventory and the routing problems into a single framework. Optimal inventory levels are first estimated from historical data. Each station is individually modeled as a $M/M/1/K$ queuing model. The inventory is then fed to the routing problem, minimizing the maximum tour length of the vehicles. The problem is solved in a rolling horizon fashion using a cluster-first, route-second heuristic. The model performance is tested on The Hubway and Capital Bikeshare bikesharing systems from Boston and Washington, respectively.

Caggiani and Ottomanelli (2013) also address both problems proposing a decision support system that considers stochastic demand and minimizes vehicle reposition cost. The day is divided into fixed time intervals. At the beginning of each time interval future demands are updated and the decision support system is launched. The outputs of the model are the relocation matrix and the relocation path of the redistribution vehicles that minimize the total operator costs, including relocation costs and lost users costs. The model is tested on a 5-station bikesharing system discretizing the day into 5-minute intervals under three different demand scenarios. The results indicate that significant reductions of lost users can be achieved.

Pfrommer et al. (2013) use model-based receding horizon optimization techniques to combine operators' relocation operations and user-based relocations by means of a reward system. The goal is to use users to do the repositioning instead of the operator. Truck routes and incentives are computed online at periodic time instances and a predictive model is introduced to estimate the expected evolution of the system in the near future. Truck routes are also modeled under a time-expanded network and the solution approach is based on a greedy heuristic. The

model was tested on data from the London's Barclays Cycle Hire and results indicate that price incentives can be used to reduce system inefficiencies, but that they are not enough to increase service levels considerably during weekday operations.

The focus of this dissertation is on the dynamic problem. The goal is to find the optimal inventory levels at each station and the repositioning vehicles optimal routes to redistribute bikes across the system.

Current methodologies rely on solving a variation of the pickup and delivery problem (PDP) to compute optimal vehicle routes. However, PDP is an NP hard problem that becomes intractable for large bikesharing systems. As a result, current practice consists of finding heuristics that can quickly find a near-optimal solution to the routing problem for a given time period. Here, a different approach to reduce the complexity of the problem building on the idea of proactive dynamic vehicle routing (Ferrucci 2013) and the use of machine learning techniques is proposed. First, different datasets to infer future station inventory levels are combined. Then, based on the inferred inventory levels, a stochastic linear integer problem is solved to determine the estimated number of bikes required at the stations with inefficiencies. Once pickup and drop off events are identified, they are treated as deterministic and are the input to a vehicle routing model that optimally routes vehicles based on anticipated events. The vehicle routing problem is solved for a single vehicle at a time and is limited to those stations that have inefficiencies and are within threshold travel time of the current station, bounding the size of the problem and allowing the use of traditional solvers to find a solution.

The proposed proactive vehicle routing approach differs from current proactive routing approaches such as the hybrid predictive control (Núñez et al. 2013) or the real-time control and request-forecasting (Ferrucci 2013). The previous approaches focus on real time deliveries and

paratransit type problems and incorporate into the objective function expected future events. Vehicles are routed through areas with higher probabilities for events to occur in future time steps so that when an event occurs, vehicles are able to respond faster. In the proposed approach this idea is also included into the objective function of the vehicle routing problem. However, the bikesharing rebalancing problem requires the solution of an inventory model first. Therefore, future expected demands are not only used in the routing problem but they are also used to anticipate inventory needs, successfully combining machine learning techniques with mathematical programming. As a result, vehicles are routed before events occur, leading to a truly proactive approach.

2.3. Proposed Framework

The proposed framework has five main models: 1) a demand forecasting model at the station level, 2) a station inventory model, 3) a redistribution needs model, 4) a user rebalancing model and 5) a vehicle routing model. The relationship between the various models and the available data in most new-generation bikesharing systems is shown in Figure 3. Dashed boxes represent available data that serves as an input to the models.

Historical data is fed into the demand forecasting model and the inventory levels model. Details on the data needs are given in section 2.3.1, but historical data generally refers to the bike counts time series prior to the current time and other such related data as weather and transit data that may improve the accuracy of the forecasting model.

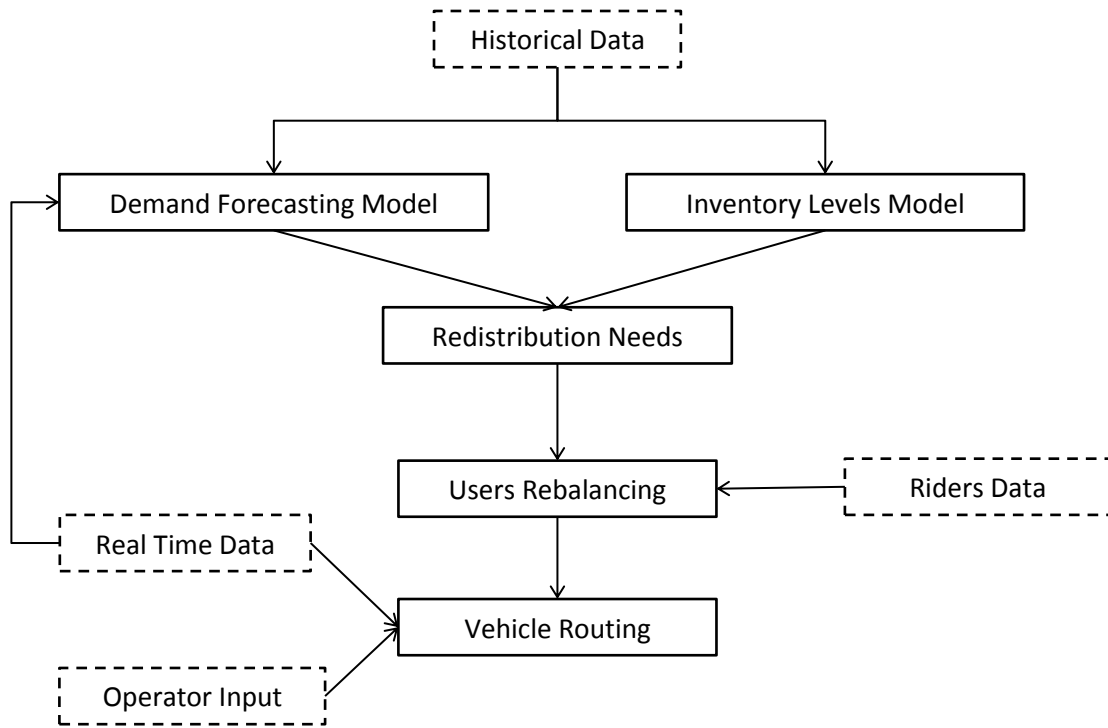


Figure 3: Bikesharing rebalancing framework

The goal of the forecasting model is to anticipate inventory levels (number of bikes at a given station) at different time windows from the current time; for example, after 20, 40 and 60 minutes, based on current system states. The inventory levels model uses historical data to determine the optimal initial inventory level such that in the next time period (next 20 minutes) system inefficiencies—defined by either pickup or drop off events—will not occur.

The output of these two models is used to estimate the system redistribution needs. In other words, for each station in the system the number of bikes that should be picked up or dropped off such that at the beginning of the next time period the station will have the initial optimal inventory level is estimated. If riders’ data were available by matching user ids with bike travel patterns, a user-based relocation procedure could be incorporated, taking the redistribution needs outputs as an input. As future inefficiencies are anticipated, it should be possible to contact users in advance and offer them an incentive to change their behavior. This model is optional and

it is left to the bikesharing operator to decide whether or not implement it. The goal of this model is to reduce the redistribution needs by incentivizing users to change their behavior. The final step of the process is to dispatch and route the available vehicles to address system inefficiencies. In this later step, system operators can interact by altering vehicle-dispatching decisions based on their own expertise or unpredicted events that the demand-forecasting model cannot anticipate. Note that the vehicle routing and the user rebalancing can be exchanged. It is possible to first solve the vehicle routing and then the user-rebalancing problem. This decision is an operational decision that should be based on the cost structure of the bikesharing operator.

The demand forecasting and inventory levels models are periodically retrained, as system data are being acquired. For example, the predictive model can be updated weekly using data from the past week to train the new model.

The proposed framework is seen as a heuristic by itself that successfully decouples the inventory and the routing problem and that intrinsically handles the vehicle decomposition in the routing problem. As a result, the framework is scalable, both, in terms of the number of stations and vehicles used for the repositioning. As detailed in coming sections, the routing problem is bounded and the other three models, the rebalancing needs model (P1), the station inventory model and the predictive model are defined at the station level and solved for a single station at a time.

Detailed descriptions on each model are found in the following subsections.

2.3.1. Data Description

The dataset from The Hubway bikesharing system in Boston is used. The Hubway was launched in July 28, 2011 with 600 bikes and 61 stations (Hubway 2011). The data were made publically available for The Hubway Data Visualization Challenge (Hubway 2012) and include:

- *Trip history data*: all trips made by users detailing length, departure and arrival times and origin and destination stations.
- *Rebalancing operations aggregated at the daily level*: the number of bikes that were picked up and dropped off for rebalancing purposes in a given day.
- *Station snapshot data*: number of available bikes and docks per station in 1-minute intervals.
- *Other related data*: census information, neighborhoods, elevation, employment characteristic, population, vehicle miles traveled, etc.

Figure 4 depicts the stations on The Hubway bikesharing system on 2012, when the data used was collected.

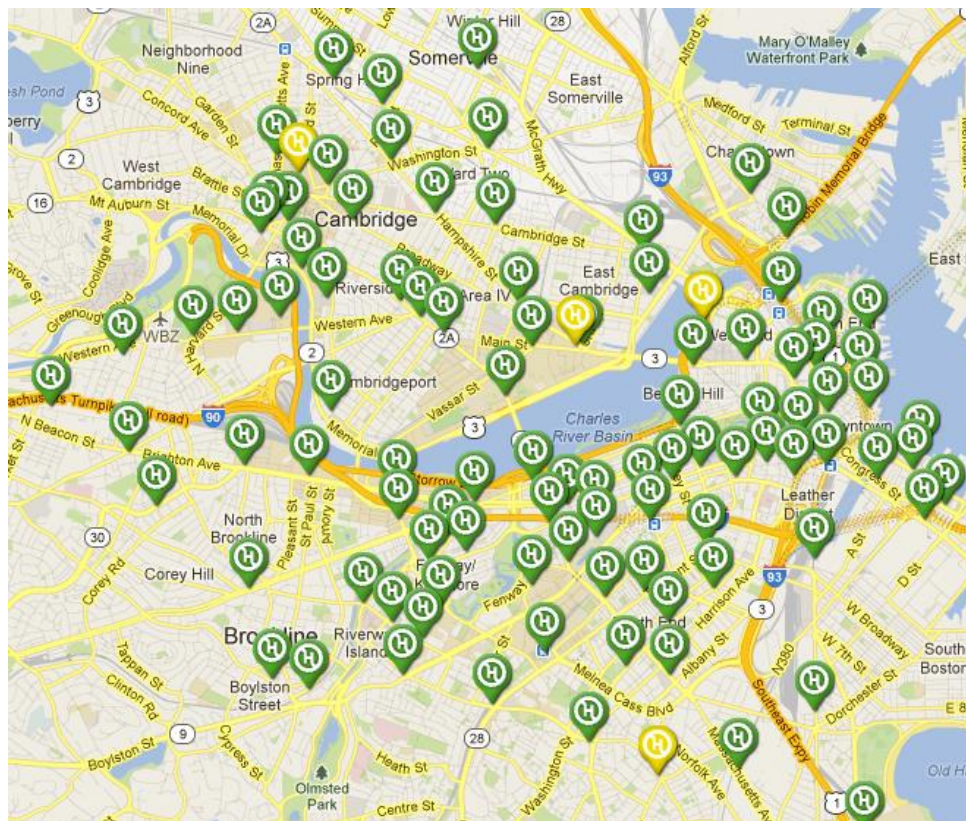


Figure 4: The Hubway bikesharing system (2012)

In this study only the trip history data and the station snapshot data from Sunday May 6th to Sunday July 29th 2012 is used, a period in which the number of working stations was constant at 61 stations (during the following months the system gradually expanded, having 108 stations and more than 1000 bikes before it was shut down for the winter season in October 2012.). For training, validating, and testing the models the original dataset was first split into training, validation and test datasets. The validation data are used to validate the model trained on the training data set. Data from Sunday July 15th to Saturday July 21st (1 week) are considered for validation purposes. The test dataset, which has been excluded from previous model calibration steps, goes from Sunday July 22nd to Saturday July 28th (1 week).

It should be noted that the station snapshot data already contains the rebalancing operations undertaken by the operator.

The dependent variable represents the number of available bikes and it is drawn from the station snapshot data and grouped into 10-minute intervals. The mean of the observations that fall within the 10-minute intervals is then taken.

The Hubway data are merged with hourly weather data for station 14739 at Logan International Airport from the National Climatic Data Center (National Climatic Data Center 2012), and with sunset and sunrise data from (Time and Date AS 1995)—in Gebhart & Noland (2013) the authors claim that bike riders behavior is affected by daylight and darkness.

As a result, all of the available variables that are used in the demand-forecasting model are shown in Table 1

Variable	Description	Source
Number of bikes	Number of bikes at the station at current time t	Hubway
Number of bikes at Station X	Number of bikes at surrounding stations X at current time t	Hubway
Number of bikes at Delta Y	Number of bikes that were present at current station, Y hours before the prediction (ie. -1hr, -24hr, -168hr)	Hubway
Hourly Rain	Dummy variable (0,1) that indicates if it rained during current time t	NCDC
Hourly Drizzle	Dummy variable (0,1) that indicates drizzling during current time t	NCDC
Hourly Shower	Dummy variable (0,1) that indicates if it shower during current time t	NCDC
Hourly Temp	Mean temperature in Fahrenheit for current time t	NCDC
Snowfall	Dummy variable (0,1) that indicates if it snowed during the day	NCDC
Hourly Humidity	Relative humidity for current time t	NCDC
Wind Speed	Average wind speed in miles per hour	NCDC
Visibility	Statute visibility in miles	NCDC
Daylight	Dummy variable (0,1) that indicates if it was daylight or dark.	time & date
Weekday	Categorical variable representing the weekday (1-Mon...7-Sun). It can optionally be set to 0 or 1 to disaggregate between week days or weekends	
Hour	Hour of the current time t	
Minute	Minute of the interval for the current time t	
US Holidays	Dummy variable (0,1) that indicates if a given day was an official holiday	Google Calendar
Station Activity	Standard deviation of the number of bikes at the current station during the last 6 time intervals	

Table 1. Variable Definition. NCDC: National Climatic Data Center

Other variables that could be incorporated into the dataset and potentially increase the predictive power are detailed below. However, those are not considered because of a lack of data availability or the difficulties of merging time dependent historical data, such as the arrival or departure of transit systems.

- *Transit arrivals*: variable that accounts for the number of transit arrivals in the surrounding transit stations of a bikesharing station.
- *Indicator variables to capture special events*: indicator variables may be introduced to monitor particularities of the system. For example, major events that occur in a weekly based such as soccer/football games can modify the travel patterns of particular stations.

- *Indicator variables to capture location characteristics*: land use data could be used to classify bikesharing stations into different classes depending on their location.
- *Near freeways traffic flow*: traffic flow in freeways can be a good indicator of the number of people getting into or leaving the city and used as a mobility indicator. The more people in the city, the more users of the bikesharing system that can be expected.
- *Underground/metro counts*: number of users currently in the transit system can also be a measure to capture mobility in the city.
- *Cellphone towers information*: cell phone carriers collect data about the number of calls a particular tower is serving. This information may be useful to identify clusters of people around the city that may be potential bikesharing users.

Figure 5 shows the mean number of bikes over the study period at two different stations of The Hubway bikesharing system. The shaded area corresponds to one standard deviation. Note the completely different behavior between both stations. Whereas station 14 has a traditional “central business district behavior” with low levels of bikes over night and larger levels during working hours, station 15 has a relatively flat behavior throughout the day and week. It is also worth noting the different patterns observed on weekends and weekdays for station 14. All weekdays follow the same trend, which differs from the weekend trend, being the later flatter.

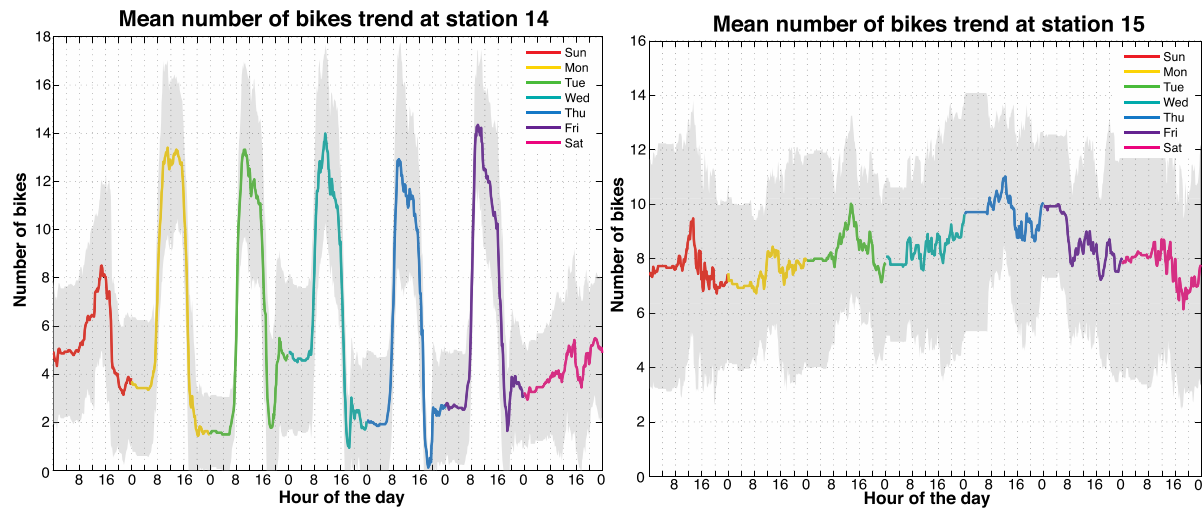


Figure 5: Mean number of bikes with 1 standard deviation for Station 14 and Station 15

2.3.2. Demand Forecasting Model

A perfect demand forecasting model would be such that can make predictions with high accuracy, it is fast to train and predict, that it is not data intense and that can be easily retrained overtime as new data is being gathered. Traditionally, the field of transportation has been reticent to the implementation of the so called “black box” forecasting techniques, or machine learning, mostly because of the difficulties on understanding the meaning of the parameters of the models and the fact that this types of models are not policy sensitive, in the sense that different scenarios cannot be modeled by changing the values of a parameter. However, for the particular application of forecasting bikesharing station states with the purpose to cut relocation costs, the above issues do not apply since parameters do not necessarily need to be meaningful and the data set is being constantly updated. As a result, “black box” models can be used and retrained to account for seasonal, weekly, daily or hourly variations and capture recent behavioral changes of bikesharing users. The main issues of the approach are the computational time, since a trade off has to be made in terms of how in advanced predictions should be made and the data availability

in a particular time period. For instance, data gathered over the last hour may bring insightful information about the state of the system in the coming hour. However, if the forecasting algorithm training is time consuming, that piece of data may not be used until a later time step forecast. Similarly, if there are not enough observations in the defined time period, prediction errors may increase considerably.

In this work, the performance of Gradient Boosting Machines (GBM), Neural Networks (NN) and linear regression (LR) is compared.

2.3.2.1. Gradient Boosting Machines

Gradient boosting machines (GBM) were introduced by Friedman (Friedman 2001; Friedman 2002) and it is a supervised and regression-based machine learning method that repeatedly fits a weak classifier (typically a decision tree) and ensembles them to make the final prediction.

The use of GBM in the transportation literature is limited. However, it has been successfully implemented to enhance reliability in real-time risk assessment (Ahmed & Abdel-Aty 2013) and to forecast traffic flow under abnormal conditions (Wu et al. 2012).

The main advantages of using GBM, as detailed in Friedman (2001), are that its performance is invariant to transformations in the explanatory variables and insensitive to outliers. Furthermore, variable selection is internalized in the decision tree, making the algorithm robust toward irrelevant input variables, and there is no need to impute missing values since the decision tree can also handle them. Also, overfitting is avoided by carefully selecting the regularization parameters: learning rate (ν) and number of iterations (M). Another advantage is that there is no need to retrain the entire model when new data are acquired, as the boosting procedure can continue from the previous model, saving computational time and making GBM

attractive for online applications. Finally the importance of the explanatory variables can be measured, easing the interpretation of model results and coefficients.

As in any regression method, the goal of GBM is to find the model $F(\tilde{\mathbf{x}})$ that minimizes the expectation of the loss function $\Psi(y, F(\tilde{\mathbf{x}}))$ given a dataset $D = \{(\tilde{\mathbf{x}}_1, y_1), \dots, (\tilde{\mathbf{x}}_N, y_N)\}$, with input, or explanatory, variables $\tilde{\mathbf{x}}$ and output variable, or prediction, y (Eq. 1). The estimated model $\hat{F}(\tilde{\mathbf{x}})$ is found by the weighted sum of weak models $h(\tilde{\mathbf{x}}_s, a_m)$. In this study h is a decision tree in which the input variable $\tilde{\mathbf{x}}_s$ is a sub sample of $\tilde{\mathbf{x}}$ drawn at random and a_m are the parameters of the m th decision tree (Eq. 2). Random sampling introduces stochastic behavior, which reduces computational times and improves accuracy (Friedman 2002).

$$F^*(\tilde{\mathbf{x}}) = \arg \min_{F(\tilde{\mathbf{x}})} E_{y, \tilde{\mathbf{x}}}[\Psi(y, F(\tilde{\mathbf{x}}))] \quad (\text{Eq. 1})$$

$$\hat{F}(\tilde{\mathbf{x}}) = \sum_{m=1}^M \beta_m h(\tilde{\mathbf{x}}_s, a_m) \quad (\text{Eq. 2})$$

The optimal weights β_m are derived in a greedy approach, minimizing (Eq. 3) following a forward stagewise procedure. Starting from a constant function, $F_0(\tilde{\mathbf{x}})$, the new model is updated using (Eq. 4). The shrinkage parameter, ν , is a regularization parameter that controls for overfitting. Typically, smaller values of ν lead to better estimates, but there is a tradeoff with the number of iterations M . Both parameters need to be calibrated using cross-validation or using an independent dataset not used in training the algorithm.

$$(\beta_m, a_m) = \arg \min_{\beta, a} \sum_{i=1}^N \Psi(y_i, F_{m-1}(\tilde{\mathbf{x}}_i) + \beta h(\tilde{\mathbf{x}}_i, a)) \quad (\text{Eq. 3})$$

$$F_m(\tilde{\mathbf{x}}) \leftarrow F_{m-1}(\tilde{\mathbf{x}}) + \nu \beta_m h(\tilde{\mathbf{x}}_s, a_m) \quad (\text{Eq. 4})$$

In this implementation, the R package *gbm* by Ridgeway (2012) and the Gaussian loss function have been used.

2.3.2.2. *Neural Networks*

Neural networks (NN) or Artificial Neural Networks (ANN) are another commonly used machine learning technique that offers great flexibility and that were initially designed to mathematically represent information processing in biological systems (Bishop 2006). NNs in its essence try to model how the neurons in our brain operate and McCulloch and Pitt in 1943 (McCulloch & Pitts 1943) proposed the first abstract model of a neuron. NN are a parameterized function that takes a set of inputs and maps them to a set of outputs as shown in (Eq. 5) where y represents the output, x the input, w a set of weights, $g(\cdot)$ a non-linear activation function (that is usually a sigmoid) and b that is a bias term.

$$y(x, w) = g\left(\sum_i w_i x_i + b\right) \quad (\text{Eq. 5})$$

Due to its representation, NN are a flexible way to model input/output functions. To be able to learn non-linear decision surfaces multiple layers are introduced, in the form of hidden units, leading to the multilayer neural nets. A representation of a two layer NN is depicted in Figure 6.

