A data-driven discrete simulation-based optimization algorithm for car-sharing service design

Tianli Zhou [*a], Carolina Osorio [†c], and Evan Fields [‡b]

[a] *Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, USA*
[b] *Operations Research Center, Massachusetts Institute of Technology, USA*
[c] *Department of Decision Sciences, HEC Montréal, Canada*

Abstract

This paper formulates a discrete simulation-based optimization (SO) algorithm for a family of large-scale car-sharing service design problems. We focus on the profit-optimal assignment of vehicle fleet across a network of two-way (i.e., round-trip) car-sharing stations. The proposed approach is a metamodel SO approach. A novel metamodel based on a mixed-integer program (MIP) is formulated. The metamodel is embedded within a general-purpose discrete SO algorithm. The proposed algorithm is validated with synthetic toy network experiments. The algorithm is then applied to a high-dimensional Boston case study using reservation data from a major US car-sharing operator. The method is benchmarked versus several algorithms, including stochastic programming. The experiments indicate that the analytical network model information, provided by the MIP to the SO algorithm, is useful both at the first iteration of the algorithm and across subsequent iterations. The solutions derived by the proposed method are benchmarked versus the solution deployed in the field by the car-sharing operator. Via simulation, the proposed solutions improve those deployed with an average improvement of profit of 6% and of vehicle utilization of 3%.

The combination of the problem-specific analytical MIP with a general-purpose SO algorithm enables the discrete SO algorithm to: (i) address high-dimensional problems, (ii) become computationally efficient (i.e., it can identify good quality solutions within few simulation observations), (iii) become robust to the quality of the initial points and of the stochasticity of the simulator. More generally, the information provided by the MIP to the SO algorithm enables it to exploit problem-specific structural information. This leads to an algorithm with both asymptotic convergence guarantees as well as good short term performance (i.e., performance given few simulation observations). We view this general idea of combining analytical MIP formulations with general-purpose SO algorithms, or more broadly with general-purpose sampling strategies of high-resolution data, as an innovative and promising area of future research.

Keywords: discrete simulation-based optimization, metamodel, large-scale car-sharing fleet allocation

[*] tzhou90@mit.edu
[†] carolina.osorio@hec.ca
[‡] efields@mit.edu

1 Introduction

In recent years, the most successful trend in the space of urban mobility services has been the widespread use of shared mobility services, such as ride-sharing, car-sharing, bike-sharing and, most recently, scooter-sharing (Shaheen and Chan, 2016). Car-sharing has become a popular transportation mode in urban areas. Its deployment, as of 2010, covered over 31,600 vehicles in over 1,100 cities in 26 countries with over 1 million members (Shaheen and Cohen, 2013). The car-sharing literature has studied its potential to reduce the transportation cost of households (Duncan, 2011), to complement private-vehicle ownership (Shaheen and Cohen, 2013; Becker et al., 2017) and public transportation systems (Chiraphadhanakul 2013 Chapter 4, Nair and Miller-Hooks 2014, Zhou 2015 Chapter 3), as well as to mitigate greenhouse gas emissions and total vehicle miles traveled (Firnkorn and Müller, 2011; Shaheen and Cohen, 2013).

Major technology companies have been behind the rapid growth of these shared mobility services. The operators of these services collect disaggregate data of vehicle usage and of client (user) behavior. This data provides a high-resolution description of the interaction of demand and supply. This paper is motivated by the following research question: how can we exploit the rich disaggregate information in this data to optimize the design and the operations of these new urban mobility services?

A current trend among major technology companies is to design optimization methods that exploit the rich information in their disaggregate data. Companies are building high-resolution simulators of their services that sample directly from their disaggregate data and provide a disaggregate description of the performance of their services (e.g., Greenhall (2016)). Hence, the next generation of mobility optimization algorithms will increasingly perform optimization based on models that provide a disaggregate description of mobility. This paper addresses this need. It formulates a car-sharing optimization problem as a simulation-based optimization (SO) problem, and proposes a computationally efficient SO algorithm. We use a disaggregate car-sharing service simulator, which was developed in collaboration with Ford and with the car-sharing operator Zipcar (Fields et al., 2017). The simulator samples from disaggregate car-sharing reservation data to estimate (disaggregate) demand (i.e., it yields a set of desired reservations) and then provides a stochastic mapping of how this demand interacts with supply to yield disaggregate reservations (i.e., a final set of realized reservations). The proposed algorithm is an example of how abundant disaggregate mobility data can be used to perform large-scale (e.g., city-scale) optimization.