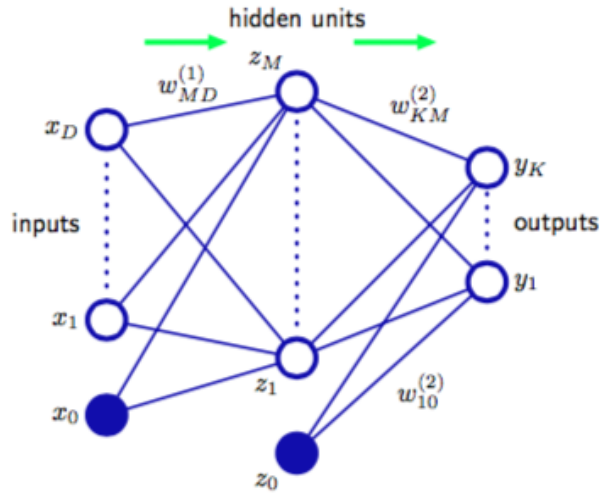


Figure 6: Network diagram for a two-layer neural networks (Bishop 2006).

One of the challenges of NN is the determining the optimal set of parameters, as it involves the solution of a non-linear optimization problem. However, an efficient technique has been developed, named error back propagation that can efficiently evaluate the derivatives. The advantages of the NN are that they are really flexible, are robust against noisy data and learning can be fast. However, the weights are difficult to interpret and the optimization problem solved to train the NN has a lot of local minima.

NNs have been implemented using the Neural Network Toolbox in MATLAB using the Gaussian loss function.

2.3.2.3. Calibration

As described on section 2.3.1 Data description, the original dataset was first split into training, validation and test datasets.

Having defined the three datasets, a calibration procedure has been designed to simultaneously calibrate the algorithmic parameters and these other parameters that modify the data set. The goal of this calibration step is to find an optimal combination of parameters such

that the error observed on the test set is minimum. The implemented calibration process goes as follows. A set of parameters that modify the dependent (y) and independent variables (\tilde{x}) is first defined. The aim of those parameters is to search over all possible combinations to obtain the best model at each particular station. The rationale is that the behavior among stations varies significantly, as observed in Figure 5, showing the mean number of bikes over the study period at two different stations.

The initial set of parameters that modify the data set include:

- *Number of Neighboring Stations (K):* neighboring stations are selected based on the activity between station pairs and not based on Euclidian distance between stations. Neighboring stations are ranked based on the sum of incoming and outgoing trips from and to the station under study. The number of bikes at current time t for the K first stations is included as an independent variable.
- *Weekdays:* it is observed that some stations have the same behavior across weekdays and weekends but, in others, Mondays and Fridays have different patterns. To account for this, weekdays can be binary or categorical, accounting for weekend or weekdays or each day of the week separately.
- *Removing the mean:* removes the mean of the dependent variable.
- *Filtering:* the dependent variable is filtered with a one-sided moving average of size τ . In some stations with smooth behavior, better results are achieved if the predictive algorithm is trained filtering the dependent variable.
- *Seasonality:* is a binary parameter that includes seasonality effects. Periodograms are used to compute the main frequency (f) at which the series oscillates and two new

independent variables are added: $\cos\left(\frac{2\pi f}{T}t\right)$ and $\sin\left(\frac{2\pi f}{T}t\right)$, where T is the maximum number of observations and t is the current observation.

- *Significant variables*: only features significant at the 95% confidence level from a linear regression are included. The set of independent variables is cleaned including only significant features from a stepwise backward linear regression.

A total of 18 models for the 61 stations are run and errors and optimal calibration parameters are obtained.

The effect of the parameters defined above, together with any other algorithmic parameters, are calibrated using a differential evolution algorithm inspired by Braak & Vrugt (2008) with an extension to accommodate integer and binary variables. As a fitness measure, the root mean squared error (RMSE) (Eq. 6) is used. Maximum error (Eq. 7) and mean error (Eq. 8) are also monitored. The fitness measure is computed on the predictions obtained on the validation data set. Once a model is validated, its performance is tested on the test set and new error measures computed. Three different time windows are tested: 20, 40 and 60 minutes. The same procedure is applied to Gradient Boosting, Linear Regression and Neural Network methods. The results obtained after the calibration step are also compared to the results obtained using a fixed set of parameters for all stations, neglecting the calibration step.

$$RMSE = \sqrt{\frac{\sum(y - \hat{y})^2}{n}} \quad (\text{Eq. 6})$$

$$MaxError = \max(|y - \hat{y}|) \quad (\text{Eq. 7})$$

$$MeanError = \frac{\sum|y - \hat{y}|}{n} \quad (\text{Eq. 8})$$

The differential evolution algorithm is coded in MATLAB and goes as follows:

1. For all time windows in tw
2. For all stations in S
3. Define number of chains to evolve N_c
4. Initialize parameter population $X(N_c \times d)$ using latin hypercube sampling (LHS)
5. Set X to bounds
6. Evaluate fitness function $f(x_i)$ for all x_i in X
 - 6.1. Train Model
 - 6.2. If GBM, compute optimal M
 - 6.3. Compute RMSE on validation dataset
7. Update Population:
 - 7.1. Sample uniformly at random without replacement two numbers r_1 and r_2 from $1, 2, \dots, N_c$.
 - 7.2. Calculate proposal point x^* as: $x^* \leftarrow x_i + k_1(x_{r_1} - x_i) + k_2(x_{r_2} - x_i)$
8. Crossover operation. If $rand(1, d) > P_{cr}$ set $x^* \leftarrow x_i$
9. Mutation operation. If $rand(1, d) < P_m$ set $x^* \leftarrow x_{new}$
10. Set x^* to bounds
11. Evaluate fitness function of proposal point $f(x^*)$
12. Acceptance. If $f(x^*) < f(x_i)$ accept and set $x_i \leftarrow x^*$ else keep x_i in population X
13. Convergence. If $iter > maxIter$ or $std(X) < \epsilon$ or $|\max(f(X)) - \min(f(X))| < \epsilon$
14. If not converged go to step 4.

2.3.3. Inventory Levels Model

Initial inventory levels are computed borrowing the concept of level of service described by Schuijbroek et al. (2013). The underlying idea is to determine whether or not the initial inventory level in a given station is enough to serve the future demand for a certain time period. Each station is modeled as a M/M/1/K process, K being the capacity of the station. Arrivals and departures are assumed Poisson and the parameters of the Poisson distribution are estimated from historical data. The outputs are $s_{i,t}^{min}$ and $s_{i,t}^{max}$, which represent the minimum and maximum inventory levels at station i and time t such that if the initial inventory level $s_{i,t}$ falls

between that range, no action is required with some level of confidence for pickups β_i^+ and for drop offs β_i^- . Table 2 summarizes the different events that can occur.

Initial Inventory level	Event type
$s_{i,t} \leq s_{i,t}^{min}$	Drop off
$s_{i,t}^{min} < s_{i,t} < s_{i,t}^{max}$	Self-sufficient station
$s_{i,t} \geq s_{i,t}^{max}$	Pickup

Table 2: Initial inventory levels and event types

Compared to Schuijbroek et al. (2013) where data from the early stages of the Hubway are used (November 1st, 2011 to May 31st, 2012) and the time interval is set to 1 hour, here the time interval is set to 20 minutes. Note that in the current dataset the activity in the system is greater, as the system was well established and more trips are observed, allowing for a shorter time window. Service level requirements parameters (β_i^-, β_i^+) are set to 95% for all stations.

As an example of the disparate nature of activity among the various stations, Figure 7 shows inventory levels as computed in (Schuijbroek et al. 2013) relative to the station capacity at stations 14 and 38, together with the observed number of bikes on July 20th 2012. For the computation of inventory levels, only weekday data between 6:00 and 23.59 have been used. Outside this interval there are not enough trip observations to properly estimate the arrival and departure pattern distributions. Note the different behavior of each station and how throughout the day, pickup and drop off events should occur to bring the station into a balanced state. Station 14 has most of the arrivals during the AM peak where the maximum inventory threshold is lower. Inventory levels for station 38 vary during the day due to its higher activity. Despite the large number of docks available, station 38 gets empty at the end of the morning peak (9:30 am) and full during the PM peak. As a result, the minimum inventory threshold is higher during the AM peak and relatively constant the rest of the day. The maximum inventory level is lower at the beginning of the PM peak, so that future arrivals can be accommodated.

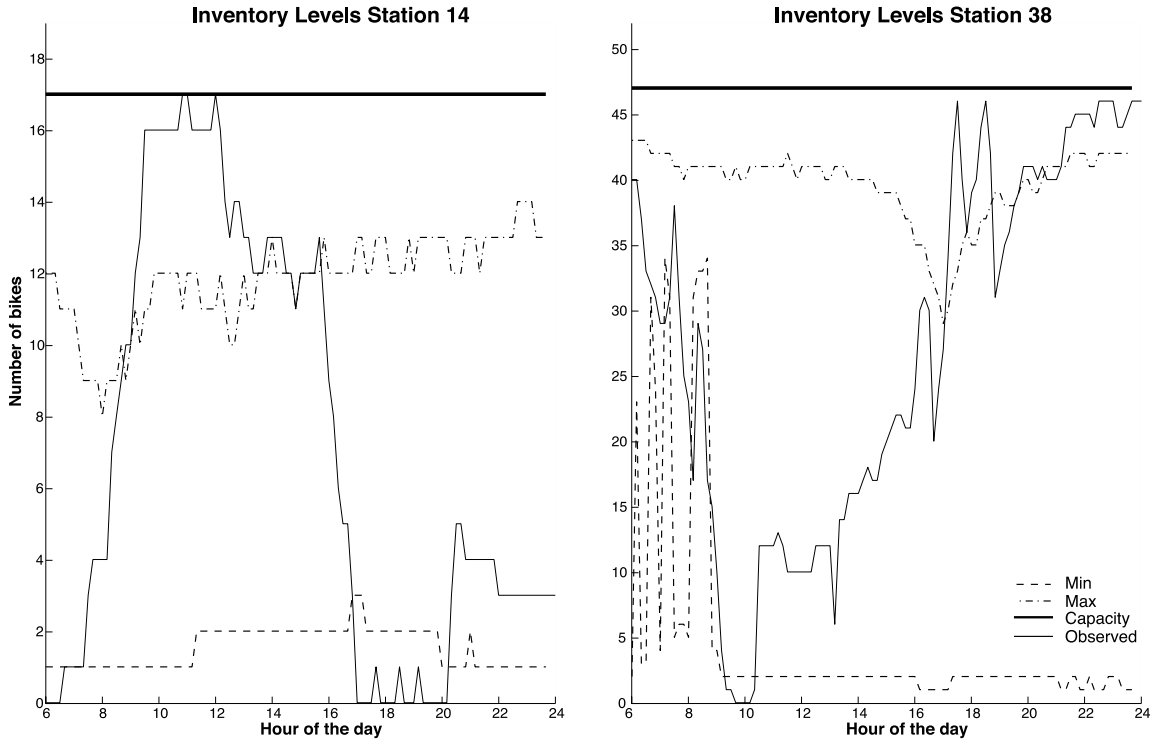


Figure 7. Inventory Levels for Station 14 and Station 38

As discussed in Schuijbroek et al. (2013), although none is observed in their data set, three different types of infeasibilities can occur. In the current dataset situations where the maximum inventory level is smaller than the minimum, $s_{i,t}^{max} < s_{i,t}^{min}$ are observed. This implies that given the current arrival and departure patterns the station cannot be balanced. In such cases operators need to prioritize. From an operational point of view, operators tend to focus on ensuring that docks are available instead of bikes, as riders may incur late fees if they go over the maximum riding time. These singularities are observed in stations with high activity during peak hours and small capacities (stations 35, 41, 47 and 58). In such events we set $s_{i,t}^{min} = s_{i,t}^{max}$.

2.3.4. Redistribution Needs Model

Decisions on the number of bikes to be picked up ($y_{i,t}^-$) or dropped off ($y_{i,t}^+$) are made solving a stochastic linear integer program that takes as input inventory levels and predictions at

different time windows, both being the outputs of the two previous models. Predictions ($\hat{s}_{i,t}$) are treated as normal distributed variables with mean the prediction and variance equal to the variance of the error made in the predictions for the test set.

The logic behind this step is to ensure that by the beginning of the next time window (in this case 20 minutes) all pickups and drop offs such that all of the stations are in balance are known (i.e., the initial inventory level $\hat{s}_{i,20}$ falls between the range $s_{i,20}^{min} < \hat{s}_{i,20} < s_{i,20}^{max}$ for all stations i). Note that in here the current observed inventory at the stations nor the arrival or departure patterns are used. The interest is on making sure that at the beginning of the next time period the number of bikes at the stations will fall between the optimal inventory ranges.

Instead of simply checking predictions $\hat{s}_{i,20}$ that do not fall in the optimal range and setting them to the boundary (i.e., $y_{i,t}^+ = s_{i,20}^{min} - \hat{s}_{i,20}$ or $y_{i,t}^- = \hat{s}_{i,20} - s_{i,20}^{max}$ for all stations i), a linear problem is formulated in such a way that future predictions $\hat{s}_{i,40}$ and $\hat{s}_{i,60}$ are considered in determining $y_{i,t}^-$ or $y_{i,t}^+$. In this manner, the information gained from the predictive model is used. This can be observed in Figure 8, where in the particular case represented, the goal is to find the number of bikes to be removed (y^-) from the station before t equals 20 minutes such that system inefficiencies in later time steps do not occur. In this process it is assumed that demand is constant during the study period. As a result, the number of bikes to be picked up or dropped off during the time period of analysis is minimized together with the probability to revisit one station in future time steps.

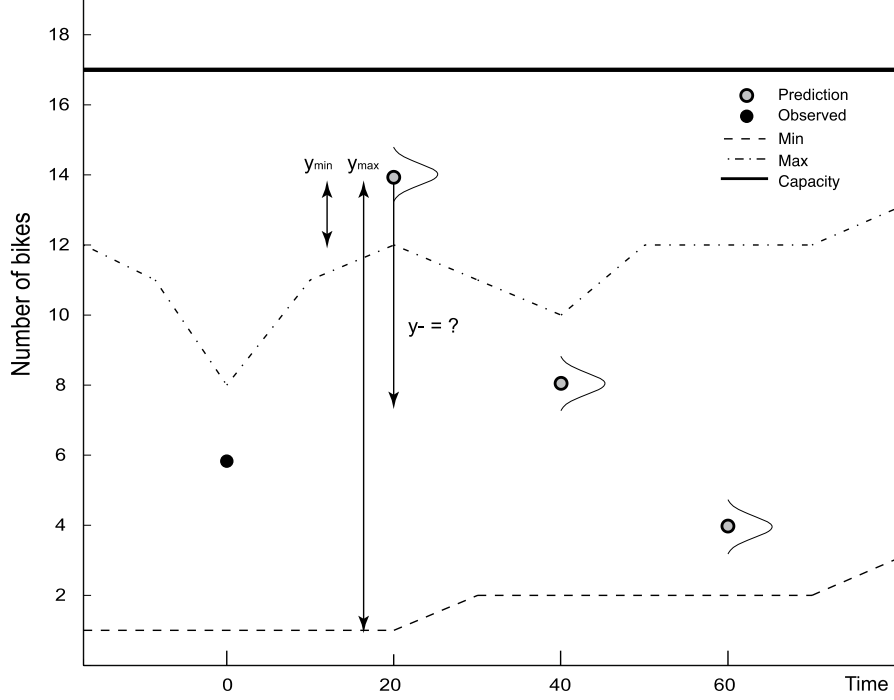


Figure 8. Redistribution needs example

The problem is formulated as a chanced constrained problem with stochastic constraints having a confidence level α and independently solved for each station. As a result, for each station i in the set of stations \mathcal{S} problem P 1: is solved.

P 1:

$$\min \sum_{t \in \mathcal{T}} \omega_t \cdot (y_{i,t}^+ + y_{i,t}^-) \quad (1.0)$$

st.

$$\Pr \left\{ s_{i,t}^{\min} - \hat{s}_{i,t} - \sum_{k=1}^t (y_{i,k}^+ - y_{i,k}^-) \leq 0 \right\} \geq \alpha, \quad \forall t \in \mathcal{T} \quad (1.1)$$

$$\Pr \left\{ -s_{i,t}^{\max} + \hat{s}_{i,t} + \sum_{k=1}^t (y_{i,k}^+ - y_{i,k}^-) \leq 0 \right\} \geq \alpha, \quad \forall t \in \mathcal{T} \quad (1.2)$$

$$\Pr \{ y_{i,t}^- - \hat{s}_{i,t} \leq 0 \} \geq \alpha, \quad \forall t \in \mathcal{T} \quad (1.3)$$

$$\Pr \{ y_{i,t}^+ - C_i + \hat{s}_{i,t} \leq 0 \} \geq \alpha, \quad \forall t \in \mathcal{T} \quad (1.4)$$

$$y_{i,t}^-, y_{i,t}^+ \in \mathbb{N}$$

where the set $T = \{20,40,60\}$ is the set of time windows and C_i is the capacity of station i

Weights (ω_t) are introduced in the objective function to penalize early deliveries, being larger for closer time windows ($\omega_1 > \omega_t > \omega_T$). This is to avoid picking up or dropping off bikes in earlier time steps than when they are needed. For example, if $s_{i,60}^{max} - \hat{s}_{i,60} = -3$, the desired solution would be $y_{i,60}^- = 3$, but $y_{i,20}^- = 3$ and $y_{i,40}^- = 3$ are also feasible solutions. If all weights are equal to one, they all have the same objective function value, 3. Introducing the weights pushes the solution to be $y_{i,60}^- = 3$.

The first two constraints ensure that after picking up or dropping off bikes the number of remaining bikes will be between the maximum and minimum inventory ranges. The last two constraints ensure that more bikes than the ones that are currently in the stations are not picked up and that more bikes than the docks available are not dropped off.

The above chanced constrained problem can be turned into its deterministic equivalent and solved with traditional linear programming solvers.

Under some circumstances, especially when a station has a highly dynamic behavior, a solution may not exist to problem P 1:. If this is the case, we set $y_{i,20}^-$ to $\hat{s}_{i,20} - s_{i,20}^{max}$ or $y_{i,20}^+$ to $s_{i,20}^{min} - \hat{s}_{i,20}$.

Using the above concept it is also possible to determine the number of bikes that can safely (i.e., without causing the station to become ‘inefficient’) be picked up or dropped off from a station that is not expected to have inefficiencies. This set of stations is referred to as the buffering stations set S^B —the set of stations that can be visited by the relocation vehicles if they run out or have an excess of bikes. For each station h in S^B , the maximum number of bikes that can be drop off z_h^+ and the maximum number of bikes that can be picked up z_h^- are defined.

2.3.5. Vehicle Routing Model

Vehicle routing is done solving the problem P 2: for a single vehicle each time it completes the previous tour. It takes as an input the results from the redistribution needs model to define a set of candidate stations that can be visited. The candidate stations are selected under the following criteria:

- 1) are within TT_{max} minutes reach of the current station,
- 2) where the expected shortage or excess of bikes is larger than a specified threshold, and
- 3) that are not being served by another vehicle.

Preprocessing the station set and solving the vehicle routing problem for a single vehicle reduces the problem size, allowing the use of traditional solvers to find the solution in real time.

The problem is formulated using an arc and sequence-indexed formulation to keep track of vehicle inventory:

P 2:

$$\max U = U_I + U_N + U_B \quad (2.0)$$

$$U_I = \sum_{i \in \mathcal{S}, j \in \mathcal{S} \setminus \{S^B, S^0\}, k \in \mathcal{K}} \frac{|u_j|}{C_j} \cdot x_{ij}^k \quad (\text{Eq. 9})$$

$$U_N = \sum_{i \in \mathcal{S}, j \in \mathcal{S} \setminus \{S^B, S^0\}, k \in \mathcal{K}} \beta_j \cdot x_{ij}^k \quad (\text{Eq. 10})$$

$$\beta_j = \sum_{n \in \mathcal{N}_j} \sum_{m \in \mathcal{T}} \frac{y_{n,m}^+ + y_{n,m}^-}{C_n} \quad (\text{Eq. 11})$$

$$U_B = \frac{\gamma}{TT_{max}} \cdot \left(- \sum_{i \in \mathcal{S}, j \in S^B, k \in \mathcal{K}} tt_{i,j} \cdot x_{ij}^k - \sum_{i \in S^B, j \in \mathcal{S}, k \in \mathcal{K}} tt_{i,j} \cdot x_{ij}^k \right) \quad (\text{Eq. 12})$$

s.t.

$$\sum_{i, j \in \mathcal{S}, k \in \mathcal{K}} (tt_{i,j} + c) \cdot x_{ij}^k \leq TT_{max} \quad (2.1)$$

$$\sum_{j \in \mathcal{S}} x_{S^0, j}^1 = 1 \quad (2.2)$$

$$\sum_{i \in \mathcal{S}, k \in \mathcal{K}} x_{i, \mathcal{S}^0}^k = 0 \quad (2.3)$$

$$\sum_{i \in \mathcal{S}, j \in \mathcal{S}} x_{i, j}^k \leq 1 \quad \forall k \in \mathcal{K} \quad (2.4)$$

$$\sum_{i \in \mathcal{S}} x_{i, j}^k \geq \sum_{i \in \mathcal{S}} x_{j, i}^{k+1} \quad \forall j \in \mathcal{S}, k \in \mathcal{K} \setminus \{\tau\} \quad (2.5)$$

$$\sum_{i \in \mathcal{S}, k \in \mathcal{K}} x_{i, j}^k \leq 1 \quad \forall j \in \mathcal{S} \quad (2.6)$$

$$\sum_{i \in \mathcal{S}, k \in \mathcal{K}} x_{i, i}^k = 0 \quad (2.7)$$

$$D^k = D^{k-1} - \sum_{\forall i \in \mathcal{S}, j \in \mathcal{S} \setminus \{\mathcal{S}^0, \mathcal{S}^B\}} u_j \cdot x_{i, j}^k - \sum_{h \in \mathcal{S}^B} z_h^k \quad \forall k \in \mathcal{K} \quad (2.8)$$

$$- \sum_{i \in \mathcal{S}, j \in \mathcal{S} \setminus \{\mathcal{S}^0, \mathcal{S}^B\}} x_{i, j}^k u_j - \sum_{h \in \mathcal{S}^B} z_h^k + D^{k-1} \leq V_{cap} \quad \forall k \in \mathcal{K} \quad (2.9)$$

$$\sum_{i \in \mathcal{S}, j \in \mathcal{S} \setminus \{\mathcal{S}^0, \mathcal{S}^B\}} x_{i, j}^k u_j + \sum_{h \in \mathcal{S}^B} z_h^k \leq D^{k-1} \quad \forall k \in \mathcal{K} \quad (2.10)$$

$$\sum_{n \in \mathcal{S}} x_{n, h}^k - \sum_{n \in \mathcal{S}} x_{h, n}^{k+1} = 0 \quad \forall h \in \mathcal{S}^B, k \in \mathcal{K} \setminus \{\tau\} \quad (2.11)$$

$$\sum_{n \in \mathcal{S}} x_{n, h}^\tau = 0 \quad \forall h \in \mathcal{S}^B \quad (2.12)$$

$$z_h^k \geq z_h^- \cdot \sum_{i \in \mathcal{S}} x_{i, h}^k \quad \forall h \in \mathcal{S}^B, k \in \mathcal{K} \quad (2.13)$$

$$z_h^k \leq z_h^+ \cdot \sum_{i \in \mathcal{S}} x_{i, h}^k \quad \forall h \in \mathcal{S}^B, k \in \mathcal{K} \quad (2.14)$$

$$D^k \geq 0 \quad k \in \mathcal{K} \quad (2.15)$$

$$x_{i, j}^k \in \{0, 1\} \quad \forall i, j \in \mathcal{S}, k \in \mathcal{K} \quad (2.16)$$

$$D^k \in \mathbb{Z} \quad \forall k \in \mathcal{K} \quad (2.17)$$

$$z_j \in \mathbb{Z} \quad \forall j \in \mathcal{S}^B \quad (2.18)$$

In the formulation above:

x_{ij}^k : Binary variable that indicates if the vehicle traverses the arc i, j at time period k .

z_h^k : Integer variable representing the number of bikes picked up or dropped off at buffering stations $h \in \mathcal{S}^B$ at time k , and is bounded by z_h^+ and z_h^- , determined in the rebalancing needs model.