This paper focuses on two-way (i.e., round-trip) car-sharing services with an application to a Boston case study with Zipcar data. Zipcar is a major car-sharing service provider in the US. It is also one of the world's largest car-sharing service provider with operations in more than 500 cities worldwide. It has deployed over 12,000 vehicles around the world (Zipcar, 2017). Currently, Zipcar offers two-way service, one-way station-based service and free-floating service. Two-way is the primary service mode for Zipcar and the foundation of its business. Studying the optimization of its two-way service is critical for Zipcar's business.

The optimization problem studied in this paper is the optimal spatial allocation of a fleet of two-way car-sharing vehicles to a set of stations. This is a tactical decision that car-sharing operators typically make on a monthly basis. The corresponding optimization problem is solved offline. As is detailed in Section 2.1, the most common approach to address these optimization problems is to aggregate the data such as to estimate parameters of a mathematical program, such as a mixed-integer programming model (MIP) or a stochastic programming model (SP). These mathematical programs provide an aggregate description of both demand and of the interaction of demand and

supply. This aggregate description enables their computational tractability and their scalability (i.e., their use for large-scale instances). Nonetheless, through this aggregation a wealth of information of the intricate interactions between demand and supply is lost. We propose a technique that embeds MIP information within a general-purpose discrete simulation-based optimization algorithm (DSO). This combination allows to exploit both (i) the detailed information from the disaggregate car-sharing data that underlies the simulator and (ii) the computational scalability and efficiency characteristic of MIPs.

The contributions of this paper can be summarized as follows.
Disaggregate data-driven technique. The most traditional approach to car-sharing service optimization has been analytical optimization. This comes at the cost of a simplified description of demand and of demand-supply interactions. In this work, our goal is to acknowledge both the intricacy of a car sharing service (e.g., intricate demand distribution, intricate demand-supply interactions), as well as the availability of high-resolution data. Hence, we propose a method that relies heavily on the rich reservation data and uses limited modeling assumptions. The information captured in the data about the underlying demand distribution and demand-supply interactions is preserved and exploited at a disaggregate level. To the best of our knowledge, this is the first work to design an algorithm that preserves this high-resolution information of the data (i.e., does not merely aggregate the disaggregate data) for car-sharing fleet allocation optimization. Case studies with data from Zipcar's Boston market are carried out.

The proposed algorithm contributes to the DSO literature in the following three ways.
1. Enhanced scalability. The proposed algorithm is suitable to address high-dimensional fleet allocation problems. In Section 4.3 and 4.4, we use it to address a Boston metropolitan area case study with 315 stations. General-purpose DSO algorithms have been extensively used to tackle problems with roughly 20 decision variables. We achieve scalability by formulating and embedding within the general-purpose DSO algorithm Adaptive Hyperbox Algorithm (AHA) (Xu et al., 2013) information from a MIP. This yields the proposed DSO algorithm, which we call MetaAHA. The approach combines the merits of both analytical optimization methods (i.e., tractability and scalability) and of simulation-based optimization methods (i.e., we can sample directly from the disaggregate data to enable a detailed description of demand and of demand-supply interactions). Our enhanced scalability comes at the cost of proposing an algorithm tailored for a specific class of fleet allocation problems, while the general-purpose DSO algorithms can be used for a broader class of problems.

2. Enhanced computational efficiency and robustness to initial points. The proposed algorithm identifies good quality solutions within few iterations (i.e., when few simulation observations are available). This differs from most DSO literature which is focused on asymptotic performance. This efficiency is achieved through the novel metamodel formulation which embeds a non-simulation-based representation (a MIP formulation) of the optimization problem. In other words, the simulator is no longer treated as a black box, instead analytical problem-specific information is embedded within the SO algorithm. The results of Section 4 indicate that this analytical structural information is the key to achieving computational efficiency. Moreover, the proposed method preserves the asymptotic convergence and performance properties of the underlying DSO algorithm. In particular, like AHA, the proposed algorithm remains a locally convergent algorithm. The combination of the proposed metamodel along with a general-purpose DSO algorithm yields an algorithm with both good short-term and asymptotic performance properties. Moreover, the metamodel enables the general-purpose algorithm to become robust to the quality of initial solutions.

3. Metamodeling for DSO The main feature of the proposed algorithm is the formulation of a

metamodel, (i.e., an analytical approximation of the simulation-based objective function) that has a physical term that is problem-specific. Such metamodel ideas for transportation problems have been successfully formulated for various continuous SO problems. This is the first paper that extends these ideas to the DSO setting. Moreover, this is the first approach to formulate a metamodel based on a MIP. The paper shows that by using such metamodel ideas, high-dimensional DSO problems can be addressed in a computationally efficient way. Since fundamental OR transportation optimization problems (e.g., routing) are naturally formulated as discrete optimization problems, the ideas of this paper lay the foundations for a variety of important and difficult transportation problems to be addressed efficiently with data-driven simulation-based network models.

Section 2 discusses related literature. Section 3 formulates the proposed methodology. Its performance is evaluated and benchmarked in Section 4 with experiments on both synthetic toy networks and Boston networks. Conclusions are presented in Section 5. Algorithmic details are presented in Appendix A. The formulation of the SP model that is used as a benchmark in Section 4.4 is given in Appendix B. Additional implementation detailes are given in Appendix C.

2 Literature review

2.1 Vehicle-sharing service optimization

The main types of car-sharing service are two-way, one-way station-based, free-floating and peer-to-peer. For full definitions, see for instance Schmöller et al. (2015). A station is a location with a certain number of vehicle-sharing parking spots. Two-way services consist of a set of vehicles parked at a set of fixed stations. In advance, customers reserve a vehicle for a given duration and a given start time. They then pick-up and drop-off the vehicle from the same predetermined station. Reservations can be made from several months in advance to minutes in advance. There is no upper limit on the duration of a reservation. As of July 2015, there were an estimated 1.17 million two-way service members along with 0.31 million one-way service members in the United States (NCSL, 2017).

Detailed reviews of vehicle-sharing studies are given in Jorge and Correia (2013); Brandstätter et al. (2016). Table 1 summarizes some of the recent vehicle-sharing service design literature. The column "Optimization" indicates whether the method is analytical, simulation-based or a combination of both. The column "Context" specifies the type of vehicle-sharing service (one-way, two-way, free-floating) and the type of vehicle (bike, car). The column "Case study size" indicates, for the main case study of each paper, the number of sites (e.g., locations, regions, stations), of integer and of continuous variables (including both decision variables and auxiliary variables). Cells are left blank for cases where these numbers are not directly reported. The "Problem" column specifies the type of decisions the problem addresses.

The most popular approach to address vehicle-sharing (both car- and bike-sharing) service design problems across all service types (two-way, one-way, floating) is the use of analytical mixed integer programming (MIP). Studies with deterministic demand include Correia and Antunes (2012), Chiraphadhanakul (2013, Chapter 4), Correia et al. (2014), Nair and Miller-Hooks (2014), Zhou (2015, Chapter 3). Past work in the field has also accounted for demand uncertainty by using a parametric probability distribution for demand combined with optimization methods such as stochastic programming and robust optimization (O'Mahony 2015, Chapter 4, Lu et al. 2017, He et al., 2017).

Table 1: Summary of recent related vehicle-sharing service design papers

Paper	Optimization		Context					Case study size			Problem			
	Analytical	Simulation-based	One-way	Two-way	Free-floating	Bike-sharing	Car-sharing	Site	Integer	Continuous	Site location	Fleet assignment	Station capacity	Other
Correia and Antunes (2012)	✓		✓				✓	75		0	✓	✓	✓	Rebalance fleet
Cepolina and Farina (2012)		✓	✓				✓	11	9	0			✓	Fleet size
Chiraphadhanakul (2013, Chapter 4)	✓		✓			✓		27	27		✓			Route user flow
Correia et al. (2014)	✓		✓				✓	116		0	✓	✓		Select trips
Nair and Miller-Hooks (2014)	✓		✓			✓	✓	64	4420	6295	✓	✓	✓	Route user flow
Boyacï et al. (2015)	✓		✓				✓	100	All together $\sim 10^5$		✓		✓	Determine fleet size, regions served by each station and number of relocation personnel, rebalance fleet
Deng (2015, Chapter 5)		✓	✓				✓	8	11	2		✓	✓	Determine fleet size, rebalance fleet
Jorge et al. (2015)	✓		✓	✓			✓	391		0		✓	✓	Select trips, rebalance fleet
O'Mahony (2015, Chapter 3)	✓		✓			✓		300				✓	✓	
Zhou (2015, Chapter 4)	✓		✓			✓		30	5133	$\sim 1.5 \times 10^7$	✓			Route user flow
Jian et al. (2016)	✓	✓	✓			✓		466	932	0		✓	✓	
Lu et al. (2017)	✓		✓	✓	✓		✓	9				✓	✓	Route user flow, rebalance fleet
He et al. (2017)	✓				✓		✓	61			✓			Route user flow, rebalance fleet
This paper	✓	✓		✓			✓	315	315	0		✓		