D^k : Integer variable to keep track of vehicle load at time k .

$tt_{i,j}$: Travel time from i to j .

c : Fixed loading and unloading cost.

C_i : Station capacity.

u_j : Difference between number of bikes drop off and picked up. It is defined as $u_j = y_{j,20}^+ - y_{j,20}^-$, being $y_{j,20}^+$ and $y_{j,20}^-$ the outputs of the previous model. Note that u_j can be positive or negative, depending on if it is a drop off event or a pick up event, respectively and that a simultaneous pick up or drop off cannot occur at the same station.

τ : Parameter that defines the number of visited stations. Setting τ to 1 makes P2 a purely dispatching problem, where the vehicle will be dispatched to the station that maximizes the utility. Under the current scenario, we set τ to 3, as it is unlikely that a vehicle can visit more than 3 stations in TT_{max} minutes.

TT_{max} : Maximum travel time for a vehicle when is being routed. It is currently set to 20 minutes.

γ : Dimensionless parameter that accounts for the relative importance of the travel time utility term.

The following sets are defined:

\mathcal{K} : Set of time steps $\{1, \dots, k, \dots, \tau\}$.

S^+ : Set of stations with drop off events.

S^- : Set of stations with pick up events.

S^B : Set of buffering stations.

S^0 : Current station.

S : Set of all stations $\{S^+ \cup S^- \cup S^B \cup S^0\}$.

\mathcal{N}_j : Set of neighboring stations of station j .

T: Set of time windows {20,40,60}.

The objective function $U ()$ is a combination of three elements: 1) the utility gained by visiting a station with a large inefficiency U_I (Eq. 9), 2) the utility gained by visiting a station with a neighborhood of stations (\mathcal{N}_j) that is expected to have inefficiencies in future time steps U_N (Eq. 10), and 3) in the case that buffering stations need to be visited, minimize the travel time involved in going to those stations, U_B (Eq. 11).

The utility of visiting a station with a large inefficiency is measured as the ratio between the inefficiency and the station capacity. It has been observed that using a ratio leads to better repositioning results rather than using the inefficiency measured in number of bikes. The second utility term aims to route the current vehicle to a region that is expected to have inefficiencies in the future. To measure this term, the neighborhood of station j , \mathcal{N}_j , is defined as all the stations that are within 0.5 miles from the current station j . The sum over all time windows $T = \{20,40,60\}$ of the expected inefficiencies ($y_{i,t}^+, y_{i,t}^-$) is divided by the capacity of station (C_i).

The last term is introduced as a penalty for visiting buffering stations. If a penalty were not added, vehicles may be routed through buffering stations without any need to. This term is normalized dividing by the maximum travel time permitted TT_{max} , which in this problem equals 20 minutes. The coefficient γ is introduced to change the relative importance of the penalty. For example, setting γ to TT_{max} makes the penalty one order of magnitude larger than the other utility terms, meaning that if there is any other alternative rather than visiting a buffering station, independent of the magnitude of the inefficiency, the buffering station will not be visited. Sensitivity tests on γ have been done and shown in the numerical examples.

Constraint (1) ensures the routing is completed by the next simulation time. Constraints (2) to (7) take care of the routing. Constraints (2) and (3) ensure that the vehicle leaves the depot

(its location at the start of the period) and that it will not come back to it in later time steps k . Constraint (4) enforces that only one trip per time step is allowed, constraint (5) is the connectivity constraint and we also enforce that stations cannot be revisited in (6).

Constraints (8)-(10) keep track of vehicle load and make sure that more bikes than the ones available are not dropped off or that vehicle capacity is not exceeded.

Constraint (11) forces buffering stations to be transshipment nodes and constraint (12) does not allow buffering stations to be demand nodes. Constraints (13) and (14) set the bounds on z_h^k if the corresponding buffering station is visited.

Problem P2 can be infeasible, a condition that occurs most frequently when the travel time constraint is violated because the vehicle does not have either enough bikes or empty spaces to serve an inefficiency and has to first visit a buffering station to either pick up or drop off bikes. Under such circumstance, the vehicle remains idle at its current station. The idle time is set to 10 minutes in current simulation scenarios.

Note that P2 is not formulated as a multi-period problem that uses information from previous models regarding future time steps. In other words, u_j is defined using only the estimated inefficiencies for the first time window. The underlying reasons to do so are because there is a trade off to be made which is due to the accuracy of the predictions and because future information is already accounted for when solving P1 to determine the inefficiencies for the first time window $(y_{i,20}^+, y_{i,20}^-)$.

Regarding the accuracy of the predictions, the longer the time horizon, the less accurate are the predictions. In fact, if the predictions were to be 100% accurate, the preferred approach would be to solve a multi-period problem using all the information from previous models. However, as the predictions are not 100% accurate, a heuristic would be needed to make sure

that current routes are the best routes possible given current, up to date, information. Instead of proposing such heuristic and the more complex multi-period problem, the problem P2 is proposed, which is solved more frequently with more recent information. In addition, the current approach may be more appealing and flexible from an operational perspective, as it is likely more responsive to re-routings implemented by the operations center and can better capture the “true” travel times observed from the network in real time, given that travel times can be obtained every time a tour ends (limited to 20 minutes) rather than after the 3 periods (limited to 60 minutes), if information from previous models were used.

An illustrative example is shown below to demonstrate the impact of the utility terms in the objective function of the routing problem. This example takes a subset of The Hubway system where stations 10, 46 and 27 have an estimated inefficiency of $u_{10} = -2$, $u_{46} = 4$ and $u_{27} = 6$ bikes respectively. Each station with inefficiencies has been assigned a neighboring set, being $\mathcal{N}_{10} = \{9,33\}$, $\mathcal{N}_{46} = \{21,57\}$ and $\mathcal{N}_{27} = \{30\}$ and they all have the same capacity $C_{10} = C_{46} = C_{27} = 20$. The vehicle is initially empty, $D^0 = 0$, to force visiting buffering stations. τ is set to 3 and TT_{\max} to 20 minutes.

Problem P2 is solved under three different scenarios:

- *Scenario 1*: Inefficiencies at later time windows are zero for all stations and $\gamma = TT_{\max}$
- *Scenario 2*: Inefficiencies at later time windows are zero for all stations and $\gamma = 1$
- *Scenario 3*: Station 57 has an expected inefficiency $y_{57,40}^- = 4$ at time window 40 minutes and $\gamma = 1$

Figure 9 depicts the different routing solutions obtained for each scenario. Note that when $\gamma = TT_{\max}$, the vehicle is routed to station 10, where there is a need to remove 2 bikes, even

though the inefficiencies at other stations are larger. Other stations are not visited because the vehicle does not have enough bikes and it does not have time to go through a buffering station. In scenario 2, instead, when setting γ to 1 we reduce the penalty on visiting buffering stations and the vehicle is now routed to station 27, which has the largest inefficiency ($u_{27} = 6$), after going through the buffering station 32 to pick up the required bikes. Even though it is not apparent from the figures, going to station 32 minimizes the buffering stations travel time.

Finally, in scenario 3 $y_{57,40}^-$ is set to 4, and now the neighborhood utility term comes into play. Under this case, the vehicle visits station 46, as it is likely that in the next iteration, station 57 would also need to be visited.

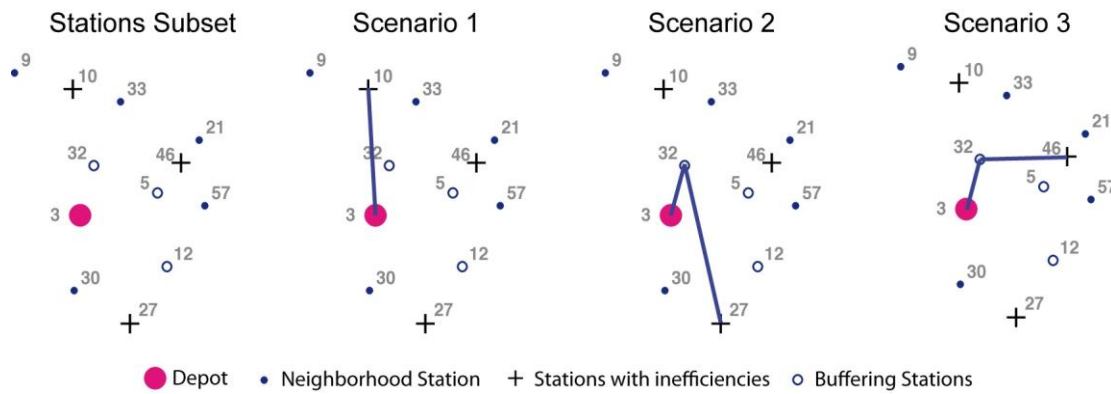


Figure 9: Vehicle routing illustrative example

2.3.6. User Relocation Model

The user relocation model is aimed to investigate the feasibility of using riders to help on the rebalancing efforts. It has been suggested in the literature that if a monetary reward is in place, bikesharing operators can force behavioral changes that can benefit the entire system. Under the proposed methodology the rebalancing needs are known in advanced, information that can be leveraged to target specific users and remove inefficiencies from the system. The purpose

of this model is not to determine the monetary reward that will encourage the behavioral change; instead the goal is to test the impacts on the overall system performance of implementing a user relocation model under different scenarios. It is left for future research how to target users and what should be the optimal incentive. To test the user model the following assumptions are made:

- Rebalancing needs are known in advanced (20 min) and deterministic.
- Only users that are already in the system, that already have a bike, can be targeted for repositioning. This assumption is imposed by the data, as the user behavior prior the use of the system is unknown.
- User re-routing does not add travel time. The expected arrival time to the suggested station is the same as for the original station the rider was heading to. The monetary reward should compensate for the rider inconvenience of making such a change.
- All candidate riders that are suggested a new destination will accept.

The user relocation model is modeled as an integer linear problem formulated as shown in P 3 below.

P 3:

$$\min Z = \sum_{\forall j \in S} h_j \quad (3.0)$$

s.t.

$$\sum_{\forall i \in \{N_s(j) \setminus S^+\}} x_{i,j} + h_j = u_j \quad \forall j \in S^+ \quad (3.1)$$

$$\sum_{\forall i \in \{N_s(j) \setminus S^-\}} x_{j,i} + h_j = u_j \quad \forall j \in S^- \quad (3.2)$$

$$\sum_{\forall i \in S_n} x_{j,i} \leq A_j \quad \forall j \in S_B \cup S^- \quad (3.3)$$

$$\sum_{\forall i \in S_n} x_{i,j} - \sum_{\forall i \in S_n} x_{j,i} \leq z_j^+ \quad \forall j \in S_B \quad (3.4)$$

$$\sum_{\forall i \in S_n} x_{i,j} - \sum_{\forall i \in S_n} x_{j,i} \geq -z_j^- \quad \forall j \in S_B \quad (3.5)$$

$$\sum_{\forall i \in S_n \setminus N_s(j)} x_{i,j} + \sum_{\forall i \in S_n \setminus N_s(j)} x_{j,i} = 0 \quad \forall j \in S \quad (3.6)$$

$$\sum_{\forall i,j \in S_B} x_{i,j} = 0 \quad (3.7)$$

$$\sum_{\forall i \in S_n} x_{i,j} = 0 \quad \forall j \in S^- \quad (3.8)$$

$$\sum_{\forall i \in S_n} x_{j,i} = 0 \quad \forall j \in S^+ \quad (3.9)$$

$$h_j - M \cdot (1 - B_j^1) \leq 0 \quad \forall j \in S \quad (3.10)$$

$$u_j - h_j - M \cdot (1 - B_j^2) \leq 0 \quad \forall j \in S \quad (3.11)$$

$$B_j^1 + B_j^2 = 1 \quad \forall j \in S \quad (3.12)$$

$$h_j, x_{i,j} \in \mathbb{N}_0, B_j^1, B_j^2 \in \{0,1\}$$

In the formulation above:

$x_{i,j}$: Integer variable that measures how many riders travel from station i to station j .

h_j : Integer variable that captures the slack on the balance constraints.

u_j : Inefficiency at station j in absolute value.

A_j : Number of expected riders arriving to station j .

z_j^+ : Maximum number of bikes that can be added to a station j .

z_j^- : Maximum number of bikes that can be removed j .

The following sets are defined:

$N_s(i)$: Set of neighborhood stations of station i for the riders.

S^+ : Set of stations with drop off events.

S^- : Set of stations with pick up events.

S : Set of stations with inefficiencies ($S = S^+ \cup S^-$).

S_B : Set of neighboring stations of all stations in S .

S_n : Set of a all stations.

The objective function Z (3.0) minimizes the slack variable h_j over all the station with inefficiencies S . The slack variable measures the difference between the predicted inefficiency u_j and the number of riders arriving to station j ($x_{i,j}$), from any other station i . If $u_j = h_j$, it is not feasible for the riders of the system to help on the rebalancing of station j . As a result, minimizing h_j , maximizes the number of inefficiencies resolved by the users. The remaining inefficiencies are left to the vehicle routing problem.

Constraint (1) imposes the balance on drop off stations. All the riders diverted to station j plus the slack has to be equal to the inefficiency, where the slack represents the new inefficiency after the user relocation. Constraint (2) does the same for pick up stations.

Constraint (3) limits the number users diverted for pickup and buffering stations to those that arrive during the time period (A_j). Constraints (4) and (5) are the balance equations for the buffering stations that are in the neighborhood of stations with inefficiency. After diverting riders, a station that was originally in balance cannot go off-balance.

Constraints (6) - (9) rule out illogical flows. A rider can only be diverted to a neighboring station of their original station. Trips between buffering stations are not allowed, drop off stations can only be demand nodes and pickup stations can only be supply nodes.

Constraints (10) to (12) enforce that whenever a rider is diverted to a station, there must be enough riders to resolve the inefficiency at that station, otherwise it is not desirable to divert a user. In other words, the slack variable h_j is either zero or the value of the inefficiency. The rationale behind this set of constraints is to avoid situations where to operators need to take two actions to resolve an inefficiency. For example, having to reward a user and having to route a vehicle to the same station.

A simple example, building on the routing example detailed in the previous section, is shown to demonstrate the outcomes of the user relocation model. The same initial conditions as before are assumed. Rider arrivals A_j are set to 2 for all stations in the neighborhood of stations with inefficiencies ($A_9 = A_{33} = A_{21} = A_{57} = A_{30} = 2$) and z_j^+ and z_j^- are set to 3 bikes for the same stations. For station 10, which has a pick up inefficiency, it is assumed that A_{10} is also 2, meaning that two riders are heading to station 10. The results of this scenario are shown in Figure 10, where two users are diverted from station 21 and station 57 to station 46, which has a drop off inefficiency of 4 bikes, the two users that were heading to station 10 are diverted to station 9 to resolve the inefficiency of 2 units at station 10 and the vehicle is routed through the buffering station 32 to pick up 6 bikes and resolve the drop off inefficiency at station 27. Note, that under this example, all inefficiencies in the system have been successfully resolved through the combination of the user and the routing rebalancing models.

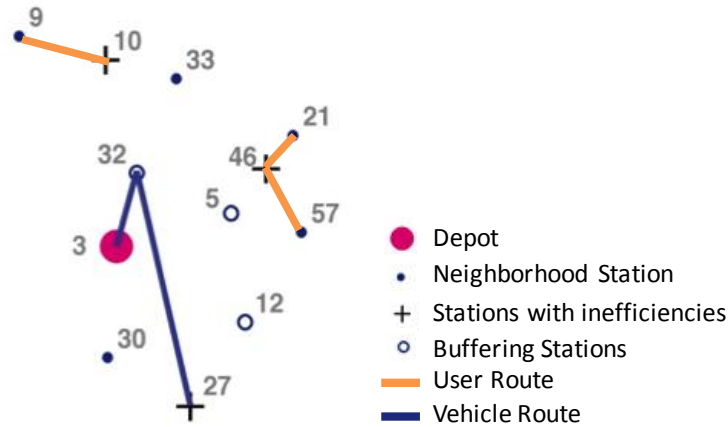


Figure 10: User relocation model outcomes

During the integration of the user model and the routing model it is necessary to update the demand matrix so the movement of users is also accounted for when computing performance measures. It should also be noted that implementing a user model changes the dynamics of the

system, as it did the routing, however under the user rebalancing model a higher number of stations are visited, and therefore affected by the assumption made that the rebalancing effects will last for an hour.

The proposed model imposes that all users have to accept the request to change their behavior. However, in a real scenario this is not a reasonable assumption to make. If a behavioral model were to exist, it could be possible to model the reward needed so that the balance between user and operation based relocation is optimal from a financial perspective. Such a model could be added before the user rebalancing model in such a way that only those users that are more likely to accept the reward are fed into the user rebalancing model.

2.4. Performance measures

The performance measures used to compare before and after simulation results are: system performance after (Eq. 13), system performance before (Eq. 14), system performance increase (Eq. 15), relative performance increase of empty events (Eq. 16), relative performance increase of full events (Eq. 17), the total number of bikes picked up, the total bikes dropped off, the total distance travelled, the duration of the inefficiencies and the number of users being diverted.

The subscript A is used to indicate “after” simulation and the subscript B for the “before” simulation. *Empty* indicates the number of time periods where a station was empty and *Full* the number of time periods when it was full. D refers to the mean duration of consecutive empty or full events.

$$SysPerf_A = \frac{(\#Observations - (Empty_A + Full_A))}{\#Observations} \quad (\text{Eq. 13})$$

$$SysPerf_B = \frac{(\#Observations - (Empty_B + Full_B))}{\#Observations} \quad (\text{Eq. 14})$$

$$SysPerf_{INC} = \frac{SysPerf_A - SysPerf_B}{SysPerf_B} \quad (\text{Eq. 15})$$

$$relEmpty = -\frac{Empty_A - Empty_B}{Empty_B} \quad (\text{Eq. 16})$$

$$relFull = -\frac{Full_A - Full_B}{Full_B} \quad (\text{Eq. 17})$$

$$D_{Full} = \frac{D_{Full_A} - D_{Full_B}}{D_{Full_B}} \quad (\text{Eq. 18})$$

$$D_{Empty} = \frac{D_{Empty_A} - D_{Empty_B}}{D_{Empty_B}} \quad (\text{Eq. 19})$$

2.5. Experimental Results

A MATLAB Graphical User Interface (GUI) has been created that takes the parameters listed in Table 3 as inputs to test for different scenarios and settings. A snapshot of the GUI is shown in Figure 11.

In the application, travel times are computed using the Google Distance Matrix API. For simplicity, a static OD distance matrix has been computed and a fixed vehicle speed is assumed. In online applications, observed travel times can be queried in real time and a dynamic OD travel time matrix that would reflect any congestion in the network could replace this static assumption.

Due to the different nature of pick up and drop of events when making predictions, combined with the relatively small $s_{i,t}^{min}$ threshold, the *minThreshold* parameter plays a significant role in the simulation outputs. Setting *minThreshold* to a large value considerably reduces performance in dealing with empty events. After sensitivity testing, *minThreshold* is set to 1.

The simulation flow goes as follows:

- 1) Set simulation parameters (Table 3).
- 2) Read the current system state and make predictions at 20, 40 and 60 minutes.

- 3) If first iteration, repeat steps 4 to 7 for all vehicles.
- 4) Solve the rebalancing needs model to find station inefficiencies and select buffering stations.
- 5) Solve user rebalancing model, if required.
- 6) Solve the routing problem.
- 7) Update system states.
- 7) Check for convergence: if convergence is not obtained go to step 2, otherwise dispatch vehicle to the initial depot and compute performance measures.

The convergence check consists of checking if the current simulation time of the vehicle has reached the end of the simulation period. There is also an end of day convergence check if multiple day simulations are considered, where the vehicle is dispatched to the initial depot and initial inputs are reset for the next day to start.

In order to update the state of the system based on the rebalancing events, an assumption about how to treat the latent demand is needed. This is because of the limitation imposed by applying the models to existing Hubway data (rather than to outputs from a real-time closed loop control system under actual operation). For demonstration purposes, it is considered here that one hour after the relocation event has occurred, the demand (or number of bikes) will be the same as that observed in the original data. Note that this assumption may lead to sudden jumps in the number of bikes observed in a station and therefore trigger a relocation event. Note also that in a real-time online scenario this assumption is not needed, as the input to the forecasting model would be the current state of the system.

Table 3 describes the parameters used on the simulation and the default values of each. It is worth highlighting the idling logic (*idleLogic*), which can be set to 0 or 1. If it is set to zero,

the vehicle will remain idling on the current station for a period of *idleTime*. Alternatively, if it is set to 1, the vehicle will be dispatched to the closest station. This later case is implemented to impede vehicles from getting trapped on stations under periods of time where the density of inefficiencies is low. Another parameter that changes the routing logic is the *Utility*, which allows selecting different combinations of objective functions for the routing vehicle. After sensitivity analysis, *idleLogic* is set to 0 and *Utility* to the max-ratio, which is the utility function that has been described in the routing problem section.

Parameter	Description	Default value
technique	Forecasting Method	'GBM'
solveUserModel	If user rebalancing model is used	False
timeWindows	Time windows for predictions in minutes	[20, 40, 60]
nv	Number of relocation vehicles	2
Vcap	Relocation vehicles' capacity	[20, 20]
inLoad	Initial load of each vehicle	[10, 10]
Depot	Station serving as a depot for each vehicle (selected at random)	[3, 3]
Day Start	Starting time for the vehicle	[7, 7]
Day End	Ending time for the vehicle	[21, 21]
minTreshold	If the redistribution need is below the threshold, it won't be considered in the dispatching algorithm	1
idleTime	Maximum time a vehicle can remain idling before a new dispatching decision is made in minutes	10
idleLogic	0 – Remain idle or 1 - dispatch vehicle to neighboring station	0
speed	Relocation vehicles speed (mph)	15
loadTime	Loading and unloading time of bikes in minutes per event	5 min
Utility	Combinations of names for the routing utility function	'max-ratio'
Tau	Maximum number of stations visited when routing	3
nStationDistance	Threshold distance to determine neighborhood stations in miles	0.5
inTime	Start time for the simulation	'2012-07-23 07:30'
finTime	End time for the simulation	'2012-07-28 00:00'
Stochastic	Redistribution needs are solved using stochastic approach	1
PlotRoutes	Routes are plotted in the UI	0
TT_{max}	Maximum travel time allowed per time period	20 min
γ	Relative importance of visiting buffering stations utility	TT_{max}

Table 3: Simulation Parameters

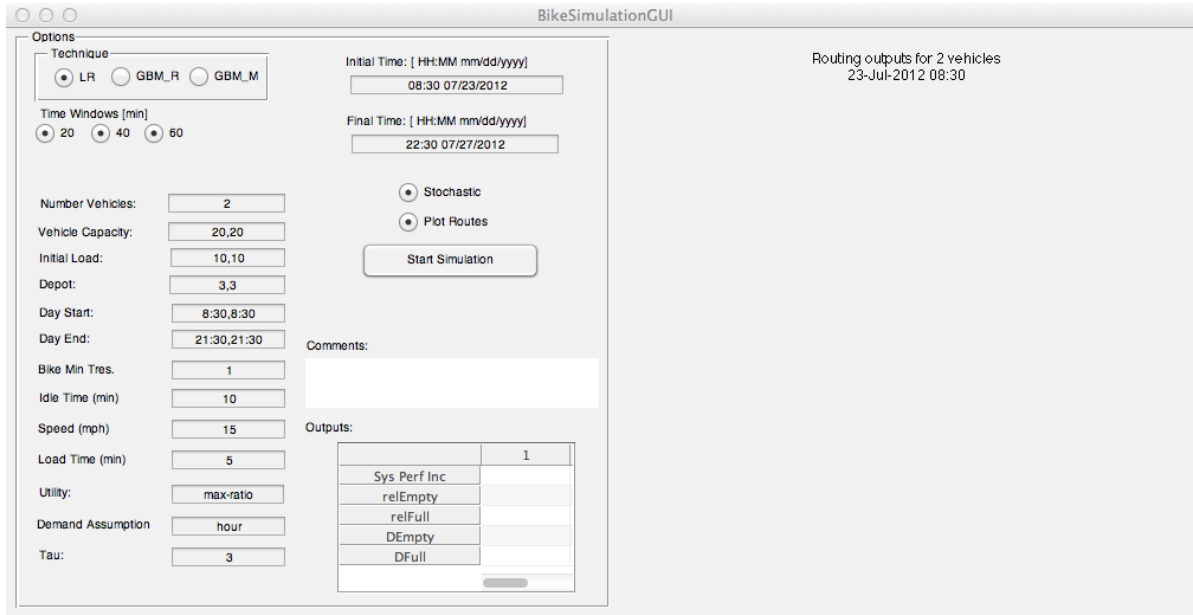


Figure 11: Software User Interface

2.5.1. Demand forecasting techniques comparison

Figure 12 shows the aggregated error measures over all the stations in the system grouped by time windows and whether or not the calibration step was implemented. For example, the first two bars on the top left corner plot represent the RMSE for the GBM model at 20 minutes with and without calibration. During the aggregation process all the errors have been normalized by the capacity of each station and the mean across all stations taken.