Car-sharing demand-supply interactions are intricate to model, yet are critical to account for when planning and operating car-sharing services. Studies of car-sharing demand include Millard-Ball et al. (2005); Stillwater et al. (2009); Ciari et al. (2013); De Lorimier and El-Geneidy (2013); Coll et al. (2014); Ciari et al. (2014, 2016b). The analytical modeling of demand involves accounting for how the distribution of demand varies as a function of space, time, user-specific attributes (e.g., value of time, willingness to walk, trip purpose) and other transportation system attributes (e.g., alternative travel modes for that user and that trip purpose). Moreover, for two-way car-sharing, the analytical modeling of the interaction of demand and supply is particularly difficult due to the often low supply capacity: there are typically few car-sharing parking spots available at each station. Hence, if a user does not find a vehicle available at the desired time and station, the user may opt out of renting a vehicle (the demand is said to be lost, and the historical reservation data is said to be truncated) or may opt into renting a nearby (e.g., in space, in time) reservation (the demand is said to spillover or spillback and the historical reservation is said to be censored). For a detailed description of truncation and censoring in the context of car-sharing, see Fields et al. (2021). The likelihood of truncation and of censoring can depend on user characteristics (e.g., willingness to walk, car ownership), on trip attributes (e.g., trip purpose) as well as on the general mobility system (e.g., availability of other competitive travel alternatives). Additionally, given this low supply capacity, it is important to account for the temporal order in which users make reservations. In other words, modeling the *first-come-first-reserve* principle (i.e., the fact that reservations are prioritized or processed in the order of their creation time) is important. Due to the difficulties of accurately modeling car-sharing demand, as well as demand-supply interactions, we propose to directly use disaggregate car-sharing reservation data that embeds a detailed description of the interaction of demand and supply.

Compared to pure analytical models, stochastic simulators enable a more detailed modeling of demand and supply uncertainties, and of demand-supply interactions, their use to address optimization problems of realistic dimensions remains intricate. In the context of vehicle-sharing, simulation tools have mostly been used to evaluate the performance of fleet allocations obtained from analytical models, i.e., the simulator is used to perform what-if analysis (Cepolina and Farina 2012, O'Mahony 2015 Chapter 5, Ciari et al. 2015). Various simulation studies that account for car-sharing (Ciari et al., 2009, 2016a; Balac et al., 2016, 2017) have been carried out with the MATSim transportation simulation software (MATSim, 2018). Studies, such as Cepolina and Farina (2012) and Deng (2015, Chapter 5), have included the simulator as part of an optimization framework and have resorted to general-purpose black-box optimization algorithms such as simulated annealing and particle swarm optimization. The study of Jian et al. (2016) exploited problem-specific information to yield gradient-type information. Interestingly, Jian et al. (2016) use the solution of an analytical linear integer program as the initial solution for a simulation-based optimization algorithm. Such an approach is also used as benchmark method in the case studies of this paper. Of particular notice is the large-scale bike-sharing optimization instance studied in Jian et al. (2016), which considers a set of 466 stations.

2.2 Discrete simulation-based optimization (DSO)

In this paper, in order to enable the direct use of disaggregate car-sharing reservation data for optimization, we formulate the problem as a DSO problem. The problem has a simulation-based objective function with discrete decision variables. Constraints are analytical (i.e., they are not simulation-based). The main challenges of addressing such problems are the following. There is no analytical expression available for the objective function, hence traditional (analytical) discrete optimization algorithms cannot be used. The objective function can only be estimated by running

a set of stochastic simulation replications. DSO problems inherit the curse of dimensionality of discrete analytical problems. Since simulation is used, the objective function is often an intricate (e.g., non-convex) function of the decision variables with several local optima.

There are a variety of DSO algorithms in the literature; recent reviews include Nelson (2010) and Hong et al. (2015). DSO algorithms include Convergent Optimization via Most-Promising-Area Stochastic Search (COMPASS) (Hong and Nelson, 2006), Adaptive Hyperbox Algorithm (AHA) (Xu et al., 2013), R-SPLINE (Wang et al., 2013), and cgR-SPLINE (Nagaraj, 2014). Methods that aim to identify solutions with good performance at an early stage (i.e., within few simulations) include an extension of COMPASS known as the Industrial Strength COMPASS (ISC) (Xu et al., 2010), as well as extension of AHA known as ISC-AHA (Xu et al., 2013). For DSO problems with a finite, and often low-dimensional, feasible region, ranking-and-selection (R&S) techniques, such as Chick and Inoue (2001); Frazier et al. (2008), are commonly used. An R&S review can be found in Swisher et al. (2003).

DSO algorithms are most often designed: (i) as general-purpose algorithms, i.e., they can be used to address a broad family of optimization problems, their use is not limited to transportation problems, and (ii) based on asymptotic convergence properties, there is limited focus on their short-term (i.e., small sample performance). The performance of these general-purpose DSO algorithms is typically illustrated with low-dimensional problems (e.g., around 20 decision variables). Few studies have reported higher-dimensional instances. The work of Xu et al. (2013) reported experiments where AHA successfully addressed problems with up to 100 decision variables. Developing DSO algorithms that are suitable for high-dimensional problems remains a challenge. Past studies, such as Xu et al. (2013), illustrate that for locally convergent general-purpose DSO algorithms, the quality of the final solution is sensitive to the quality of the initial solution. Hence, there is also an interest to develop algorithms with enhanced robustness to the quality of the initial solution.

There is a lack of studies that evaluate the performance of general-purpose DSO algorithms for high-dimensional problems and under tight computational budgets (i.e., within few simulation runs). Nonetheless, when using simulators to address optimization problems, practitioners often use the algorithms under tight computational budgets (e.g., they terminate the algorithm once a fixed time or a fixed number of iterations have elapsed). Hence, there is a need for computationally efficient algorithms. These are algorithms that provide solutions with improved performance (compared to initial solutions or solutions deployed in the field) within few simulation runs.

This paper focuses on metamodel SO approaches. A metamodel is an analytical approximation of a simulation-based function. In this paper, we consider metamodels of the DSO objective function. In past work, we have formulated metamodel SO algorithms for various continuous SO transportation problems (Osorio and Nanduri, 2015; Chong and Osorio, 2017; Zhang et al., 2017; Chen et al., 2019; Osorio, 2019). A recent review of metamodel SO methods appears in Osorio and Chong (2015). A more detailed description of commonly used metamodels is given in Section 3.2. To the best of our knowledge, the use of metamodel approaches for DSO has been limited to low-dimensional problems (with up to 15 decision variables). In the broader area of transportation (i.e., not limited to vehicle-sharing) DSO has been used in studies such as Jung et al. (2014); Chen et al. (2015); Sebastiani et al. (2016); Jian et al. (2016); Boyacï et al. (2017).

General-purpose functions (e.g., low-order polynomial functions, radial-basis functions, Kriging functions) are the most common metamodel choice both for discrete (Xu 2012, Sun et al. 2014, Salemi 2014 Chapter 4, Xie et al. 2016) and for continuous (Jones et al., 1998; Barton and Meckesheimer, 2006; Wild et al., 2008; Kleijnen et al., 2010; Ankenman et al., 2010) SO problems. Osorio and Bierlaire (2013) formulate a metamodel that combines a general-purpose function

with a problem-specific functional approximation of the SO objective function.

3 Methodology

This section presents the proposed methodology. The fleet allocation problem is formulated in Section 3.1. The general metamodel SO framework is discussed in Section 3.2. The metamodel for the considered car-sharing fleet allocation problem is formulated in Section 3.3 and the proposed algorithm is described in Section 3.4. The car-sharing network simulator used in this paper as well as the role of the car-sharing data are summarized in Section 3.5.

3.1 Fleet allocation problem formulation

We consider a two-way car-sharing system from the perspective of the car-sharing operator. The fleet allocation problem is to assign a fleet of vehicles across a network of stations such as to maximize the expected profit. We also refer to this problem as the fleet assignment problem. The fleet allocation problem is studied for a given finite time horizon, which we refer to as the planning period. To formulate the problem, we introduce the following notation.

x_i	number of cars assigned to station i (decision variable);
$\boldsymbol{x}$	vector of x_i's for all $i \in \mathcal{I}$;
$R(\boldsymbol{x}; \boldsymbol{q}_1)$	random variable representing the revenue;
$g(\boldsymbol{x}; \boldsymbol{q}_1)$	expected profit (DSO objective function);
c_i	cost, over the planning period, of a parking space at station i;
$\boldsymbol{q}_1$	exogenous simulation parameter vector (e.g., reservation pricing);
N_i	capacity of station i (i.e., number of parking spots);
X	total fleet size (i.e., number of cars to assign);
I	total number of stations;
$\mathcal{I}$	index set of all stations, $\mathcal{I} = \{1, 2, \ldots, I\}$;
$\mathcal{F}$	feasible region.