The differential evolution algorithmic parameters k_1 , k_2 , N_c , P_{cr} and P_m have been set based on a small test dataset and found that best results are obtained for $k_1 = 0.4$, $k_2 = 0.6$, $N_c = 16$, $P_{cr} = 0.8$ and $P_m = 0.1$. The value of $maxIter$ has been set to 6, meaning that $(N_c + 1) \times maxIter = 112$ model evaluations are performed.

For the GBM case, the optimal number of iterations M is obtained from the validation dataset. Initially, M is set to a large number (1,000), and the model performance is evaluated every 50 iterations. The M that leads to a better model is selected.

The parameter ranges and the parameter types (I: Integer, B: Binary and D: Double) are shown in Table 4. For the linear regression case, the last tree parameters are set to 0 and for the Neural Network the Min Obs field represents the number of hidden neurons. Shrinkage and Tree Depth are set to zero. The parameter settings for the non-calibration case are shown in the last row (NonCal) of Table 4.

	K	Weekdays	Mean	Filtering (τ)	Seasonality	Sig. Features	Shrinkage	Tree Depth	Min. Obs.
Min	1	0	0	0	0	0	0.001	1	5
Max	10	1	1	8	1	0.05	0.03	11	20
Type	I	B	B	I	B	B	D	I	I
NonCal	10	1	0	0	0	0	0.017	7	13

Table 4: Calibration parameters ranges and types

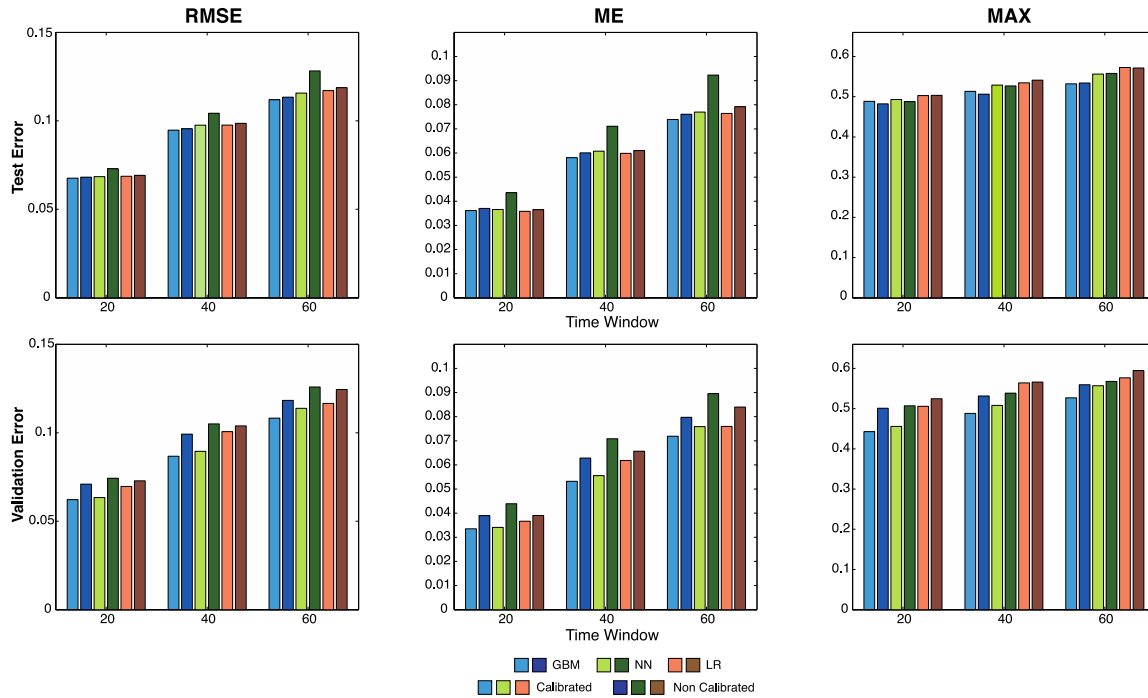


Figure 12: Error Measures for the Test and Validation datasets

As a general trend, from Figure 12, it can be observed that: 1) errors increase with the time window, 2) GBM outperforms other techniques, and 3) Neural Networks rely more on the calibration step since errors on the test set without calibration are larger than with calibration. This latter effect, however, does not occur with GBM, showing its robustness against irrelevant independent variables. Avoiding the calibration step reduces considerably the computational time and means that more recent data can be used to train the models, which can improve predictive power. For reference, training a model for a single station takes on average 13.3 and 11.28 seconds for GBM and NN, respectively on a MAC laptop with a 2.3Ghz Intel Core i5 processor. Furthermore, the RMSE error of the GBM without calibration is smaller than that for all the other techniques with and without calibration for the respective time window. The error differences among techniques increase with time window. Note that LR results are comparable to GBM for the 20-minute time window, but LR performance decreases as the time window increases. This result is due to the high importance that the current-time number of bikes has on the predictions at smaller time windows. For example, using current time observations as a prediction an RMSE of 0.074, 0.11 and 0.13 is obtained for the 20-, 40- and 60-minute time windows, respectively. As expected, the relative error with respect to the linear regression error increases with time window, going from 7% at time window 20 to 12% at time window 60 minutes.

Examining the error figures, the mean error with GBM is 3.6, 9.5 and 11.2% of the capacity of the station for the 20-, 40- and 60-minute time windows, respectively. For example, the mean error of predictions for a station with a capacity of 15 bikes, would be expected to be 0.55, 1.42 and 1.68 bikes, for the respective time windows. In terms of the RMSE, when

comparing GBM models without calibration to the equivalent NN, it is 1.33, 8.7 and 13.27 % better for the 20-, 40- and 60-minute predictions, respectively.

The relatively large maximum errors obtained—in the range of 50% of the station capacity—are, at first, somewhat surprising. However, in the above dataset relocation activities are not accounted for, which can explain sudden variations in the number of bikes that cannot be captured by the models.

Regarding the calibration step, there was no apparent correlation between the number of neighboring stations with time window, learning technique, station capacity or station activity. Similar trends are observed with the other parameters. This behavior can be explained by the stochastic nature of GBM and NN.

Considering previous results, showing that a validation step is not required when dealing with GBM models, a new set of models is trained and tested. This time the previously used validation dataset is added to the training set. Doing so impedes the use of the previous calibration process for the parameter M - the number of averaged trees. Instead, M is set to 450 and the results are compared to what would be the best M calibrated using the test set, which cannot be done in a real time application but serves to compare the trade-off between the calibration of M and the addition of more recent data. The same set of parameters used before for the non-calibrated models (Table 4) is also used here.

For each time window and M setting, 5 full iterations are run, as a cross-validation step. The RMSE values obtained for the 5 iterations and the original RMSE with the validation dataset are reported in Table 5. Note that including one week of more recent data in the training set leads to a mean improvement over the time windows of 0.8% when M is set to 450 and of 1.6% when

the best M is used. Additionally, error measures are consistent, as they remain constant across the different iterations, which validate the model.

Iteration	GBM20		GBM40		GBM60	
	Best M	M=450	Best M	M=450	Best M	M=450
1	0.0672	0.0676	0.0938	0.0947	0.1114	0.1126
2	0.0671	0.0676	0.0937	0.0947	0.1113	0.1124
3	0.0671	0.0676	0.0938	0.0948	0.1113	0.1125
4	0.0672	0.0676	0.0938	0.0945	0.1115	0.1123
5	0.0672	0.0676	0.0937	0.0946	0.1114	0.1124
Mean	0.0672	0.0676	0.0938	0.0947	0.1114	0.1124
% Improvement	1.2	0.6	1.8	0.9	1.8	0.9
Validation	0.068		0.0955		0.1134	

Table 5: RMSE values for GBM models without validation dataset

Concerning the independent feature set, one of the advantages of GBM is that variable importance can be computed, providing insight on the most significant variables that explain the number of bikes at a given station and time.

Figure 13 shows the variable importance plot for the non-calibrated GBM models at the three different time windows and aggregated across the entire system. As expected, the most important variable is the number of bikes (nbBikes) at the current time step, which for visualization purposes has been set to zero and the remaining variables normalized such that the maximum importance is 1. The neighboring station effects are shown by the variables nbBikes x that are non-zero.

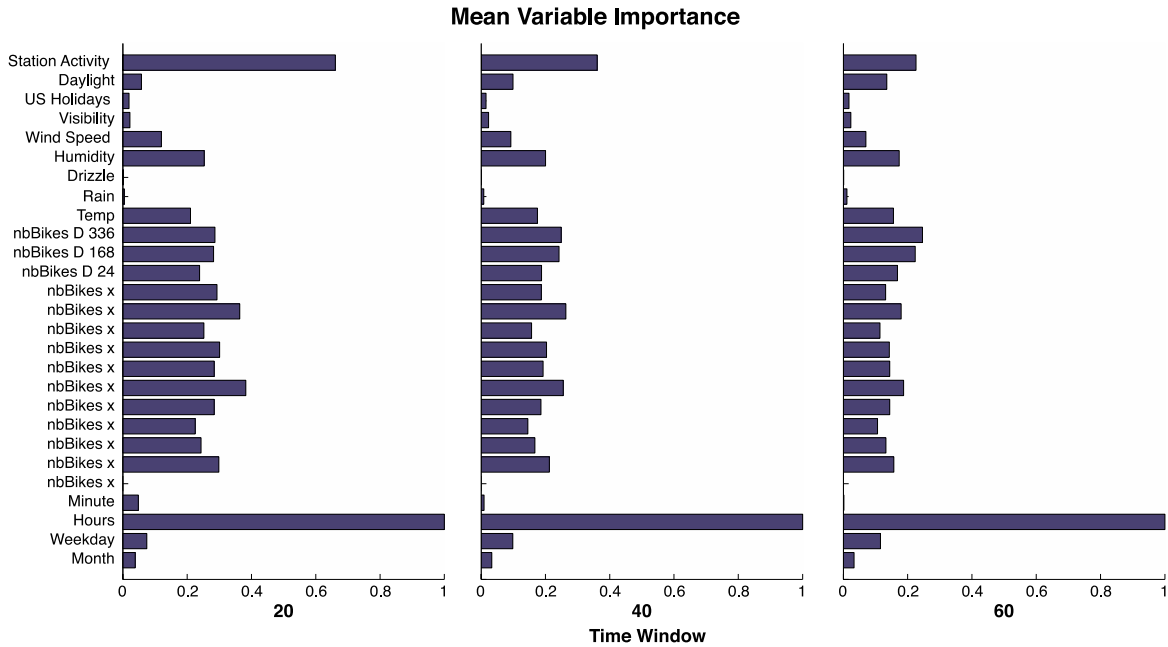


Figure 13: Mean Variable Importance for the entire system at different time windows for the non-calibrated GBM models

Interesting observations, besides that the hour of the day is the second most important variable—after the current number of bikes—are that humidity is on average more relevant than temperature, rain or drizzling and that station activity importance considerably decreases with time window. Additionally, the importance of the time-of-day variables (hour, minutes, weekday and month) and daylight appears to increase with time window. These observations indicate that the larger the time window the more general the variables are, meaning that current information about the system is less relevant when making predictions for longer time windows, and instead, general variables that capture the mean trend over time are preferred.

2.5.2. Case 1: PM peak hour

To demonstrate the potential of the modeling framework to improve current system performance, a small test case based on the Hubway dataset is proposed. The study period goes from 16:30 to 20:30 on July 24th, 2012. The PM peak has been deliberately selected as the effects

of the static rebalancing are no longer present. The settings for this case are the default settings, where 2 vehicles with capacity of 20 bikes and with an initial load of 10 bikes are considered. The simulation is run under four different predictive techniques: GBM-User, GBM, LR, and Random. This latter case tests for the impact of having inaccurate (random) demand predictions in the overall framework; for this case we sample uniform random numbers between 0 and the station capacity as the demand at that station for the next time period. The GBM user is run considering only one vehicle and the user rebalancing model.

The results of the simulation are shown in Table 6. From the table it can be observed that the system performance was already at $SysPerf_B = 90.16\%$ and after the relocation operation it was boosted $SysPerf_{INC} = 6.84\%$ and 6.02% using GBM and LR, respectively.¹ Under the random case, the system performances is degraded, which demonstrates the ability of the predictive model to anticipate events and how the overall performance depends on the ability to make accurate predictions.

Looking at the relative performance regarding empty and full events, under GBM the number of empty and full events is reduced by 57% and 76%, respectively. Note that empty events performance is significantly increased under GBM due to the ability to better predict empty events. Furthermore, the total duration of empty and full events was reduced—by 46% and 44%, respectively—as well as the average duration per empty event—average decrease from 30 minutes to 17 minutes per event—and the average duration per full event—average decrease from 39 minutes to 21 minutes per event.

¹ It should be noted that the current Hubway dataset already contains the rebalancing operations that were made by the operators of the system, meaning that what is being addressed are the remaining inefficiencies that the operator was not able to handle.

Regarding the case where the user rebalancing model is considered, it can be seen how the overall performance is slightly better than without considering the user rebalancing. In addition, due to the nature of user relocations that usually deal with small inefficiencies, the performance on empty events increases and so does the total miles traveled and number of bikes picked up and dropped off by the vehicles. In line of these results, it is left to the operators, and based on their cost structure, what is the best strategy to implement rebalancing. A compromise solution could be to limit the user rebalancing operations to only empty events, and let vehicles take care of full and empty events.

Performance Measure	Unit	GBM	LR	Random	GBM-User
<i>SysPerf_A</i>	%	96.33	95.59	88.52	96.52
<i>SysPerf_B</i>	%	90.16	90.16	90.16	90.16
<i>SysPerf_{INC}</i>	%	6.84	6.24	-1.82	7.05
<i>relEmpty</i>	%	57.41	48.28	-5.56	60.4
<i>relFull</i>	%	76.19	76.74	-45.24	71.79
<i>D_{Empty}</i>	%	45.79	40.00	27.78	52.81
<i>D_{Full}</i>	%	44.44	53.49	27.78	40.17
# Bikes picked up	#	72	75	78	62
# Bikes dropped	#	54	58	59	47
# Diverted Users	#	-	-	-	113
Total Distance Travelled	Miles	70.43	64.09	63.93	58.87

Table 6: PM Peak period simulation results

2.5.3. Case 2: One week simulation

A second, more extensive set of simulations was run for the time period from 8:30 to 21:30 for an entire week—from Monday July 23rd to Friday July 27th, 2012—using GBM as the predictive technique. In this set of simulations the impacts on the various performance measures of using different fleet size, from 1 vehicle to 5 vehicles, different vehicle capacities, ranging from 15 to 50 bikes (in intervals of 5 units) and different γ parameters (0, 0.5, 1, 2, 5, 10, 20) has been tested (separately). In all the cases all of the other parameters are set to the default values,

and the initial load of the vehicle is set to be half of the vehicle capacity.² The impact of the user rebalancing model is tested in a separate run, where the default values have been assumed.

2.5.3.1. Scenario 1: Measuring the impact of the fleet size

Figure 14 compares different performance measures for variable fleet size. As expected, the larger the fleet size, the better the performance. Using a single vehicle, system performance increases by 2.75% and with five vehicles the performance increase tops 5.86%, reaching an overall system performance of 97.88%. The largest jump is observed in going from 1 to 2 vehicles. A similar trend is observed in terms of empty and full events. A significant increase in the number of time steps that vehicles remain idle as the fleet size increases is also observed, as shown by the average miles travelled per vehicle per day; for the case of 1 or 2 vehicles, they travel on average about 100 miles per day, whereas as fleet increases, average distance travelled drops to 70 miles per day. In all cases the duration of empty and full events is significantly reduced, leveling off at around 20 minutes for 4 and 5 vehicles. Even with 5 vehicles the system cannot be projected to full performance using the proactive approach, which is not surprising due to the inaccuracies inherent to the predictions.

² Under the scenario in which vehicles can visit buffering stations, the overall results are not sensitive to the initial load parameter, as it only has an effect during the initial simulation steps.

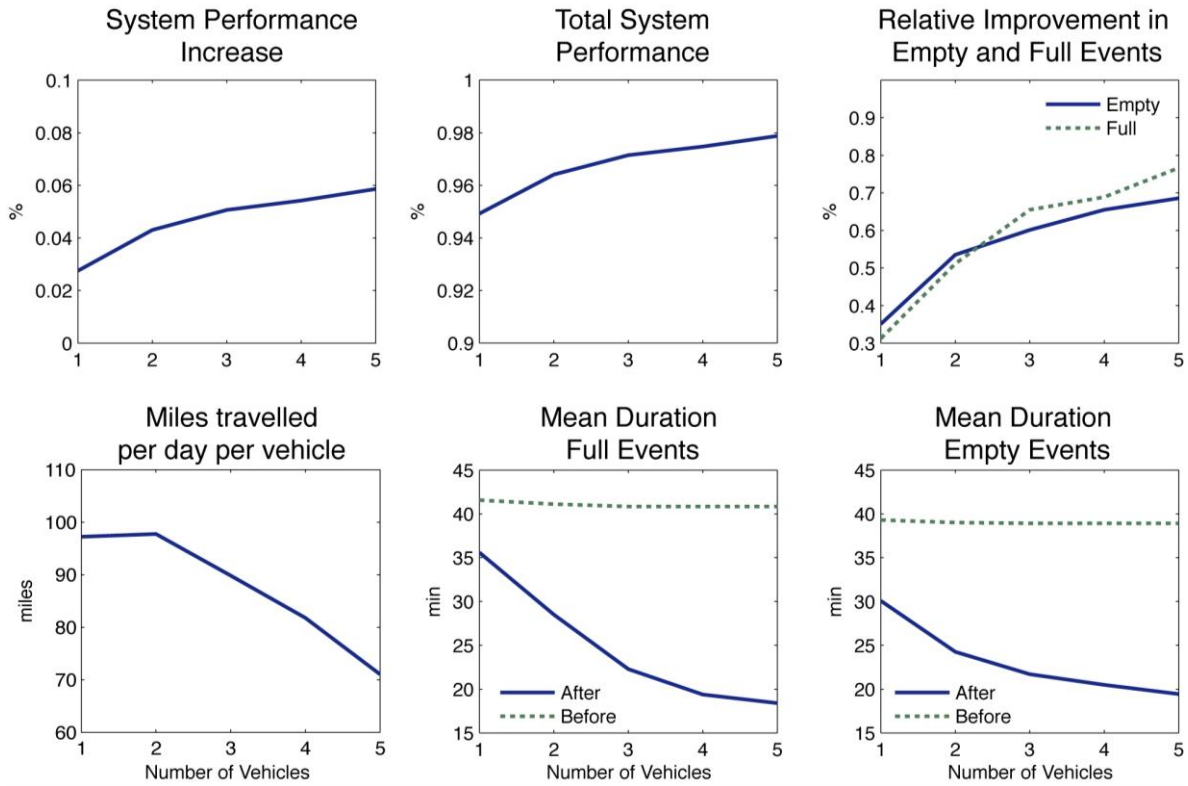


Figure 14: Performance measures trends based on fleet size

2.5.3.2. Scenario 2: Measuring the impact of the vehicle capacity

Figure 15 shows the trends in performance with gradually increasing the vehicle capacity (and initial load of the two vehicles, which is set at half of the vehicle capacity). As can be observed, all performance measures remain relatively unaffected by the vehicle capacity. This behavior is somewhat expected because of the relatively small inefficiencies that are detected.

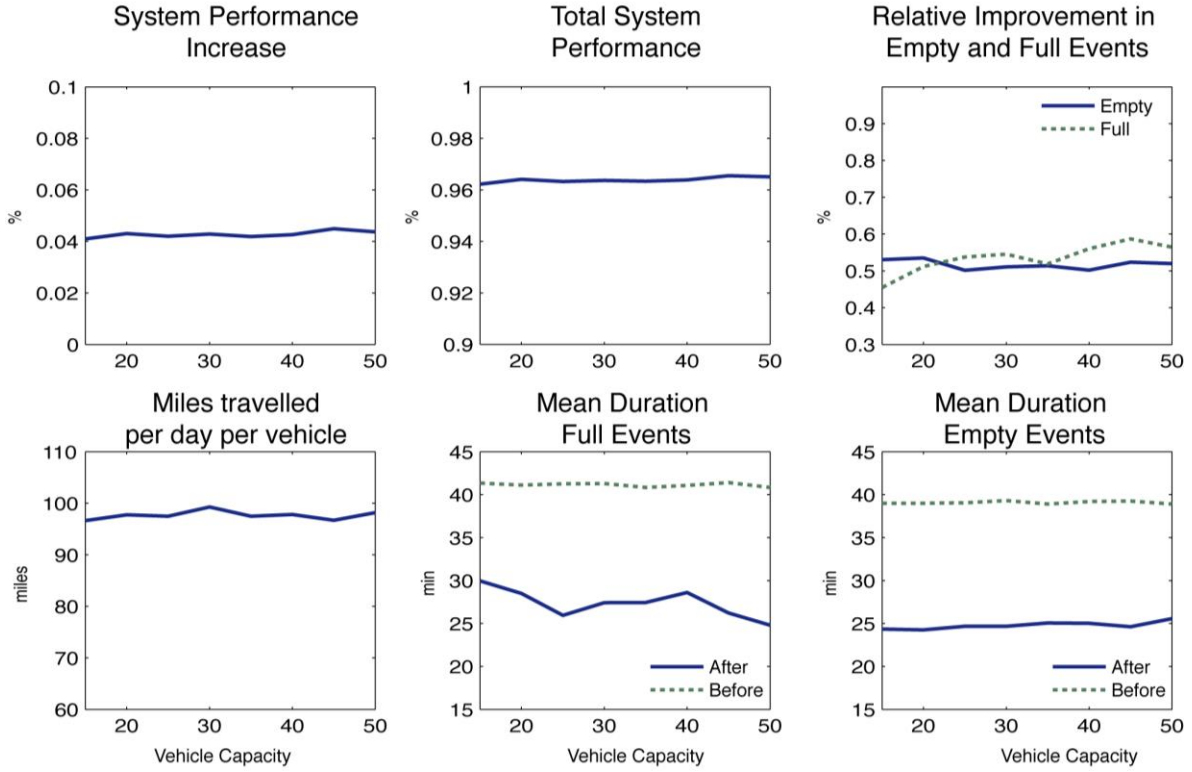


Figure 15: Performance measures trends based on vehicle capacity

2.5.3.3. Scenario 3: Measuring the impact of visiting buffering stations

Figure 16 depicts the results for a varying γ . Similarly to what occurs when changing vehicle capacity, performance measures remain relatively flat, only showing incremental increases when $\gamma > 0$. It can also be noted that when $\gamma = 0$ the miles travelled per day per vehicle are minimum, as well as the mean duration of full events. This behavior can be explained because if the penalty is not imposed on visiting buffering stations, larger inefficiencies that require visiting buffering stations have preference towards smaller inefficiencies that do not require visiting buffering stations. Larger inefficiencies tend to be related with longer and pick up events as the range $C_i - s_{i,t}^{max}$ is often larger than the range $s_{i,t}^{min} - 0$ (Figure 7). Furthermore, if a vehicle goes through a buffering station, fewer stations with inefficiencies are visited due to

the travel time constraint. Overall, because fewer stations with inefficiencies are visited, the system performance is reduced, but miles travelled and mean duration of full events decreases.