The problem is formulated as follows:

$$\max_{\boldsymbol{x}} \quad g(\boldsymbol{x}; \boldsymbol{q}_1) = E[R(\boldsymbol{x}; \boldsymbol{q}_1)] - \sum_{i \in \mathcal{I}} c_i x_i \tag{1}$$

subject to

$$\sum_{i \in \mathcal{I}} x_i \leq X \tag{2}$$

$$x_i \leq N_i \quad \forall i \in \mathcal{I} \tag{3}$$

$$x_i \in \mathbb{Z}_+ \quad \forall i \in \mathcal{I}. \tag{4}$$

The objective function represents the expected profit for a given fleet assignment vector, $\boldsymbol{x}$. It is defined as the difference between the expected revenue $E[R(\boldsymbol{x}; \boldsymbol{q}_1)]$ and the costs. The expected revenue is a simulation-based function, estimates of which are obtained via simulation. The cost parameters, c_i, are exogenous. In this work, they represent parking space leasing fees. Constraint (2) bounds the total number of cars assigned across all stations with the fleet size. Constraint (3) bounds the number of cars assigned to each station i with the space capacity of the station. The

number of cars assigned to each station are assumed to be non-negative integers (Constraint (4)). Constraints (2)-(4) specify the feasible region, $\mathcal{F}$.

The expectation in the objective function accounts for the stochasticity in the simulation process. The simulator, which is summarized in Figure 1 and described in more detail in Section 3.5, combines a sampling procedure that samples from a set of car-sharing reservation data and an assigning procedure that determines whether a reservation request will be satisfied and how it will be satisfied. In other words, realizations of the revenue random variable R are obtained by sampling from car-sharing reservation data. The sources of uncertainty include: (i) a stochastic process that samples from the historical reservation data to obtain a realization of disaggregate latent demand (or desired reservations); (ii) a stochastic description of how demand and supply interact and how truncation and censoring occur (e.g., probability with which a user, for which their desired reservation is not available, decides to opt out of making a reservation or decides to find a substitute reservation).

The challenges of addressing DSO problems, such as Problem (1)-(4), were detailed in Section 2.2. Given these challenges, we propose an algorithm that at every iteration, uses the set of estimates of g obtained so far to formulate and solve an (approximate) analytical discrete problem that: (i) provides good quality solutions to the underlying DSO problem, (ii) can be solved efficiently for high-dimensional instances, and (iii) can be solved with a variety of widely-used commercial solvers. As is detailed below, the proposed methodology combines the advantages of using a simulation-based model (which allows for the use of more detailed models and historical data) and those of an analytical mathematical program (which allows for computational tractability and scalability).

3.2 General metamodel approach

Let us first briefly present the main ideas of the metamodel SO approach, which are based on the continuous SO framework of Osorio and Bierlaire (2013). To formulate the problem, we introduce the following notation.

k	iteration index of the SO algorithm;
m_k	metamodel at SO iteration k;
$\boldsymbol{\beta}_k$	vector of metamodel parameters at SO iteration k, element i is denoted $\beta_{k,i}$;
$\boldsymbol{z}$	vector of endogenous variables;
$\boldsymbol{q}_2$	vector of exogenous parameters;
g_A	approximation of g (Equation (1)) derived by the analytical network model;
h	constraints of the analytical network model.

Recall that a metamodel is an analytical approximation of a simulation-based function. We consider a metamodel of the DSO objective function (1). Based on the idea of Osorio and Bierlaire (2013), the metamodel is defined by (5) as the sum of a problem-specific function (g_A) and a general-purpose linear function (term within parenthesis of (5)). We specify the problem-specific function (g_A) as the analytical objective function of a mathematical program (more specifically of a mixed-integer linear fleet assignment problem), which embeds a simplified representation (compared to the simulator) of the mapping between the supply configuration ($\boldsymbol{x}$) and the expected profit (g of (1)). The goal of g_A is to provide a good analytical approximation of the simulation-based objective function for the considered problem. Nonetheless, this analytical approximation is not expected to be accurate (due to the more detailed and intricate models of demand and of supply embedded in

the simulator, that are not accounted for in the mathematical program). Hence, the metamodel (5) can be thought of as the objective function of a MIP that is corrected for parametrically by both a scaling term (scalar $\beta_{k,0}$ of (5)) and an additive linear error term (term within parenthesis of (5)).

At a given iteration k of the SO algorithm, we solve the following analytical problem, referred to as the metamodel optimization problem.

$$\max_{\boldsymbol{x},\boldsymbol{z}} \quad m_k(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\beta}_k,\boldsymbol{q}_2) = \beta_{k,0} g_A(\boldsymbol{x},\boldsymbol{z};\boldsymbol{q}_2) + \left(\beta_{k,1} + \sum_{i\in\mathcal{I}} \beta_{k,i+1} x_i\right) \tag{5}$$

$$h(\boldsymbol{x},\boldsymbol{z};\boldsymbol{q}_2) = 0 \tag{6}$$

$$\boldsymbol{x} \in \mathcal{F}. \tag{7}$$

Since g_A is the objective function of a MIP, the corresponding constraints of the MIP are represented here through the function h of (6). These are formulated in detail in Section 3.3. The constraints of Section 3.3 consist of both equality and inequality constraints. They are summarized here as a set of equality constraints (they can equivalently be represented as a set of inequality constraints). Recall that the feasible region $\mathcal{F}$ of (7) is defined by Constraints (2)-(4).

The metamodel optimization Problem (5)-(7) differs from the simulation-based optimization Problem (1)-(4) in that: (i) it replaces the (unknown) simulation-based objective function (g of (1)) with an analytical function (m_k of (5)); (ii) it has additional constraints (Eq. (6)). The main feature that has allowed us in the past to design efficient algorithms for continuous SO problems is the formulation of a metamodel that embeds an analytical and problem-specific approximation of $g(\boldsymbol{x})$. This is the key component of the approach, yet this is also where the main methodological challenge lies because it is necessary to formulate an analytical model that: (i) provides a good approximation of the intricate function $g(\boldsymbol{x})$, which as will be discussed in Section 3.3 is particularly difficult for this car-sharing context, (ii) is scalable (i.e., is suitable to address high-dimensional instances), and (iii) is computationally efficient. The latter is critical because the metamodel optimization problem is solved at *every* iteration of the SO algorithm. Hence, it should be sufficiently efficient to warrant the allocation of computing resources to solving it rather than to run the simulator (i.e., simulating new points or increasing the accuracy of the estimates of simulated points).

The metamodel (m_k of (5)) is a parametric function with parameter vector $\boldsymbol{\beta}_k$. The latter is fitted, at every iteration of the SO algorithm, by solving a problem that minimizes a least-squares distance between metamodel predictions and simulation observations. For more details, see Problem (14) in Appendix A. As discussed above, the main challenge in this approach is the formulation of a computationally efficient and scalable problem-specific approximation of g, denoted here g_A. Let us now present the proposed formulation.

Figure 1 summarizes the proposed approach. We use a car-sharing network simulator (Fields et al., 2017), which relies on few demand modeling assumptions. Instead, it relies primarily on sampling from historical disaggregate car-sharing reservation data (which represents potentially censored or truncated demand) to estimate disaggregate latent (i.e., uncensored and untruncated) demand. For a detailed description of this demand estimation methodology, see Fields et al. (2021). For a given latent demand realization and a given supply solution (i.e., a given spatial allocation of the vehicle fleet), the simulator stochastically evaluates the performance of the solution, this yields a set of disaggregate reservations (which may have been censored). Section 3.5 provides a more detailed description of the simulator. Every time new supply solutions are evaluated via simulation, the metamodel is updated (i.e., its parameters are fitted by using all available simulation observations) and then used to solve an analytical optimization problem. More specifically, the metamodel

optimization problem is a mixed-integer program (MIP). While the analytical metamodel (i.e., the MIP) provides an aggregate description of demand and of demand-supply interactions, the simulator operates based on a disaggregate representation of demand (i.e., individual desired reservations) and of demand-supply interactions (i.e., individual realized reservations). Because the simulator uses the demand data in disaggregate form, we refer to it as a data-driven DSO algorithm.