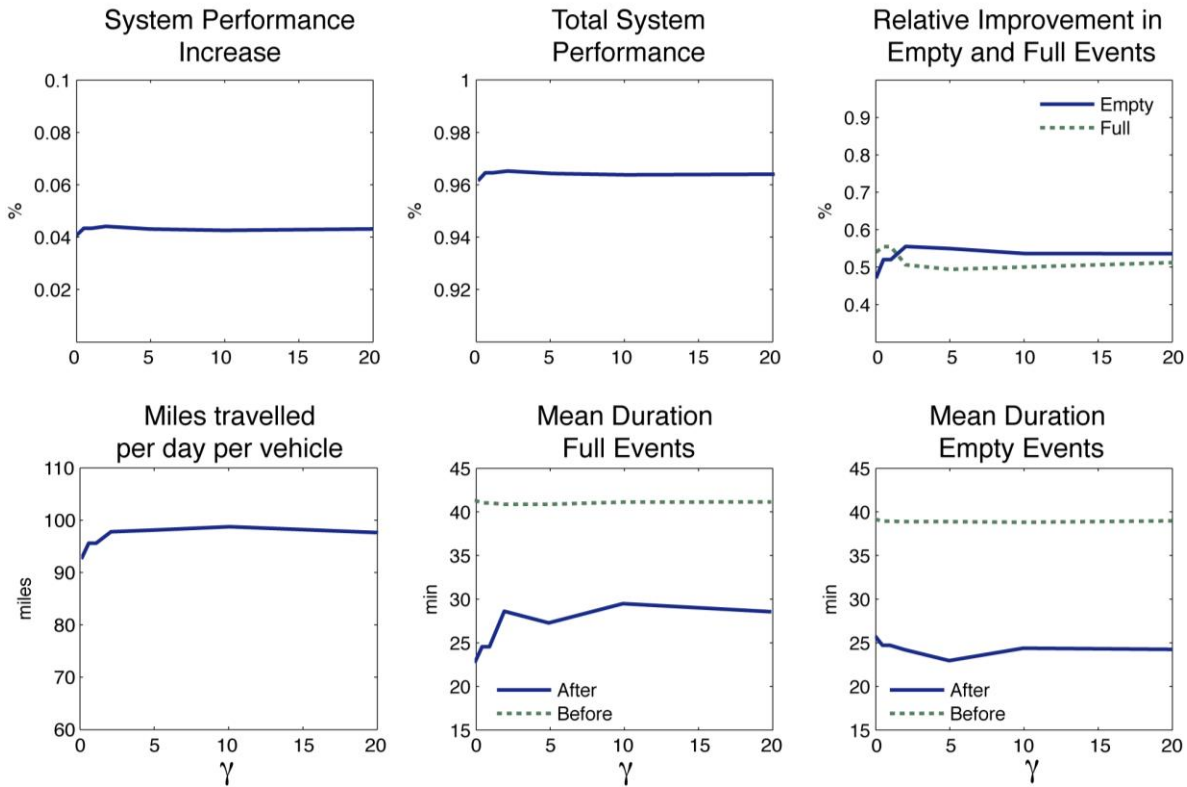


Figure 16: Performance measures trends based on γ coefficient.

2.5.3.4. Scenario 4: Measuring the impact of user relocation policies

Figure 17 shows the performance measures when varying the number of vehicles and considering the user relocation model. This scenario 4 is comparable to the first scenario, which looked at the impacts of the fleet size. It can be observed that the combination of a user rebalancing policy plus a high number of vehicles leads to unexpected results. The initial intuition would be to expect the greatest performance when both policies are combined, however, this is not the case as the performance decreases when 3, 4 and 5 vehicles are used. The

reason that explains this behavior is the propagation of the errors made when predicting the number of bikes at any given station and the sequence on which the user rebalancing and the vehicle rebalancing are run. Under the current setting, the user rebalancing is run first, and then, the “leftovers”, are fed into the vehicle rebalancing. As already pointed out, the sequence on which those models should be run is left to the operator, as it depends on the cost structure. When taking this approach, the user rebalancing model resolves as much as inefficiencies as possible, and reduces the total number of large inefficiencies. If there are less large inefficiencies, the vehicles are forced to deal with possibly more, smaller and with lower confidence inefficiencies. This would explain why the vehicles miles travelled remain constant and the overall system performance decrease.

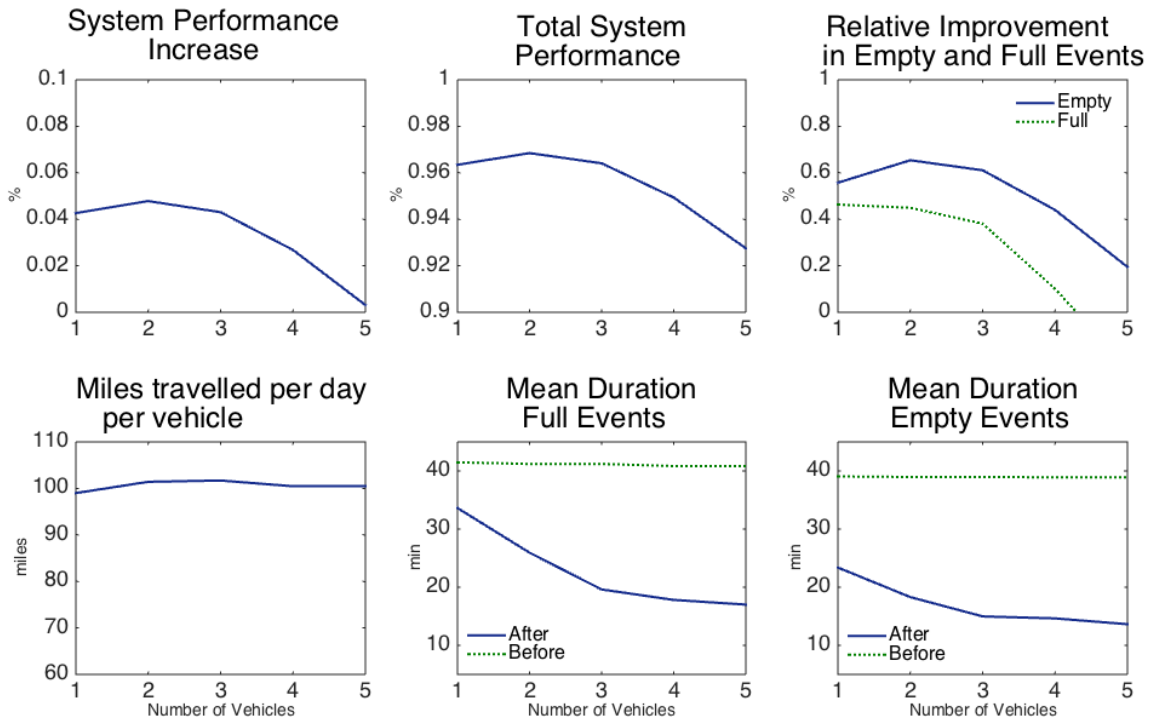


Figure 17: Performance measures trends based with user relocation policies and number of vehicles

It is also worth noting that using one vehicle and the user-rebalancing model leads to slightly better results than using two vehicles, without the user-rebalancing model. The mean number of diverted user per day, those that will require a reward to encourage a behavioral change, is 173 under the current settings, which resolve an average of 133 inefficiencies per day. The ratio of users diverted and inefficiencies resolved is 1.3, supporting the claim that users mostly deal with small inefficiencies.

A conclusion that could be drawn from the numerical examples is that, from an operational perspective, a better strategy for increasing system performance (independent of the costs) would be to use more vehicles with smaller capacities rather than fewer vehicles with greater capacity and set γ to a value of 5 as a compromise solution between system performance and travel time. This result is intuitive for this particular application of the methodology to the Hubway dataset since the former strategy affords greater flexibility in addressing what likely are relatively small “residual” system inefficiencies in an operation that has already been tuned to be efficient by the system operators.

The impacts of the user rebalancing model are incremental, even though they help to complement the vehicle routing bike relocations, especially for those where the estimated inefficiency is small. The decision on whether or not users should be used on the rebalancing operations, is an operational decision based on the cost structure of the operator. Introducing user relocation reduces leads to similar system performance measures as adding an extra vehicle.

2.6. Conclusions And Future Work

A methodological framework to solve the bikesharing rebalancing problem based on five core models has been proposed. The models are: 1) a demand forecasting model at the station level, 2) a station inventory model, 3) a redistribution needs model, 4) a user rebalancing model

and 5) a vehicle-routing model. The novelty of this approach is that it is proactive instead of reactive, as the bike redistribution occurs before inefficiencies are observed, increasing system performance and, potentially, customer satisfaction, and uses the outputs of a machine learning technique to decompose the inventory and the routing problem. The decomposition approach proposed: 1) makes the problem scalable to large bikesharing systems, 2) allows for real time implementation, making routing decisions every time a vehicle completes a limited tour, 3) is responsive to operator inputs, and 4) can accommodate user-specific models. In addition, the underlying models based on historical data are self-adaptive, as they are constantly being retrained using the most recent data available.

Simulation results based on data associated with the Hubway Bikesharing system show that significant improvements to the overall system performance could have been made (over and above that being achieved under current operation) using the proposed modeling approach—achieving improvements of nearly 7% in the afternoon peak. As expected, performance measures are better when the predictive model makes better predictions. More comprehensive tests using a full week of data demonstrate how the methodology could be used to evaluate such decisions as fleet size and vehicle capacity. To test the full potential of the framework, tests using real-time data under closed-loop control should be used (i.e., data that are not already the outcomes of operational decisions) and compared to the current rebalancing decisions made by the operator. However, access to such data was not possible.

Having outlined the fundamental philosophy behind, and demonstrated the applicability of, the framework, further research should focus on: 1) fine tuning each model, 2) testing for the optimal sequence of the user and vehicle rebalancing models, 3) building it as a web application tool for bikesharing operators to use, and 4) testing its transferability to other bikesharing

systems. In terms of improving the various models, the forecasting model should incorporate more independent variables such as large events (e.g., basketball games, concerts, etc.), transit arrivals and departures in areas nearby bikesharing stations or real time distribution of people in the city, which could be gathered from cellphone towers or social media data. For the routing model, different objective functions can be proposed that incorporate the concept of maximizing a cumulative reward instead of the immediate benefit gained by routing a vehicle to a station. Additionally, some of the input parameters, such as τ or the *idleLogic* could be made self-adaptive to better handle idling situations.

3. CAR2WORK: A NEW MOBILITY CONCEPT FOR COMMUTERS

3.1. Introduction and motivation

Over the last decade there has been a surge of shared-use mobility concepts that are redefining how people move in urban areas. In this context, Car2work is new shared-use mobility concept that fills a gap in the existing approaches that are not able to integrate with the existing transit network. Car2work differs from the traditional dynamic-ridesharing approaches in the sense that 1) it is designed for recurring trips, focusing on commuter trips, 2) the concept of drivers is dropped; instead vehicles are used that carry at least one commuter when travelling, 3) commuters announce their trips in advanced and, 4) multiple trips per commuter are allowed along the day. The main goal is to connect commuters with workplaces while guaranteeing a trip home and offering some degree of flexibility.

Carsharing can take different forms and providing a general definition of the concept that embraces all possible cases may be challenging. Barth et al. (2002) provide a framework to properly classify car-sharing programs between Carsharing, Station Cars and other combined approaches. Generally speaking, car-sharing schemes are membership programs where the members share a fleet of vehicles and are typically charged in an hourly basis. As a result, car-sharing is aimed to offer similar flexibility as a private automobile but without having to cope with car ownership costs (Millard-Ball et al. 2005). In addition, car-sharing efficiently serves transportation demands by reducing the number of automobiles, cuts transportation costs, since are shared among a pool of users, reduces environmental impacts when low-polluting vehicles are used and when are linked to major transit stations, can increase transit ridership (Barth & Shaheen 2002). Some recent studies about environmental impacts of carsharing can be found on

Firnkorn & Mueller (2011) and Musso et al. (2012) dealing with CO2 emissions reductions per car-sharing user in a free floating carsharing program and air quality improvements in a multistation car-sharing program, respectively.

Carsharing history begins in the late 40s in Zurich, Switzerland, and its implementation was purely economically driven to provide car accessibility to people that could not afford an automobile. Later on, in mid 80s, a second wave of successful carsharing programs started in different European countries, being the most popular ones in Switzerland and Germany (Shaheen et al. 1998). In the US, two demonstration projects were launched during mid 80s, Mobility Enterprise at Purdue University and Short-Term Auto Rental (STAR) in San Francisco, but both ceased their activities after the pilot project concluded (Shaheen et al. 1998).

The third wave of carsharing programs started on mid 90s, when in North America, between 1994 and 2004, 50 new car-sharing programs were implemented. As of July 2014, there were 23 active programs in US alone (Shaheen & Chan 2015). The growth in terms of number of users and number of vehicles it is shown in Figure 18.

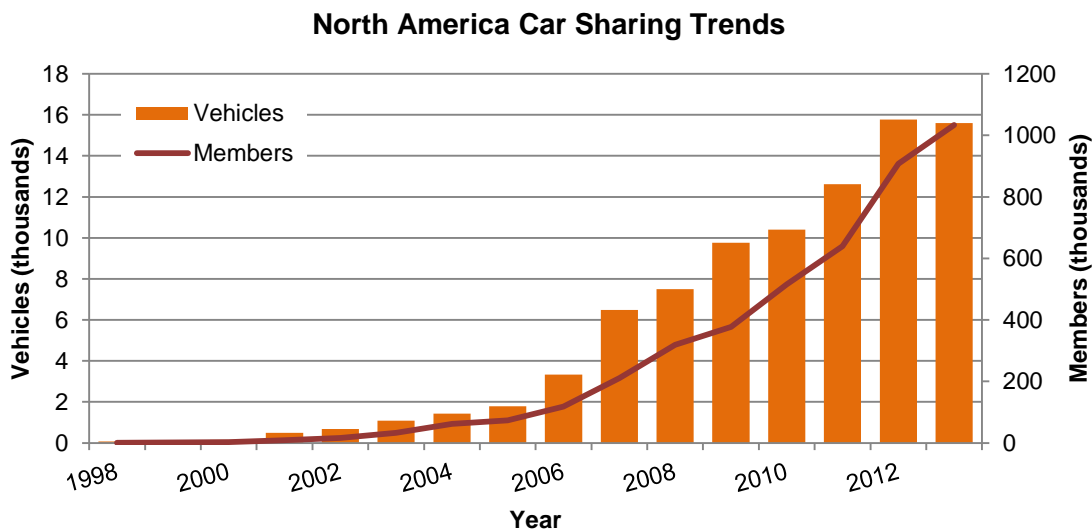


Figure 18: North America Car-Sharing trends. Adapted from (Shaheen 2013)

It is worth noting that car-sharing development is linked to technological advances, where the main barriers for its implementation have been overcome by the use of technology. For example, the switch from universal door keys located at lockers to the smart cards, or from phone reservations to automated reservation and access systems, which make car-sharing programs more attractive to new users. In addition, real time data acquisition is now feasible, which has improved vehicle fleet management and therefore the efficiency and cost-effectiveness of car-sharing programs has increased (Millard-Ball et al. 2005).

In line with technological advances, new and innovative approaches of car-sharing programs are emerging to meet customers' expectations. The current trends are the introduction of instant access to the vehicles, open-ended reservations, one-way trips, floating cars, pre-paid usage cards or alternate fuel vehicles. Those new car-sharing programs market themselves as the best alternative to car ownership since the users can have access to a vehicle any time, without reservation and without the need to specify the returning time or final destination (Shaheen & Cohen 2007). The downside of this approach is the increasing modeling efforts and the fleet management complexity to efficiently handle the potential demand for the program, providing a research gap that can be exploited.

To better understand those new carsharing programs, different research oriented pilot projects have been launched. Among them, it is worth pointing out the followings: Praxitele (1997), Intellishare (1999), ICVS (2002), ZEV·NET (2002) and Mobility on Demand (2012). These projects have been selected for their significance, different types of operation, literature available, and novelty at the time they were implemented. For each of the programs a basic description and the related research that has been done are given below.

Praxitele is known to be the first station car system deployed in large scale using fully electric vehicles. It was implemented in Saint-Quentin-en-Yvelines in 1997 by a French consortium created in 1993. It had 50 electric vehicles, 5 stations and a pool of 500 users. The project lasted for one year and the conclusions were that the users express great satisfaction and its willingness to expand the project but it was expensive to run and the vehicles were underutilized. Concerning management, the project used smartcards and had incorporated demand-forecasting algorithms to anticipate vehicle relocation operations developed by the Institute National de Recherche en Informatique et en Automatique (INRA) (Massot et al. 1999). The system was modeled using queuing and discrete events theory (Allal et al. 1996). During the project several research initiatives were launched, such as a simulation model to simulate Praxitele vehicles in the urban environment before its physical deployment (Arnaldi et al. 1996), the management of the share-used vehicles fleet by using the concept of favorable and unfavorable states in a static fashion (Chauvet et al. 1997), the vehicle allocation problem formulated as a pickup and delivery problem with non simple paths and deterministic demand (Dror et al. 1998) or the development of a balancing method based on defining a fix tour that a towing truck follows to redistribute the vehicles among stations and where a decision is made at each station on the number of vehicles to relocate (Chauvet et al. 1999).

The UCR IntelliShare program was launched at the University of California, Riverside (UCR) in cooperation with Honda Motor Company on April 1999 with an initial fleet of 15 electric vehicles and three stations located on and near the campus. The research objectives were to understand the operational aspects of an electric vehicle carsharing, increase the accuracy of current simulation models and provide the campus with a new and clean transportation system (Barth et al. 2000). As of 2003 the program expanded to 35 electric vehicles and 5 stations. The

program uses smartcards, the vehicles are equipped with advanced monitoring technology and can be returned to any of the stations and reservations are not required. When making a reservation or checking out a vehicle, the user is asked to provide basic information about the trip, such as the destination, trip type, estimated travel time or number of passengers. In addition, users can communicate the operator any malfunctioning of the vehicles through an onboard touchscreen (Barth & Todd 2004). The program has originated a rich dataset that is used to test user behavior (Barth & Todd 2001), improve vehicle relocation algorithms by implementing route prediction and time of arrival estimation techniques (Karbassi & Barth 2003) or user-based vehicle relocation techniques (Barth & Todd 2004). An addition, it is considered the test bed for the posterior development of ICVS technology that was later on implemented in Singapore.

The Intelligent Community Vehicle System (ICVS) was launched by Honda on March 2002 in Singapore in cooperation with Singapore's Economic Development Board (EDB) with the aim to proof itself as economically feasible. The program was marketed as Honda Diracc. The ICVS concept by Honda was tested in previous projects, such as the Carlink pilot project at University of California, Davis or IntelliShare project described before (Honda 2012). The program ceased operations on March 2008 but was taken over by Kah Motor Co, and is now running under KahShare commercial name. ICVS Singapore used contactless smartcards to access vehicles that could be picked up and returned to any of the 12 stations without the need of reservation. A total of 50 hybrid vehicles were used. The user, however, had to specify the destination station in advanced, but could be changed during the trip. In addition, user demands were anticipated to reduce relocation costs and passenger waiting times. The dataset generated from ICVS has been used in several studies to test new algorithms. Cheu et al. (2006) tested two trip-forecasting approaches: Neural Networks and Support Vector Machines. Kek et al. (2006)

designed a relocation simulation model using ICVS data and Kek et al. (2009) improved the previous model by incorporating a three-phase optimization-trend-simulation (OTS). Nair & Miller-Hooks (2011) suggested a different approach to the vehicle relocation problem by introducing stochastic demand in the formulation and solving a chance constrained problem.

Mobility on demand is a research project launched by Massachusetts Institute of Technology (MIT) to rethink urban mobility and provide a better service than competing transportation modes such as private cars, taxis or transit. It was launched under the Smart Cities Program (Mitchell 2012). The underlying concept is a shared-use vehicle program with different types of vehicles, CityCars, RoboScooters, Bicycles or Segways, which are used according to city characteristics. The key success factors for mobility on demand are the use of technology to get real-time demand monitoring, real-time management for vehicle relocation to balance supply and demand and the use of dynamic pricing for demand management. The real time information collected is then fed into a queuing theory model that outputs the system performance. As vehicle relocation strategy strong pricing policy is suggested to encourage self-relocation, and, if needed, operator relocations can be performed using driverless vehicles. The possession of demand forecasting models is seen as a competitive advantage for the operators and is a strong barrier to new entrants in the market. However, up to date, no real implementation exists but case studies in Florence, Lisbon, San Francisco and Shanghai are being carried out (Mitchell 2012).

ZEV·NET is a corporate station car program launched by University of California, Irvine (UCI) in 2002 that uses clean vehicles. ZEV·NET provides accessibility from rail stations to employment areas in a flexible and sustainable manner. Currently, the main station is located at Irvine Transportation Center, there are five corporations involved, including UCI, and has a fleet of 16 Toyota RAV4 EVs and 2 Toyota Prius (Heling et al. 2009). The scheme works as follows.

A commuter has a vehicle available at home that uses to commute to the train station, where the vehicle is parked, and then takes the train to the final destination. A reverse commuter that needs access to the employment area picks up the vehicle parked in the station for this last commuting leg. During the day the corporation employees use the vehicle as a regular carsharing program. Finally, the second commuter will use the same vehicle to go to the train station, and the initial commuter will have a vehicle available to go home. Each corporation enrolling the program can subscribe for 2 or 4 vehicles and need to assign a pool of 8 to 10 drivers among the employees (City of Irvine 2014).

The purpose of ZEV·NET is to provide an alternative mode of transportation with the benefits of the private automobile and the costs of transit that can increase commuter rail patronage and reduce air pollution and carbon emissions. In addition it reduces parking needs at train stations, which is a main issue according to the 2008 Metrolink on board customer satisfaction survey (McCourt 2008).

Universtiy of California, Irvine researchers have been monitoring the performance of the scheme throughout the years. The data generated by the program has been used to understand user characteristics and responses towards shared-used vehicles (Heling et al. 2009), to characterize charging behavior and vehicle energy consumption (Heling & Brown 2010), the inference of activities undertaken by users by means of a hybrid dynamic mixed network (Gogate et al. 2006) and the development of the personal travel assistant (PTA) to understand the dynamics of human travel behavior (Recker et al. 2010).

Besides the above research driven projects, a range of private programs have been launched around the World (Car2go, DriveNow, ZipCar) with different operational characteristics. More recently, during the last decade there has been a surge of shared-use

mobility concepts. As defined in Shaheen & Chan (2015), shared-use mobility is “an innovative transportation solution that enables users to have short-term access to transportation modes on an as-needed basis.”. Under this broad definition of shared-use mobility, any means of transportation that serves the above purpose is considered, including carsharing, peer-to-peer carsharing, bikesharing, ridesharing, ridesourcing, among others.

Several providers are taking different approaches to shared-use mobility alternatives. For example, ride-sourcing companies such as Uber and Lyft started offering on-demand mobility, but quickly launched Uperpool and Lyft Line to let their customers share rides. According to Lyft CEO, within 6 months Lyft Line became the most popular service the company is offering in San Francisco (Terdiman 2015). Others, such as Leap, Chariot, Bridj and Via offer an alternative to transit by providing on-demand, flexible private bus lines. ZipCar provides short-term rentals, Carma and Zimride allow users log their trips so users can find matches, and Scoop automatically creates carpools on a per-trip basis. There are also more traditional vanpool and carpool services that co-workers arrange by themselves.

Numerous benefits have been reported of this new systems, including: car-ownership and vehicle usage reduction, increase network connectivity and encouraging multi-modality (Shaheen & Chan 2015). However, its long-term effects are yet to be understood. In addition, most, if not all the approaches presented above do not fully integrate with the existing transit network. Instead, they compete with it and it is left to the users to decide whether or not it is more efficient to transfer to transit along their trips. Furthermore, most of the companies focus their efforts on urban areas with rather large population densities, leaving out residential areas where it may be more costly or difficult to provide the service. As pointed out in Agatz et al.

(2012), effectively integrating a ride-sharing system with transit systems has the potential to increase the transit system coverage area, leading to societal and environmental benefits.

A motivation of this work comes from real-world observations (in the Orange County, California) that in the types of regional development patterns that tend to dominate the post automobile era, one of the main barriers for the use of public transit (and especially rail) is the connectivity between workplaces, homes and rail transit stations—effective line-haul rail service between outlying residential areas and concentrated employment centers, being largely negated by the need for personalized mobility between the transit stations, homes, and places of employment. An attempt to address this issue was Zev·Net (City of Irvine 2014), as described earlier, in cooperation with Toyota Motor Sales. However, its implementation was limited to a single station and it was never modeled from an operational perspective. To graphically demonstrate the lack of connectivity, directions provided by Google Maps from University of California, Irvine (UCI) to the Irvine Transportation Center by different modes are shown in Figure 19. As it can be observed, it takes about 1hr and 7 min using transit and 15 min using a private vehicle, which is a 52 minutes time saving. Note that the total length of the trip is 8 miles. Another, longer, trip, from UCI to Los Angeles Union station, is shown in Figure 20. In this particular case, there is a time saving of 50 minutes, even under congested conditions. It is clear that under this scenario transit is not competitive, at least in terms of travel time, and that most of the time involved on the transit trip is to get access to the transit station.

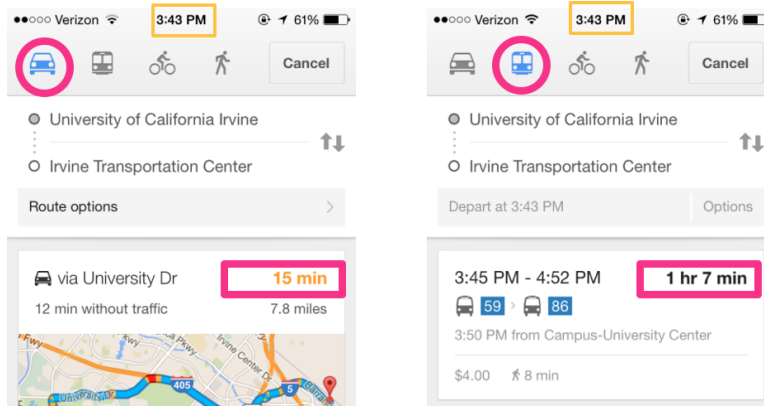


Figure 19: Directions from University of California, Irvine to Irvine Transportation Center

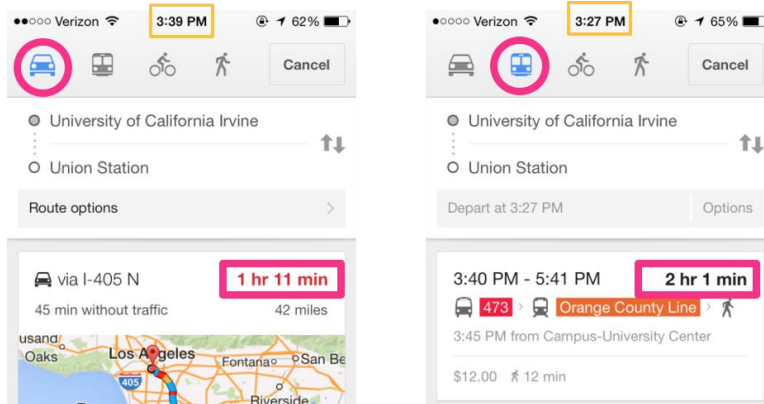


Figure 20: Directions from University of California, Irvine to Los Angeles Union Station

In this work, the focus is on the modeling aspect of Car2work and its design considerations without accommodating two possible extensions: handling dynamic requests in real time when a vehicle is en-route, and a short-term car rental service while vehicles are idling at transit or workplaces. Furthermore, it is not within the scope to investigate on the business model that will make it economically viable.

The system is modeled under a simulation framework that at its core relies on a variation of the peer-to-peer ride-matching problem presented in Masoud & Jayakrishnan (2014) that is extended to accommodate the existing specifications. Commuters announce their trips and the

model finds the optimal trip plan, including transit connections and guaranteeing a match for the returning trip home. Although not considered here, the methodology is general enough to address a variety of scenarios, including the use of autonomous vehicles, different fleet splits, multiple transit modes, and/or varying commuter preferences, dynamic requests, and the short-term car rental service while vehicles are idling. An exact solution method based on an aggregation/disaggregation algorithm is proposed.

3.2. Related work

The peer-to-peer (P2P) ride-matching problem is not new to the literature. Agatz et al. (2012) provide an extensive review of the dynamic ride-sharing problem detailing its characteristics, variants and different strategies that have been proposed to solve the problem. Among the variants of the problem presented in Agatz et al. (2012), the current problem falls under the category of multiple-riders, multiple-drivers with the added multi-modality and multi-hop component.

Similar problems in the literature that resemble the Car2work concept are the pick-up and delivery problem with transfers (PDPT) (Shang & Cuff 1996; Cortes et al. 2010; Masson et al. 2013) and the dial a ride problem with transfers (DARPT) (Masson et al. 2014). Shang and Cuff (1996) propose a scheduling heuristic to solve the problem on a real case instance with 167 deliveries. Optimal number of vehicle is also optimized and transfers can occur at any location, for any item, and any vehicle. Cortes et al. (2010) provide an extensive review of the existing literature of the pick-up and delivery problem (PDP), its extensions, various solution approaches, and propose a new extension to handle transfers. The solution method is exact, using a branch-and-cut method based on Benders decomposition (Benders 1962), and implementing combinatorial Benders cuts (Codato & Fischetti 2006). The largest instance proposed has 6

requests, 2 vehicles and 1 transfer point. Masson et al. (2013) use an adaptive large neighborhood search (ALNS) to solve the PDPT. All delivery points can be transfer points and the larger instances have 106 requests, 24 delivery locations and 24 transfer points and 193 requests, 5 delivery locations and 5 transfers. More recently the same authors have extended the previous heuristic to solve the DARPT (Masson et al. 2014). In all the cases, it is shown that by including transfer points the transportation costs are reduced. However, the added user inconvenience of transfers is not accounted for.

Transit concepts, similar to the one presented, aiming to increase transit flexibility also exists, notably flexible route transit systems (Li & Quadrifoglio 2010; Quadrifoglio et al. 2008; Qiu et al. 2014), flexible taxi pooling dispatching systems (Lee et al. 2005) or variations of the dial-a-ride (DAR) problem, as the High-Coverage Point-to-Point Transit (HCPPT) system (Cortés & Jayakrishnan 2002). Herbawi and Weber (2012) also propose a multi-hop ride matching problem with time windows where drivers can cooperate to bring riders' to their destinations, that is solved using a tailored genetic algorithm. Most of the examples above, except the Mobility Allowance Shuttle Transit (MAST) by Quadrifoglio et al. (2008), which is solved by introducing logic cuts to a MIP formulation based on user behavioral assumptions, rely on building custom heuristics to find near optimal solutions. In most cases, such custom heuristics are employed to respond to real-time rider's requests.

Car2work is very similar to P2P ridesharing systems, in the sense that it is inherently very spatiotemporally sparse. This makes using the most efficient matching algorithms that make the best use of very limited available resources important. Although using heuristic algorithms in DAR problems (which may seem very close to P2P ridesharing systems in terms of formulation) is very common, and can yield good quality solutions, this is not the case with the proposed

system. One reason is that DAR problems have multiple drivers that work for the system, and may perform pick-up and drop-offs at any location, and at any point in time. In addition, the requests are not all concentrated in peak-hours (as opposed to commuter trips), and therefore there is a higher possibility of multiple drivers being idle when a request arrives. In such a setting, a heuristic algorithm can provide a good solution on assigning requests to drivers, mostly based on spatial proximity (and other potential measures). In the proposed system, however, a multi-trip approach is implemented, where commuters can announce more than one trip, namely the home to work and the work to home trips, and transfers (multi-hop) and multiple modes are allowed. These conditions make the use of clustering heuristics impractical, as not only the spatiotemporal constraints of a single trip need to be met, but also trip connectivity constraints that in most cases span over the entire day. Another reason why the use of heuristics is not recommended in this case is because the demand for the proposed system is not dynamic: trips are announced well in advanced (at least one day), and the matching does not need to happen in real time (unlike a typical DARP system that should be able to handle dynamic requests). Since finding a matching is not time-sensitive, use of heuristic algorithms will only lead to sub-optimal solutions, for no additional benefits. As a result, the problem has been formulated as a binary problem and an exact solution method based on an aggregation/disaggregation algorithm that renders optimal solutions has been proposed.

3.3. System Definition

Car2work is a mobility alternative designed for commuters with relatively regular commuting schedules. It differs from the traditional dynamic-ridesharing problem described in (Agatz et al. 2012), where the matching occurs in short-notice, drivers are independent private

entities, the system is designed for occasional or non-recurring trips, and the trips are pre-arranged. The main differences between dynamic ridesharing and car2work are:

- 1) Car2work's core it is based on recurring (commuting) trips. However, it can be extended to non-recurring, occasional trips, if such trips are added on top of already existing routes, as in a dynamic-ridesharing problem. As a result, the initial concept is not dynamic and the problem can be solved "overnight".
- 2) The concept of *drivers* is dropped; and instead *vehicles* that carry at least one commuter when travelling are used. Under this definition, the driver can be any commuter in the vehicle, and a vehicle cannot travel alone. Transit vehicles are treated differently, as they do not require having a driver to travel.
- 3) Commuters announce their trips in advanced and an automated all-or-nothing matching strategy is performed. All trips announced by the user need to be completed, otherwise the traveler is not matched and does not participate in the system. A simple trip announcement consists of an origin, destination, earliest departure time, latest arrival time and maximum deviation from the shortest travel time (or alternatively, a maximum travel time budget). Commuters can announce one or multiple trips.
- 4) Because of the possibility of multiple trips, the routing decision variables are indexed over trips, not commuters.
- 5) The cost of the ride can be shared among the users or among the participating commuter's companies.

Note that most of the points listed above can be relaxed to accommodate other types of operations. For example, commuter preferences such as willingness or ability to drive,

autonomous vehicles that travel without a driver, or that not all trips announced by a user need to be accommodated (or limit the guaranteed trip to only the home-to-work and work-to-home trips) can also be accounted for slightly modifying the constraint set.

As an example, Figure 21 below shows a Car2work system with 5 users, 3 workplaces and 2 transit stations. The color code indicates the commuter to workplace relationship, and the solid and dashed arrows represent the morning and evening commutes respectively. A lunch and a personal business trip are also depicted, corresponding to nodes 11 and 12 in the figure. For simplicity, in this representation it is assumed that all the commuters have the same working schedule and preferences, therefore temporal constraints are not considered.

In this example, commuters 1 and 7 have pre-assigned vehicles, and another vehicle is parked at transit station 5. The optimal solution to this problem with the objective of maximizing the number of served commuters suggests that commuter 1 should leave home to pick up commuter 2, then 3, and drive to the transit station 4. Note that commuter 1 and 2 work at the same workplace (9) while commuter 3 works at workplace 8. At transit station 4, they all take the train toward the transit station 5. Here, commuter 1 and 2 take the vehicle parked to drive to their final destination, workplace 9. Commuter 3 drives to workplace 8 using a vehicle that commuters 6 and 7 have parked at station 5 on their way to workplace 10. Commuters 6 and 7 use the vehicle left by commuters 1, 2 and 3 at transit station 4 to get to their workplace 10. While commuters are at their respective workplaces, two employees from workplace 8 decide to use the vehicle for a lunch trip. Another employee from workplace 8 has a business meeting a location 12 and uses the vehicle left by commuter 3. For the return home trip, the commuters undo what they previously did to get to work.

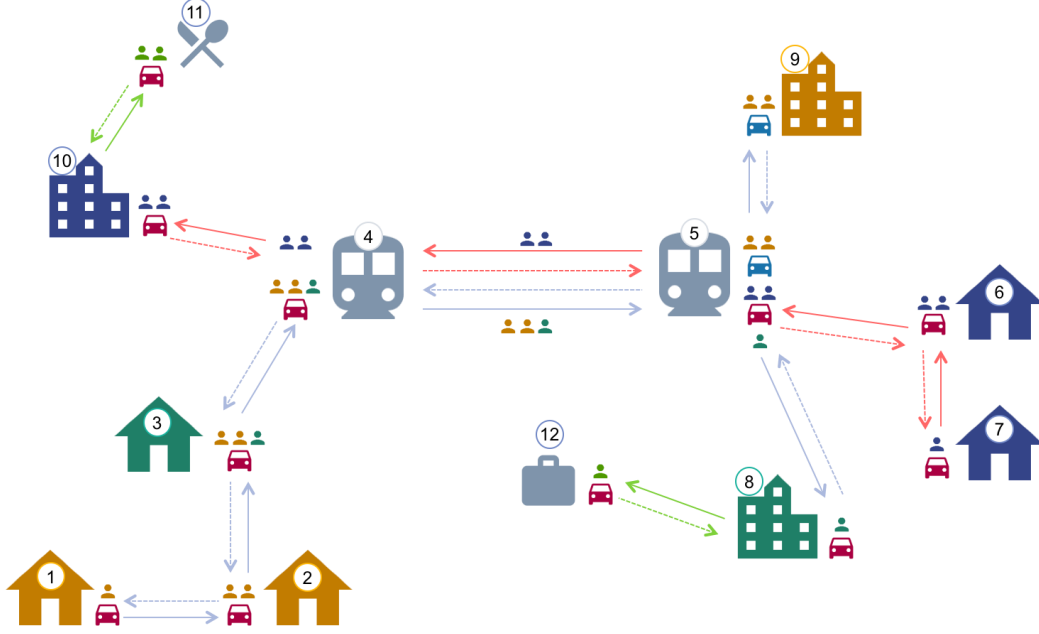


Figure 21: Car2work example.

3.4. Model Formulation

The formulation presented here is inspired by the peer-to-peer ride-matching problem defined in Masoud & Jayakrishnan (2014). The problem is formulated using a time-expanded network, and as a purely transshipment problem. A supply (S_o) and a demand (S_D) node is added to the set of stations (S) in the network. Stations or nodes are homes, workplaces, transit station locations or any other location announced by the commuters. Time (T) is discretized into T_n intervals of length dt between the earliest departure and latest arrival times observed on the set of trips TS . The trip set is split into transit trips TS_t , and commuter trips TS_r . O_k and D_k represent the origin and destination stations, respectively, of trip $k \in TS$. Similarly, the set of vehicles is V , which includes the vehicles available to the commuters (V_r), the transit vehicles (V_t) and a dummy vehicle (V_{dummy}). A dummy vehicle is introduced to ensure that commuters

can linger during some time periods in a given station as by definition of the decision variables, a commuter must be in a vehicle at all times. This may occur on transferring situations or at the beginning or end of a trip. Each vehicle $v \in V$ has a capacity C_v . Commuters are represented by the set R . Finally, the set of links L is defined as the 4-tuple (s_i, t_i, s_j, t_j) , representing a link from station s_i to station s_j , departing at time interval t_i and arriving at time interval t_j . Note that the travel time between stations s_i and s_j can be defined as $tt_{i,j} = (t_j - t_i) \cdot dt$. The following subsets of L are also defined. L_t is the subset of transit links where s_i and s_j are transit stations. L_k is the subset of feasible links for a trip $k \in TS$. L_v is the subset of feasible links for vehicle $v \in V$. The set $L_{kv} = (L_k \cap L_v)$ is the intersection of the feasible links for trip k and the vehicle links. The set of feasible links for the dummy vehicle is represented by L'_v .

The decision variables for the problem are defined as follows:

$$X_l^{kv} = \begin{cases} 1 & \text{if trip } k \text{ includes traveling on link } l \text{ with vehicle } v \\ 0 & \text{otherwise} \end{cases} \quad (\text{Eq. 20})$$

$$X_l^v = \begin{cases} 1 & \text{if vehicle } v \text{ travels on link } l \\ 0 & \text{otherwise} \end{cases} \quad (\text{Eq. 21})$$

$$Y_r = \begin{cases} 1 & \text{if commuter } r \text{ is matched} \\ 0 & \text{othreewise} \end{cases} \quad (\text{Eq. 22})$$

The complete formulation of the mathematical problem is shown below on problem P 4.

P 4:

$$\max Z = \beta_1 \sum_{r \in R} Y_r - \beta_2 \sum_{v \in V_r} \sum_{\substack{l \in L_v: \\ i \neq j}} c_{i,j} X_l^v - \beta_3 \sum_{r \in R} \sum_{k \in TS_r} \sum_{v \in V} \sum_{l \in L_{kv}} c_{i,j} X_l^{kv} + \beta_4 \sum_{v \in V_r} \sum_{\substack{l \in L_v: \\ i=S_0, j=S_D}} X_l^v \quad (4.0)$$

s.t.

$$\sum_{\substack{t_i, s_i: \\ l=(s_i, t_i, s_i, t) \in L_v}} X_l^v = \sum_{\substack{t_j, s_j: \\ l=(s, t, s_j, t_j) \in L_v}} X_l^v \quad \forall v \in V_r, \forall t \in T, \forall s \in S - \{S_0\} \quad (4.1)$$

$$\sum_{\substack{l=(s_i, t_i, s_j, t_j) \in L: \\ s_i=S_D, s_j=S-\{S_0\}}} X_l^v = 0 \quad \forall v \in V_r \quad (4.2)$$

$$\sum_{l=(S_D, t_i, S_0, t_j) \in L} X_l^v = 1 \quad \forall v \in V_r \quad (4.3)$$

$$X_{l'}^v = 1 \quad \forall l' \in L'_v, \forall v \in V_{dummy} \quad (4.4)$$

$$X_l^v = 1 \quad \forall l \in L_t, \forall v \in V_t \quad (4.5)$$

$$\sum_{v \in V} \sum_{\substack{l \in L_{kv}: \\ s_i=O_k}} X_l^{kv} - \sum_{v \in V} \sum_{\substack{l \in L_{kv}: \\ s_j=O_k}} X_l^{kv} = Y_r \quad \forall r \in R, \forall k \in TS_r \quad (4.6)$$

$$\sum_{v \in V} \sum_{\substack{l \in L_{kv}: \\ s_j=D_k}} X_l^{kv} = Y_r \quad \forall r \in R, \forall k \in TS_r \quad (4.7)$$

$$\sum_{v \in V'} \sum_{\substack{t_i, s_i: \\ l=(t_i, s_i, t, s) \in L \\ s_i \neq O_k, s_j \neq D_k}} X_l^{kv} = \sum_{v \in V'} \sum_{\substack{t_j, s_j: \\ l=(t, s, t_j, s_j) \in L \\ s_i \neq O_k, s_j \neq D_k}} X_l^{kv} \quad \forall k \in TS_r, \forall t \in T, \forall s \in S - \{S_0, S_D\} \quad (4.8)$$

$$\sum_{\substack{k \in TS: \\ l \in L_{kv}}} X_l^{kv} \geq X_l^v \quad \forall v \in V_r, \forall l = (s_i, t_i, s_j, t_j) \in L_v: s_i \neq s_j, s_i \neq S_0, s_j \neq S_0, s_i \neq S_D, s_j \neq S_D \quad (4.9)$$

$$\sum_{\substack{k \in TS: \\ l \in L_{kv}}} X_l^{kv} \leq C_v X_l^v \quad \forall v \in V, \forall l \in L \quad (4.10)$$

$$X_l^v, X_l^{kv}, Y_r \in \{0,1\} \quad \forall v \in V, \forall l \in L, \forall r \in R, \forall k \in TS_r \quad (4.11)$$

As an objective function a weighted multi objective approach that maximizes the number of served participants, minimizes the total vehicle cost and total commuter cost, as suggested in Agatz et al. (2010) is used. Furthermore, an extra term to minimize the fleet size is also considered. Given a fixed number of vehicles, those that remain unused travel directly on the link of cost zero that connects the supply and demand node, S_0 and S_D . By including this link in

the maximization objective function, the total number of used vehicles is minimized. Note that finding the optimal fleet size is assumed to be a system design stage decision, rather than an operational stage decision, where the exact number of vehicles may be known. We estimate the cost of traveling on link (i, j) , $c_{i,j}$, using the travel time on the link. However, any other linear cost structure can be used. Note also that the cost of vehicles does not account for the idling time at any given station, whereas for the commuters, the idling time is included as a part of the total travel. The objective coefficients β_i , can be selected depending on the desired operational goals and the relative scale of each objective.

Constraint sets (1) to (5) deal with vehicle routing. Constraint (1) imposes flow conservation on the commuter vehicles V_r in all stations but supply node S_O . This forces all vehicles in V_r to depart the supply node. Constraints (2) and (3) rule out illogical flows between S_O and S_D , enforcing that vehicles cannot travel from S_D to any other station but S_O , and that for every vehicle, a link connecting S_D to S_O must exist. Constraint (4) handles dummy vehicles, and constraint (5) enforces transit vehicles schedules. These two constraint sets are not strictly required, as long as the decision variable is not forced to be zero. However, for clarity purpose, they are considered. Furthermore, it is reasonable to assume that transit vehicles will travel even though commuters are not travelling on them.

Constraint sets (6) – (8) route commuters. Constraint (6) performs the vehicle-trip-commuter matching. If the net outflow of any origin station O_k of all trips $k \in TS_r$ is one, commuter r is matched and all the other trips in TS_r need also to occur. Similarly, if a commuter is matched, all the destination stations of TS_r need to be reached. These two constraint sets together enforce the all-or-nothing matching strategy. In addition, it is worth noting that the net flow is used, meaning that station O_k could potentially be revisited. However, the set of links L_{kv}

only includes the feasible links for that particular trip k that are constrained by spatiotemporal constraints, as it will be discussed later, which limit the revisiting effect. Constraint (8) is the transshipment constraint on the commuters.

Constraints (9) and (10) are the connectivity constraints between commuters and vehicles. The first constraint set ensures that all vehicles carry at least one commuter (since one of the commuters has to operate the vehicle) and the second constraint set guarantees that the vehicle capacities are not violated.

3.5. Solution Approach

To solve the problem P 4 an iterative aggregation/disaggregation algorithm is proposed. The underlying idea is to decompose the problem into a master problem (MP) and a sub-problem (SP). In the master problem, the decision variable X_l^{kv} from the original formulation that takes care of the trip-vehicle-link assignment is aggregated over v , and the new variable X_l^k is defined as:

$$X_l^k = \sum_{v \in V} X_l^{kv} = \begin{cases} 1, & \text{if trip } k \text{ includes traveling on link } l \\ 0, & \text{otherwise} \end{cases} \quad (\text{Eq. 23})$$

Using this aggregation procedure, the number of decision variables and constraints from the original problem is considerably reduced, making the problem more tractable. The aggregation leads to a reduction of $(|V| - 1) \cdot |L| \cdot |K|$ in the number of binary variables and $(|V_r| - 1) \cdot |L_v| + (|V| - 1) \cdot |L|$ in the number of constraints. In addition, note that using this aggregation procedure, the set of feasible links does not proportionally increase with the number

of commuters or trips (as in the original formulation), since it is likely that multiple commuters share the same links in their paths, given the similarity of schedules.

After solving the aggregate problem and finding out which links each commuter should travel on, the vehicle-commuter assignment needs to be recovered by solving another problem - the sub-problem (SP). In the master-problem, it is ensured that the aggregate capacity (i.e., the sum of capacities of all the vehicles that travel on that link) of each link is not violated. Since in the SP commuters are assigned to vehicles, there is a possibility that a solution that respects each individual vehicle's capacity does not exist. In such instances, the SP can become infeasible. In this case, a new master problem solution needs to be generated.

The iterative aggregation/disaggregation algorithm steps are shown in Figure 22.

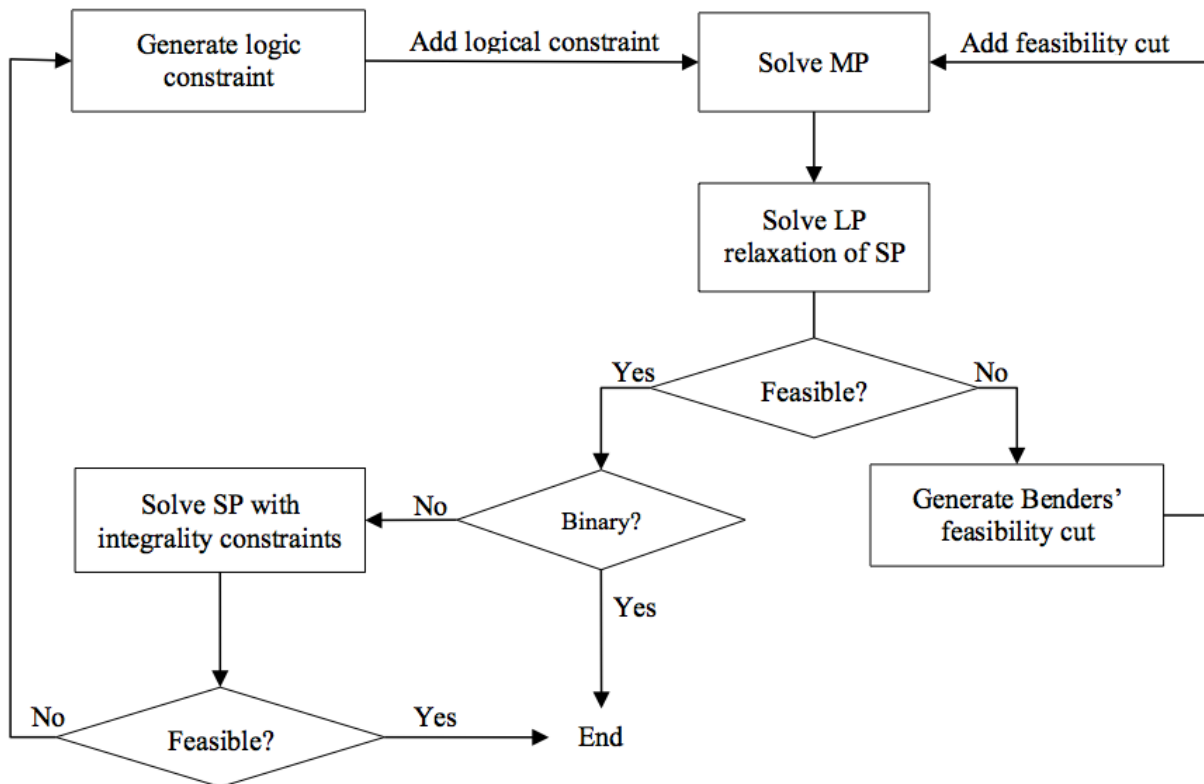


Figure 22: Aggregation/Disaggregation Iterative Algorithm Steps

The algorithm starts by solving the MP and use its solution to solve a linear relaxation of the SP. If the SP is infeasible, a Benders' feasibility cut is found and added to the MP. If the SP is feasible, the integrality of the solution is checked. If all variables are binary, this is the optimal solution; otherwise the SP is solved with the integrality constraints. If the solution is feasible, this is the optimal solution; otherwise, a logical constraint is added to the MP to eliminate this non-integer solution from the pool of feasible solutions. This process is repeated until the convergence criterion is met.

This algorithm is finite, since on each iteration a new constraint is added to the MP, shrinking the feasible region. Therefore, either an optimal solution is reached, or the entire feasible region is eliminated, implying that the problem is infeasible. In addition, note that in case the solution to the SP relaxation is feasible, but not integer, the SP is solved for a second time, this time enforcing the integrality constraints. If the solution turns out to be integral, we claim that the solution is the optimal solution. The reason for such claim is that the SP is a feasibility problem, i.e. does not have an objective function. Therefore, any integral solution that can satisfy the set of constraints in the SP is also optimal to the SP.

In the following subsections the master problem, the sub-problem, and the strategy to generate feasible links are defined.

3.5.1. Master Problem

In the master problem the decision variable that takes care of the trip-vehicle-link assignment (X_l^{kv}) is aggregated over the set of vehicles, $v \in V$. As a result, the particular vehicle in which each commuter is traveling is unknown; rather, it is known that a commuter performing trip k will be traveling on a link l . The trip-vehicle assignment is retrieved in the SP that is detailed in the following subsection. Problem P5 shows the proposed formulation for the master

problem. The objective function of the MP is the same as for the original problem, including the four objectives previously described. Constraints (1) – (5) are the same as for the original problem. Constraints (6) to (8) are equivalent to the original problem without the sum over v . Constraint sets (9) and (10) ensure that a link with trips has to have vehicles and that the link capacity is not exceeded, respectively. Constraint (10), as in the original problem is defined over every vehicle, enforcing vehicle capacity: here, under the aggregation procedure; the aggregate capacity of the link is being enforced.

P 5:

$$\max Z = \beta_1 \sum_{r \in R} Y_r - \beta_2 \sum_{v \in V_r} \sum_{\substack{l \in L_v: \\ i \neq j}} c_{i,j} X_l^v - \beta_3 \sum_{r \in R} \sum_{k \in TS_r} \sum_{l \in L_k} c_{i,j} X_l^k + \beta_4 \sum_{v \in V_r} \sum_{\substack{l \in L_v: \\ i=S_0, j=S_D}} X_l^v \quad (5.0)$$

s.t.

$$\sum_{\substack{l_i, s_i: \\ l=(s_i, t_i, s, t) \in L_v}} X_l^v = \sum_{\substack{l_j, s_j: \\ l=(s, t, s_j, t_j) \in L_v}} X_l^v \quad \forall v \in V_r, \forall t \in T, \forall s \in S - \{S_0\} \quad (5.1)$$

$$\sum_{\substack{l=(s_i, t_i, s_j, t_j) \in L: \\ s_i=S_D, s_j=S-\{S_0\}}} X_l^v = 0 \quad \forall v \in V_r \quad (5.2)$$

$$\sum_{l=(S_D, t_i, S_0, t_j) \in L} X_l^v = 1 \quad \forall v \in V_r \quad (5.3)$$

$$X_{l'}^v = 1 \quad \forall l' \in L'_v, \forall v \in V_{dummy} \quad (5.4)$$

$$X_l^v = 1 \quad \forall l \in L_t, \forall v \in V_t \quad (5.5)$$

$$\sum_{\substack{l \in L_k: \\ s_i=O_k}} X_l^k - \sum_{\substack{l \in L_k: \\ s_j=O_k}} X_l^k = Y_r \quad \forall r \in R, \forall k \in TS_r \quad (5.6)$$

$$\sum_{\substack{l \in L_k: \\ s_j=D_k}} X_l^k = Y_r \quad \forall r \in R, \forall k \in TS_r \quad (5.7)$$

$$\sum_{\substack{l=(t_i, s_i, t, s) \in L_k \\ s_i \neq 0_k, s_j \neq D_k}} X_l^k = \sum_{\substack{l=(t, s, t_j, s_j) \in L_k \\ s_i \neq 0_k, s_j \neq D_k}} X_l^k \quad \forall k \in TS_r, \forall t \in T, \forall s \in S - \{S_0, S_D\} \quad (5.8)$$

$$\sum_{\substack{v \in V_r \\ l \in L_v}} X_l^v \leq \sum_{\substack{k \in TS \\ l \in L_k}} X_l^k \quad \forall l = (s_i, t_i, s_j, t_j) \in L: s_i \neq s_j, s_i \neq S_0, s_j \neq S_0, s_i \neq S_D, s_j \neq S_D \quad (5.9)$$

$$\sum_{\substack{k \in TS \\ l \in L_k}} X_l^k \leq \sum_{\substack{v \in V \\ l \in L_v}} C_v X_l^v \quad \forall l = (s_i, t_i, s_j, t_j) \in L: s_i \neq s_j, s_i \neq S_0, s_j \neq S_0, s_i \neq S_D, s_j \neq S_D \quad (5.10)$$

$$X_l^v, X_l^k, Y_r \in \{0, 1\} \quad (5.11)$$

3.5.2. Sub-Problem

The SP is defined as shown in P 6. Note that this problem is a feasibility problem, and the objective function can be any constant as all the terms in the objective function from the original problem can be written in terms of the variables defined in the MP, thus becoming constant values on the SP. The parameters $U_l^k = X_l^{k*}$ and $U_l^v = X_l^{v*}$, where X_l^{k*} and X_l^{v*} are the optimal solutions to the MP.

Constraint (1) retrieves the vehicle-trip assignment. Constraint (2) imposes the condition that the vehicles available to the commuters' (V_r) cannot ride alone and constraint (3) is the vehicle capacity constraint.

P 6:

$$\min Z_{SP} = 0 \quad (6.0)$$

s.t.

$$\sum_{\substack{v \in V \\ l \in L_{kv}}} X_l^{kv} = U_l^k \quad \forall k \in TS, \forall l = (s_i, t_i, s_j, t_j) \in L_k: s_i \neq s_j \quad (6.1)$$

$$\sum_{\substack{k \in TS \\ l \in L_{kv}}} X_l^{kv} \geq U_l^v \quad \forall v \in V_r, \forall l = (s_i, t_i, s_j, t_j) \in L_v: s_i \neq s_j, s_i \neq S_0, s_j \neq S_0, s_i \neq S_D, s_j \neq S_D \quad (6.2)$$

$$\sum_{\substack{k \in TS: \\ l \in L_{kv}}} X_l^{kv} \leq C_v U_l^v$$

$$\forall v \in V, \forall l = (s_i, t_i, s_j, t_j) \in L_v: s_i \neq s_j, s_i \neq S_O, s_j \neq S_O, s_i \neq S_D, s_j \neq S_D \quad (6.3)$$

$$X_l^{kv} \in \{0,1\} \quad (6.4)$$

3.5.2.1. A note on Benders' Feasibility cut

If the relaxation of the SP leads to an infeasible solution, it required to find a Benders' feasibility cut that will reduce the feasible region so that the infeasibility is removed.

If the constraint set of the SP is defined as the general expression $Ax \geq b - By$, being $x = X_l^{kv}$ and y the corresponding U_l^k or U_l^v , the benders' feasibility cut will be of the form $(b - By)^T \bar{u} \leq 0$, being \bar{u} the unbounded ray of the corresponding dual problem of the SP.

3.5.3. Link Reduction Strategy

Considering all the possible links in the problem increases the size of the input sets to the optimization problem, and hence, its solution time. In this section, a link reduction strategy to reduce the size of the link sets based on the spatiotemporal properties of each trip is described. Each trip announced by the commuters is constrained by the physical location of the origin and destination points, and such temporal constraints imposed by the commuter such as the earliest departure time (ED), the latest arrival time (LA) and the maximum travel time budget (tt_B). Figure 23 depicts the definition of the departure and arrival time windows between any given pair of nodes s_i and s_{i+1} and the time windows that are used to find feasible links between these two nodes, including travel links and lingering links.

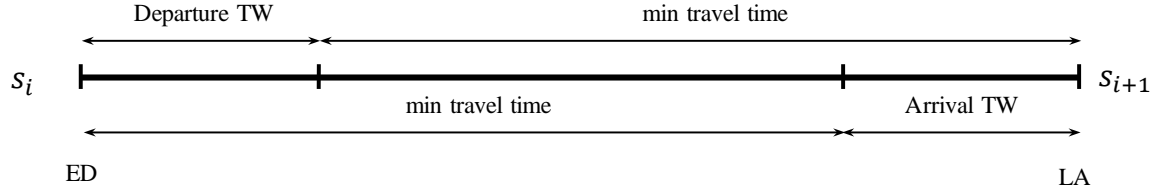


Figure 23: Time windows definition

For each of the trips announced by the commuters N shortest paths P_k are found such that either N is larger than a maximum number of paths allowed N_M (i.e $N_M = 1000$) or the travel time of the n^{th} path is larger than the travel time budget (tt_B) defined by the commuter for that particular trip k .

Given a feasible path p_n in P_k , for each node s_i in p_n the earliest arrival time ED_i and the latest arrival time LA_i to that node are defined, based on minimum travel time between nodes and the commuter travel time budget for the entire trip. Within the time window that the commuter can reach a node, it may travel to the next node or linger, as long it does not violate the time constraints to reach the next node. There is a particular case when two consecutive nodes s_i and s_{i+1} belong to the set S_t . If this occurs, the transit trip between s_i and s_{i+1} that depart during the available time window are added to the link set. In addition, an upper bound on the number of nodes visited during a trip is set to $maxNodes$. This step reduced the number of possible transfers and nodes visited by the commuters. This process is repeated for the N paths in P_k . The final set of links L_k for the trip k is the union of the traveling and lingering links. The logic on how to generate the set L_k is shown below.

1. for each trip k in TS_r :
2. find N -shortest Paths $P_k = \{p_1, \dots, p_n, \dots, p_N\}$ until $N > N_M$ or $tt(p_n) > tt_B$
3. for each p_n in P_k :
4. if $len(p_n) > maxNodes$:
5. remove p_n from P_k
6. continue
7. else:
8. for each node s_i in $p_n = \{s_1, \dots, s_i, \dots, s_M\}$:
- 9.

- 10. add travel links $L(s_i, t_i(\cdot), s_{i+1}, t_{i+1}(\cdot))$ to L_k
- 11. add lingering links $L(s_i, t_i(\cdot), s_i, t_i(\cdot) + 1)$ to L_k

The set of links on which vehicles v can travel, denoted by L_v , is the union of the set of trip links, $L_k, \forall k \in K$, the set of idling links, L_{id} , (which is formed based on the set of nodes that are visited by that vehicle), and links L_O and L_D which connect supply and demand nodes S_O and S_D together, and to all other nodes. The sets L_O and L_D are generated using the same concepts of available time windows on which vehicles can reach nodes and an extra link $L(S_O, t_0, S_D, t_0)$ is added to account for the unused vehicles. To impose transit schedules, for each transit vehicle a set of links corresponding to that transit schedule is defined. Note that although in the current approach the same set of links L_v is assigned to all vehicles in V_r , however, one could heuristically force vehicles in V_r to travel only on a subset of links based on the spatial distribution of homes and workplaces. For example, following on the example described in section 3.3, the range of one vehicle could be reduced to all links that connect nodes $\{1,2,3,4,10,11\}$, another vehicle to the links that connect nodes $\{5,6,7,8,9,12\}$ and the third vehicle to be able to visit all nodes.

Making a few observations it is possible to further reduce the number of feasible links. For example, “jump links” can be added to replace long idling periods. Under the current scenario, it is likely that a vehicle will remain at a workplace or transit station for a long time between the AM and the PM peaks. For those periods of inactivity, when there is no need for travel for any user, instead of defining 1 time step idling links, a single link that can cover the entire inactivity period can be introduced.

3.6. Numerical Tests

3.6.1. Data

Two different randomly generated scenarios to test the performance of the proposed solution method are used. The first scenario (SC1) has 10 commuters with 2 trips each, 3 workplaces and 2 transit stations. Given that each location can be a transfer point, this scenario has a total of 15 transfer points. Locations are GPS coordinates randomly sampled from a region nearby the Santa Ana Metrolink transit station and the Irvine Metrolink Station, both located in orange County, California. Workplaces are located at the station opposite to where the commuter resides (Figure 24, left) in order to simulate the potential of rail transit to service commute trips under the proposed system. The travel time matrix is computed using the great-circle distance between two coordinates, but this could be easily extended to account for real time travel time estimates.

The second scenario (SC2) has 25 commuters with 2 trips each, 4 workplaces and 2 transit stations (Figure 24, right). In this case there are 50 requests and 31 transfer points. Locations and distances are computed following the same procedure as for the previous scenario. A larger instance with 30 commuters (60 requests) and 4 workplaces, SC3-1, has also been tested without considering transit, which can be solved optimally in a reasonable amount of time.

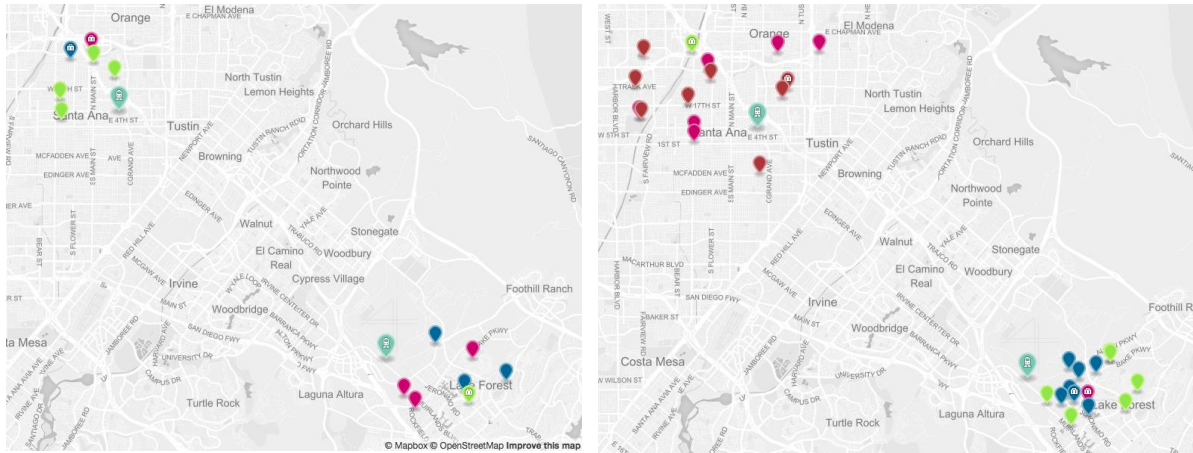


Figure 24: First scenario (left). Second scenario (right)

For each scenario different instances varying the values of the various system parameters are tested. Transit speed, transit frequency, commuter distribution and trip overlapping are among those parameters that can be tuned. The list of system parameters is provided in Table 7, where the first value listed is the default value. To encourage transit use, a higher average speed of 25mph on transit links is considered versus the 15mph assumed for regular vehicles. Each user makes two trips, from home to work and work to home and the travel time budget is assumed to be either 10% or 20% more than the shortest path travel time.

Two different parameters are introduced to measure the impact of the spatial distribution of homes and the temporal distribution of trips. Setting *balanced requests* to unity indicates that there is a perfect overlap of all commuter trips in terms of departure time windows. Otherwise, setting *balanced requests* to zero, the trip departure times are set to “random” within a predefined time window. Setting *commuter distribution* to “clustered” means that all commuters that work at the same workplace are nearby the same transit station, as shown in Figure 24. If it is set to “random” commuters are randomly assigned to workplaces.

Parameter	Values	Units
Time interval [dt]	5	minutes
Max. number of paths allowed [N_M]	1000	
Max. number of nodes in path [$maxNodes$]	10	
Travel time budget [tt_B]	1.1, 1.2	
Commuter vehicles speed	15	mph
Transit vehicles speed	25, 5	mph
Transit frequency	5, 15	minutes
Balanced requests	1, 0	
Commuter vehicle capacity [C_v]	5	
Transit vehicle capacity [C_v]	100	
Commuter distribution	Clustered, random	
β_1	$3 \cdot \text{ceil}(UB/100) \cdot 100$	
β_2	1	
β_3	1, 0	
β_4	$0, \beta_1/2$	
Gurobi setting: mipgap	Default	
Gurobi setting: mipgapabs	Default	
Solver setting: timelim	9000	seconds

Table 7: Car2work parameters settings.

In setting the utility coefficients (β_i) a strategy that prioritizes the matching of users rather than the cost and vehicle usage minimization is implemented. The underlying idea is to set β_1 larger than the upper bound (UB) on the commuters and vehicles travel time objective. An upper bound on the cost objectives is readily available from the expressed travel time budgets for each trip. If all users are matched, the total travel time cannot be larger than the sum of the individual travel time budgets. Similarly, because vehicles are always traveling with a commuter, an upper bound for the vehicle costs is also the total commuters travel time budget.

3.6.2. Results

The initial data inputs are built using MATLAB and AMPL is used for the iterative algorithm using Gurobi as a solver.

Table 8 summarizes the results of various instances for each scenario. The number of vehicles has been set to 4 for SC1 and to 10 for SC2, values which correspond to 40% of the

total number of commuters. From Table 8 the impacts of the various system parameters on the number of matched users, on the CPU time and on the objectives can be understood.

For example, when considering the transit network, the total commuter vehicle cost decreases and the number of matched users increases. In addition, it has a significant impact on the CPU time. In SC1, it increases the CPU time by a factor of 4. These results are in line with the expectations, as adding transit, increases the number of links, vehicles and stations, which leads to an increase of the number of decision variables and constraints.

In terms of transit frequency, for the same reasoning as above, a smaller frequency increases CPU time as more links and variables are added into the problem. However, under the current instances, a significant difference in terms of the total number of users matched is not observed. This effect is mostly because of the transit schedules, which if properly implemented; commuters can still take transit even with higher frequencies and similar levels of service. This is also reflected by the fact that commuter travel times remain invariant with the transit frequency for SC1.

Another important parameter to consider is whether or not the distribution of commuters is clustered or random. In a real case scenario it is more likely to have a random assignment of commuter to workplaces. Under these circumstances, solution times increase considerably. For example, looking at SC2-1 vs. SC2-3 and SC2-5 vs. SC2-6, which are comparable to each other as the only change is the commuter distribution, a nearly 10 fold increase is observed in both cases. Note that the number of links increases in those cases when considering a clustered commuter distribution assignment and the solution time decreases. This may seem counterintuitive, but having a clustered distribution may reshape the feasible region into having small sub regions with significantly better solutions that can be found quickly. In line of this

observation, SC3 has been solved setting the commuter distribution to clustered. The commuter distribution has also a significant impact on the commuter travel time, especially when the commuter requests are balanced. Note that having a clustered commuter distribution and a balanced commuter requests is the most symmetric and favorable case, leading to the best performance measures. Having balanced requests has a positive impact on the solution time; note the differences between SC1-3 and SC1-4.

Travel time budget has also a negative impact on solution time, as increasing it, adds flexibility and therefore more links and variables to the problem. In the set of instances reported in Table 8, however, travel time budget has been set to 10% increase of the total travel time.

For SC2 an optimal solution cannot be found in less than 9,000 seconds in 3 of the instances, SC2-2, SC2-4 and SC2-6. The first two include transit with unbalance requests and the latter has a random commuter distribution. However, it is worth pointing out that when transit is used the solutions found after 9,000 seconds are better than the equivalent no-transit alternative. For example, in SC2-1 and SC2-2, the number of matched users increases from 14 to 18 - an increase that is achieved even with the suboptimal solution obtained when transit is considered.

SC3-1 attempts to solve one a favorable case in terms of solution times, as the commuter distribution it is set to clustered and transit is not considered. Under these conditions the problem can be solved optimally in a reasonable amount of time. Note however, that only 14 out of the 30 users can be matched and this is mostly because the requests are not balanced and there are not enough vehicles to cover the demand.

Ins.	Balanced Requests	C. Distribution	Transit Use	Transit freq.	tt_B	CPU Time	#Veh. used	#Matched Users	#Links	Obj.	Obj1	Obj2	Obj3
SC1-1	1	Clustered	1	5	1.1	123	4	10	4503	-8762	-9000	41	197
SC1-2	1	Clustered	0	5	1.1	32	4	10	4382	-8686	-9000	82	232
SC1-3	1	Clustered	1	15	1.1	36	4	10	4439	-8764	-9000	38	198
SC1-4	0	Clustered	1	15	1.1	737	4	10	6076	-8741	-9000	54	205
SC2-1	0	Random	0	5	1.1	5329	10	14	7270	-20694	-21000	139	167
SC2-2	0	Random	1	5	1.1	9060*	10	18	7370	-26620	-27000	109	271
SC2-3	0	Clustered	0	5	1.1	545	10	13	8598	-15305	-15600	134	161
SC2-4	0	Clustered	1	5	1.1	9063*	10	18	8696	-21241	-21600	111	248
SC2-5	1	Clustered	0	15	1.1	1044	10	14	6514	-16511	-16800	105	184
SC2-6	1	Random	0	15	1.1	9042*	10	15	6054	-22107	-22500	167	226
SC3-1	0	Clustered	0	5	1.1	863	10	14	10947	-20744	-21000	110	146

Table 8: Experimental Results

Figure 25 shows the vehicle routes taken by the 4 vehicles on SC1-1 including transit during the AM peak. A general pattern that can be observed in the solutions involving transit is for a vehicle to pick up commuters, drop them at the transit station and wait for the incoming commuters to bring them to the desired workplace. This represents an ideal scenario, where all commuters have the same schedule and are grouped together by workplace. However, as already mentioned, this is not the most likely scenario in a real case implementation, unless significant densities can be achieved.

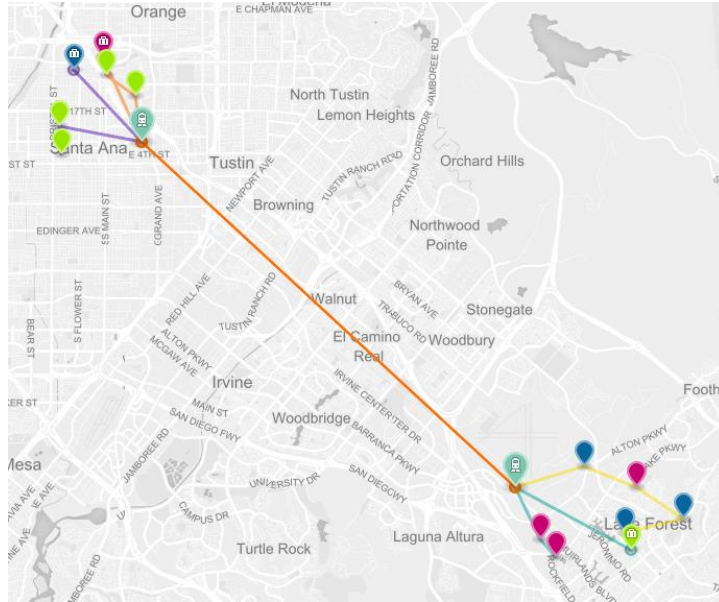


Figure 25: SC1-1 vehicle routes for the AM peak.

3.7. Conclusions And Future Work

Car2work builds upon some of the ideas on the shared-us mobility space and proposes a new shared-use mobility concept that has as its main goal connecting commuters with workplaces while leveraging the line-haul capabilities of existing public transit systems and guaranteeing a trip back home. As such, Car2work integrates with the existing transit network, and efficiently tackles the “last mile” problem that is a limiting characteristic of public transit.

It differs from the traditional dynamic-ridesharing approaches in the sense that 1) it is designed for recurring trips, focusing on commuter trips, 2) the concept of *drivers* is dropped and instead the concept of *vehicles* that carry at least one commuter when travelling is used, 3) commuters announce their trips in advanced and an automated all-or-nothing matching strategy is performed, and 4) because of the possibility of multiple trips, the routing decision variables are indexed over trips, not commuters

A formulation of the problem as a pure binary problem that is solved using an aggregation/disaggregation algorithm that renders optimal solutions has been presented. The underlying idea of the solution approach is to decompose the problem into a master problem (MP) and a sub-problem (SP), where in the master problem the variable that assigns trips to vehicles to links is aggregate over the vehicles, considerably reducing the number of decision variables and constraints. To recover the initial solution a feasibility problem is solved. As a result, various instances of the problem can be solved in reasonable amount of time, even when considering transit the transit network.

As a future work, efforts should focus on developing new strategies to reduce solution times so that a large-scale implementation of the concept can be simulated. To reduce solution times, the following could be tested: 1) a heuristic or another decomposition approach to solve the master problem, 2) a pre-processing step that would assign vehicles to commuters, fixing some of the decision variables, 3) reduce the transfer points to a subset of the stations, such as workplaces or transit stations, for example. Furthermore, on a large scale implementation of the concept and based on the observations from the simulations run, a cluster-first, route-second approach could be proposed, where commuters are grouped based on spatial-temporal proximity before solving the optimization problem.

4. CONCLUSIONS

Over the last decade shared-use mobility concepts have gone mainstream and they are expected to keep growing. However, most of shared-use mobility concepts suffer from two confounding issues: the lack of modeling tools to understand and simulate their behavior and the lack of integration with the existing transit network. To address those issues, this dissertation has focused on investigating the operational challenges of bikesharing systems, with an emphasis on the rebalancing operations and the modeling of a new mobility concept that builds upon existing carsharing ideas that successfully integrates with existing transit networks.

As a result, a methodological framework to solve the bikesharing rebalancing problem based on five core models has been proposed. The models are: 1) a demand forecasting model at the station level, 2) a station inventory model, 3) a redistribution needs model, 4) a user rebalancing model and 5) a vehicle-routing model. The novelty of this approach is that it is proactive instead of reactive, as the bike redistribution occurs before inefficiencies are observed, increasing system performance and, potentially, customer satisfaction, and uses the outputs of a machine learning technique to decompose the inventory and the routing problem. The decomposition approach proposed: 1) makes the problem scalable to large bikesharing systems, 2) allows for real time implementation, making routing decisions every time a vehicle completes a limited tour, 3) is responsive to operator inputs, and 4) can accommodate user-specific models. In addition, the underlying models based on historical data are self-adaptive, as they are constantly being retrained using the most recent data available.

The modularity of the approach provides great flexibility, as the current methodologies that have been proposed can be replaced as new modeling tools are developed, specially for the

demand forecasting model, which is built using machine learning techniques, a field that is in constant evolution.

Simulation results based on data associated with the Hubway Bikesharing system show that significant improvements to the overall system performance could have been made (over and above that being achieved under current operation) using the proposed modeling approach—achieving improvements of nearly 7% in the afternoon peak. More comprehensive tests using a full week of data demonstrate how the methodology could be used to evaluate such decisions as fleet size and vehicle capacity. To test the full potential of the framework, tests using real-time data under closed-loop control should be used (i.e., data that are not already the outcomes of operational decisions) and compared to the current rebalancing decisions made by the operator. However, access to such data was not possible. The impacts of the user-rebalancing model are incremental, even though they help to complement the vehicle routing bike relocations, especially for those where the estimated inefficiency is small. The decision on whether or not users should be used on the rebalancing is an operational decision that should be based on the cost structure of the operator. Introducing user relocation leads to similar system performance as it does adding an extra vehicle.

Car2work main goal is to connect commuters with workplaces while leveraging the line-haul capabilities of existing public transit systems and guaranteeing a trip back home.

It differs from the traditional dynamic-ridesharing approaches in the sense that it is designed for recurring trips, with a focusing on commuter trips, where commuters announce their (multiple) trips in advanced and an automated all-or-nothing matching strategy is performed, which guarantees that if the user participates on the system during the morning commute, the

same user will also participate during the evening commute, effectively guaranteeing a ride home.

Car2work is being modeled as a pure binary problem that is solved using an aggregation/disaggregation algorithm that renders optimal solutions. The underlying idea of the solution approach is to decompose the problem into a master problem (MP) and a sub-problem (SP), where in the master problem the variable that assigns trips to vehicles to links is aggregated over the vehicles, considerably reducing the number of decision variables and constraints. This decomposition allows finding optimal solutions in reasonable amount of time, even when the transit network is considered.

A range of scenarios has been simulated varying the values of the system parameters. Transit speed, transit frequency, commuter distribution and trip overlapping are among those parameters that can be tuned. The impacts on the number of users and the CPU times have been analyzed. For example, having clustered spatial and temporal distributions of commuters' lead to the best performance measures, both in terms of the total number of users participating in the system and the CPU times. These observations highlight the need of shared-used mobility systems to gain a certain level of spatial and temporal density to be successful, as until this threshold is not reached, operators struggle to provide good customer experiences, which in turn damage the potential for future growth. As a result, shared-use mobility companies tend to be highly capital intensive, due to the need to acquire user density. Another observation from the simulations is that when the system integrates with existing transit network, the automobile mileage decreases, which can lead to significant environmental impacts, at the expense, however, of user flexibility and comfort.

Future work should focus, other than finding more strategies to reduce computational times, on simulating the impacts of a large-scale implementation in areas that lack transit coverage to better understand what are those minimum density thresholds required to make Car2work economically viable while guaranteeing good levels of user satisfaction.

5. REFERENCES

- Agatz, N., Erera, A., Savelsbergh, M. & Wang, X., 2012. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2), pp.295–303.
- Agatz, N., Erera, A., Savelsbergh, M. & Wang, X., 2010. *Sustainable Passenger Transportation: Dynamic Ride-Sharing*, ERIM Report Series Research in Management. Available at: <http://hdl.handle.net/1765/18429>.
- Ahmed, M.M. & Abdel-Aty, M., 2013. Application of a Stochastic Gradient Boosting (SGB) Technique to Enhance the Reliability of Real-Time Risk Assessment Using AVI and RTMS Data. In Transportation Research Board 92nd Annual Meeting.
- Allal, C., Dumontet, F. & Parent, M., 1996. Design Tools for Public Cars Transportation Systems. In American Society of Civil Engineers. Applications of Advanced Technologies in Transportation Engineering. ASCE, pp. 6–18.
- Arnaldi, B. Cozot, R., Donikian, S. & Parent, M., 1996. Simulation Models for French Praxitele Project. *Transportation Research Record: Journal of the Transportation Research Board*, 1521(1), pp.118–125.
- Barth, M. & Shaheen, S.A., 2002. Shared-Use Vehicle Systems: Framework for Classifying Carsharing, Station Cars, and Combined Approaches. *Transportation Research Record: Journal of the Transportation Research Board*, 1791(1), pp.105–112.
- Barth, M. & Todd, M., 1999. Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research C*, 7(4), pp.237–259.
- Barth, M. & Todd, M., 2001. User behavior evaluation of an intelligent shared electric vehicle system. *Transportation Research Record: Journal of the Transportation Research Board*, pp.145–152.
- Barth, M. & Todd, M., 2004. User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. In Transportation Research Board 80th Annual Meeting. Washington: Proceedings of the 2004 Transportation Research
- Barth, M., Todd, M. & Murakami, H., 2000. Intelligent Transportation System Technology in a Shared Electric Vehicle Program. *Transportation Research Record: Journal of the Transportation Research Board*, 1731(1), pp.88–95.
- Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1), pp.238–252.
- Bishop, C., 2006. *Pattern recognition and machine learning*, Springer.
- Borgnat, P., Abry, P., Flandrin, P., Robardet, C., Rouquier, J. & Fleury, E., 2011. Shared

- bicycles in a city: a signal processing and data analysis perspective. *Advances in Complex Systems*, 14, pp.415–438.
- Braak, ter, C.J.F. & Vrugt, J.A., 2008. Differential Evolution Markov Chain with snooker updater and fewer chains. *Statistics and Computing*, 18(4), pp.435–446.
- Caggiani, L. & Ottomanelli, M., 2013. A Dynamic Simulation based Model for Optimal Fleet Repositioning in Bike-sharing Systems. *Procedia-Social and Behavioral Sciences*, 87, pp.203–210.
- Caggiani, L. & Ottomanelli, M., 2012. A Modular Soft Computing based Method for Vehicles Repositioning in Bike-sharing Systems. *Procedia-Social and Behavioral Sciences*, 54, pp.675–684.
- Chan, N.D. & Shaheen, S.A., 2012. Ridesharing in North America: Past, Present, and Future. *Transport Reviews*, 32(1), pp.93–112.
- Chauvet, F., Hafez, N., Proth, J. & Sauer, N., 1997. Management of a pool of self-service cars. *Journal of Intelligent Manufacturing*, 8(5), pp.459–465.
- Chauvet, F., Haouba, A.O. & Proth, J.-M., 1999. *Pool of Self-Service Cars: A Balancing Method*, INRA.
- Chemla, D., Meunier, F. & Wolfler Calvo, R., 2012. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2), pp.120–146.
- Cheu, R., Xu, J., Kek, A., Lim, WP. & Chen, W.L., 2006. Forecasting Shared-Use Vehicle Trips with Neural Networks and Support Vector Machines. *Transportation Research Record: Journal of the Transportation Research Board*, pp.40–46.
- City of Irvine, 2014. Irvine Transportation Network. *cityofirvine.org*. Available at: http://www.cityofirvine.org/cityhall/pw/itn_new/transit/routes/irvine.asp [Accessed June 2014].
- Codato, G. & Fischetti, M., 2006. Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54, pp.756–766.
- Contardo, C., Morency, C. & Rousseau, L.M., 2012. Balancing a dynamic public bike-sharing system. *CIRRELT*.
- Cortes, C.E., Matamala, M. & Contardo, C., 2010. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3), pp.711–724.
- Cortés, C. & Jayakrishnan, R., 2002. Design and Operational Concepts of High-Coverage Point-to-Point Transit System. *Transportation Research Record: Journal of the Transportation Research Board*, 1783(1), pp.178–187.

- Dror, M., Fortin, D. & Roucairol, C., 1998. Redistribution of Self-service Electric Cars: A Case of Pickup and Delivery. *INRIA*.
- Ferrucci, F., 2013. A New Forecasting Approach for Generating Stochastic Knowledge from Past Request Information and Utilizing the Stochastic Knowledge. In *Pro-active Dynamic Vehicle Routing*. Contributions to Management Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 167–183.
- Firnkorn, J. & Mueller, M., 2011. What will be the environmental effects of new free-floating car-sharing systems? The case of car2go in Ulm. *Ecological Economics*, 70(8), pp.1519–1528.
- Fricke, C., Gast, N. & Mohamed, H., 2012. Mean field analysis for inhomogeneous bike sharing systems. In 23rd Intern. Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms. pp. 365–376.
- Friedman, J.H., 2001. Greedy Function Approximation: A gradient Boosting Machine. *The Annals of Statistics*, 29, pp.1189–1232.
- Friedman, J.H., 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), pp.367–378.
- Froehlich, J., Neumann, J. & Oliver, N., 2009. Sensing and predicting the pulse of the city through shared bicycling. In 21st International Joint Conference on Artificial Intelligence (IJCAI-09).
- Gebhart, K. & Noland, R.B., 2013. The Impact of Weather Conditions on Capital Bikeshare Trips. In Transportation Research Board 92nd Annual Meeting.
- Gogate, V., Dechter, R., Bidyuk, B., Rindt, C. & Marca, J., 2006. Modeling Travel and Activity Routines Using Hybrid Dynamic Mixed Networks. In Transportation Research Board 85th Annual Meeting.
- Heling, M., Brown, T. & Samuelsen, G.S., 2010. *Energy and Charging Characterization of Shared Use Station Cars*, Advanced Power and Energy Program (APEP). University of California, Irvine.
- Heling, M., Saphores, J.-D. & Samuelsen, G.S., 2009. User Characteristics and Responses to Shared-Use Station Car Program: Analysis of ZEV-NET in Orange County, California. In Transportation Research Board 88th Annual Meeting.
- Henderson, J. & Fishman, A., 2013. Divvy: Helping Chicago's New Bike Share Find Its Balance. *Data Science for Social Good*. Available at: <http://dssg.io/2013/08/09/divvy-helping-chicagos-new-bike-share.html> [Accessed January 2014].
- Herbawi, W. & Weber, M., 2012. Modeling the Multihop Ridematching Problem with Time Windows and Solving It Using Genetic Algorithms. In 2012 IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI 2012). IEEE, pp. 89–96.

- Hernandez-Perez, H. & Salazar-Gonzalez, J.-J., 2004. Heuristics for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem. *Transportation Science*, 38(2), pp.245–255.
- Honda, Honda Worldwide | ICVS. *world.honda.com*. Available at: <http://world.honda.com/ICVS/project/index.html> [Accessed February 20, 2012].
- Hubway, 2012. Hubway Data Visualization Challenge. *The Hubway*. Available at: <http://hubwaydatachallenge.org/> [Accessed May 2013].
- Hubway, 2011. Hubway. *The Hubway*. Available at: <http://www.thehubway.com/> [Accessed July 2012].
- Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J. & Banchs, R., 2010. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4), pp.455–466.
- Karbassi, A. & Barth, M., 2003. Vehicle route prediction and time of arrival estimation techniques for improved transportation system management. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. pp. 511–516.
- Kek, A.G.H., Cheu, R.L., Meng, Q. & Fung, C.H., 2009. A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research E*, 45(1), pp.149–158.
- Kek, A.G.H., Cheu, R.L. & Xu, J., 2005. Trip Forecasting Models for Intelligent Community Vehicle Systems. In *Transportation Research Board 84th Annual Meeting*. Washington, pp. 1–17.
- Lee, K.-T., Lin, D.-J. & Wu, P.-J., 2005. Planning and Design of a Taxipooling Dispatching System. *Transportation Research Record: Journal of the Transportation Research Board*, 1903(1), pp.86–95.
- Li, X. & Quadrioglio, L., 2010. Feeder transit services: Choosing between fixed and demand responsive policy. *Transportation Research C*, 18(5), pp.770–780.
- Lu, C.-C., 2013. Robust Multi-period Fleet Allocation Models for Bike-Sharing Systems. *Networks & Spatial Economics*, pp.1–22.
- Masoud, N. & Jayakrishnan, R., 2014. A decomposition Algorithm to solve the multi-hop peer-to-peer ride-matching problem. In *Transportation Research Board 94th Annual Meeting*. pp. 1–18.
- Masson, R., Lehuède, F. & Peton, O., 2013. An Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Transfers. *Transportation Science*, 47(3), pp.344–355.
- Masson, R., Lehuède, F. & Peton, O., 2014. The Dial-A-Ride Problem with Transfers. *Computers & Operations Research*, 41, pp.12–23.

- Massot, M.-H., Allouche, J.-F., Bénéjam, E. & Parent, M., 1999. Praxitèle: Preliminary Results from the Saint-Quentin Station-Car Experiment. *Transportation Research Record: Journal of the Transportation Research Board*, 1666(1), pp.125–132.
- McCourt, M., 2008. *2008 Onboard Customer Satisfaction Survey*, Southern California Regional Rail Authority.
- McCulloch, W.S. & Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), pp.115–133.
- Meddin, R. & DeMaio, P., 2007. The Bike-Sharing World Map. Available at: <http://www.bikesharingworld.com> [Accessed May 8, 2013].
- Millard-Ball, A., Murray, G., Ter Schure, J., Fox, C. & Burkhardt, J., 2005. *Car-Sharing: Where and How it Succeeds*, Washington: Transportation Research Board.
- Mitchell, W., 2012. MIT Cities. *cities.media.mit.edu*. Available at: <http://cities.media.mit.edu/> [Accessed February 2012].
- Musso, A., Corazza, M.V. & Tozzi, M., 2012. Car Sharing Management Between Research and Innovation: the Rome Case. In *Transportation Research Board 91st Annual Meeting*. Washington, pp. 1–17.
- Nair, R. & Miller-Hooks, E., 2011. Fleet Management for Vehicle Sharing Operations. *Transport Science*, 45(4), pp.524–540.
- Nair, R., Miller-Hooks, E., Hampshire, R.C. & Busic, A., 2013. Large-Scale Vehicle Sharing Systems: Analysis of Vélib'. *International Journal of Sustainable Transportation*, 7(1), pp.85–106.
- National Climatic Data Center, 2012. Quality Controlled Local Climatological Data (QCLCD). NOAA. Available at: <http://www.ncdc.noaa.gov/land-based-station-data/quality-controlled-local-climatological-data-qclcd> [Accessed September 2012].
- Núñez, A.A., Sáez, D.A. & Cortes, C.E., 2013. *Hybrid Predictive Control for Dynamic Transport Problems*, Springer.
- Pfrommer, J., Warrington, J., Schildback, G. & Morari, M., 2014. Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems. *Ieee Transactions on Intelligent Transportation Systems*, 15(4), pp.1567–1578.
- Qiu, F., Li, W. & Zhang, J., 2014. A dynamic station strategy to improve the performance of flex-route transit services. *Transportation Research C*, 48, pp.229–240.
- Quadrioglio, L., Dessouky, M.M. & Ordóñez, F., 2008. Mobility allowance shuttle transit (MAST) services: MIP formulation and strengthening with logic constraints. *European Journal of Operational Research*, 185(2), pp.481–494.

- Rainer-Harbach, M., Papazek, P., Hu, B. & Raidl, G.R., 2013. Balancing Bicycle Sharing Systems: a Variable Neighborhood Search Approach. In *Evolutionary Computation in Combinatorial Optimization*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 121–132.
- Raviv, T. & Kolka, O., 2013. Optimal Inventory Management of a Bike-Sharing Station. *IIE Transactions*, 45(10), pp.1077–1093.
- Recker, W.W., Marca, J., Rindt, C. & Dechter, R., 2010. The Personal Travel Assistant (PTA): Measuring the dynamics of human travel behavior. *University of California Transportation Center*.
- Ridgeway, G., 2012. GBM R Package. *cran.r-project.org*. Available at: <http://cran.r-project.org/web/packages/gbm/gbm.pdf> [Accessed November 2012].
- Schuijbroek, J., Hampshire, R. & van Hoes, W.-J., 2013. Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems. *Tepper School of Business*.
- Shaheen, S.A., 2013. North American CarSharing: 2012-2013 Outlook. *Carsharing Association*. Available at: <http://www.slideshare.net/AlanWoodland/dr-susan-shaheen-north-american-carsharing-20122013-outlook> [Accessed August 13, 2015].
- Shaheen, S.A. & Chan, N., 2015. *Mobility and the sharing economy: Impacts Synopsis*, Transportation Sustainability Research Center.
- Shaheen, S.A. & Cohen, A., 2007. Worldwide carsharing growth: An international comparison. *Transportation Research Record: Journal of the Transportation Research Board*, 1992, pp.81–89.
- Shaheen, S.A. et al., 2014. *Public Bikesharing in North America During a Period of Rapid Expansion: Understanding Business Models, Industry Trends and User Impacts*, Mineta Transportation Institute.
- Shaheen, S.A., Martin, E. & Cohen, A., 2012. *Public Bikesharing in North America: Early Operator and User Understanding*, Mineta Transportation Institute.
- Shaheen, S.A., Sperling, D. & Wagner, C., 1998. Carsharing in Europe and North America: past, present, and future. *Transportation Quarterly*, 52, pp.35–52.
- Shang, J.S. & Cuff, C.K., 1996. Multicriteria pickup and delivery problem with transfer opportunity. *Computers & Industrial Engineering*, 30(4), pp.631–645.
- Shared Use Mobility Center, 2015. What is Shared Mobility? - Shared Use Mobility Center. *Shared Use Mobility Center*. Available at: <http://sharedusemobilitycenter.org/what-is-shared-mobility/> [Accessed August 2015].
- Terdiman, D., 2015. Lyft CEO says Lyft Line now accounts for majority of rides in San Francisco. *venturebeat.com*. Available at: <http://venturebeat.com/2015/03/16/lyft-ceo-says->

lyft-line-now-accounts-for-majority-of-rides-in-san-francisco/ [Accessed March 2015].

Time and Date AS, 1995. timeanddate.com. *timeanddate.com*. Available at:
<http://www.timeanddate.com/> [Accessed November 2012].

Wu, T., Xie, K., Xinpin, D. & Song, G., 2012. A online boosting approach for traffic flow forecasting under abnormal conditions. In 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, pp. 2555–2559.