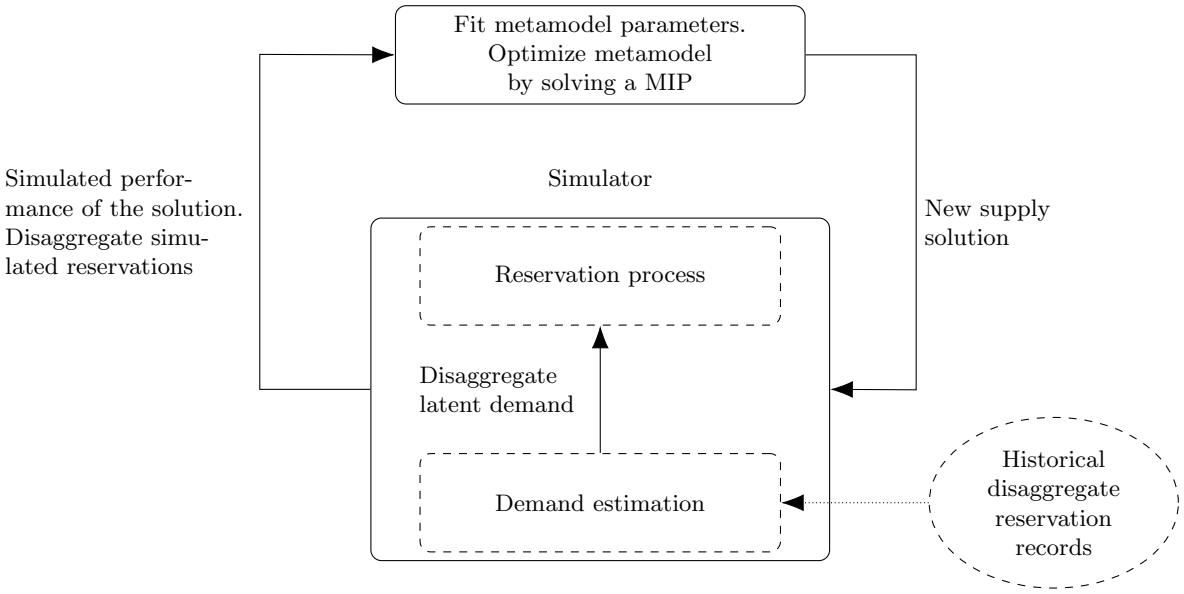


Figure 1: Data-driven metamodel DSO framework

3.3 Car-sharing fleet allocation metamodel formulation

To formulate the analytical problem-specific component of the metamodel, g_A, which approximates the profit of a given fleet allocation strategy, we introduce the following additional notation.

d^i_{tl}	number of customers that desire a reservation at station i with start time t and duration l;
r_{tl}	revenue from a reservation with start time t and duration l;
p^{ij}	discount to the revenue if a reservation is desired for station i but is fulfilled at (i.e., is made at) station j;
z^i_{tl}	number of customers that make a reservation at station i with start time t and duration l;
z^{ij}_{tl}	number of customers that desire to make a reservation at station i with start time t and duration l but make an adjusted reservation at station j with start time t and duration l;
$t_{\max}$	number of one-hour reservation start time intervals during the planning period (e.g., for an n-day planning period, $t_{\max} = n \times 24$);
$l_{\max}$	maximum reservation duration;
$\mathcal{I}_i$	set of stations "near" station i, including station i;
$\mathcal{L}$	set of reservation durations (in hours), $\mathcal{L} = \{1, 2, \ldots, l_{\max}\}$;
$\mathcal{T}$	set of reservation start time interval indices, $\mathcal{T} = \{1, 2, \ldots, t_{\max}\}$;
$\mathcal{T}_1(t, l)$	set of reservation start times for reservations with duration l that are ongoing at time t (i.e., they start prior to t and have not finished at time t).

The vector $\boldsymbol{z}$ defined in Section 3.2 consists of all variables $\{z^i_{tl}\}$ and $\{z^{ij}_{tl}\}$. The function g_A is formulated as:

$$g_A(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{q}_2) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}_i} \sum_{t\in\mathcal{T}} \sum_{l\in\mathcal{L}} p^{ij} r_{tl} z^{ij}_{tl} - \sum_{i\in\mathcal{I}} c_i x_i. \tag{8}$$

This function is defined as the difference between the total revenue and the total cost. Note that in the total revenue expression, we give a discount (p^{ij}) for reservations that are adjusted (i.e., the initial desired reservation was not feasible because a car was not available). This allows us to account for the impact on revenue of demand spillover (i.e., demand censoring). Note that demand spillover and loss are described in a more detailed and disaggregate manner in the simulator (see. Section 3.5).

The auxiliary variable z^{ij}_{tl} is related to the decision vector $\boldsymbol{x}$ through the analytical network model, which is denoted by h in Equation (6) and is defined as follows.

$$\sum_{j\in\mathcal{I}_i} z^{ji}_{tl} = z^i_{tl} \qquad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \tag{9}$$

$$\sum_{j\in\mathcal{I}_i} z^{ij}_{tl} \leq d^i_{tl} \qquad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \tag{10}$$

$$\sum_{l\in\mathcal{L}} z^i_{tl} + \sum_{l\in\mathcal{L}} \sum_{t'\in\mathcal{T}_1(t,l)} z^i_{t'l} \leq x_i \qquad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \tag{11}$$

$$z^i_{tl} \in \mathbb{R}_+ \qquad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \tag{12}$$

$$z^{ij}_{tl} \in \mathbb{R}_+ \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{I}_i, \forall t \in \mathcal{T}, \forall l \in \mathcal{L}, \tag{13}$$

where $\mathcal{T}_1(t,l) = \{t' \in \mathcal{T} : t'+1 \leq t \leq t'+l-1\}$. Equation (9) states that z^i_{tl}, the number of reservations at station i with start time t and duration l, is the sum of all desired reservations at station j (with start time t and duration l) that were shifted to station i. Note that $i \in \mathcal{I}_i$, hence this summation includes the reservations that were desired and also made at station i (with start time t and duration l). Equation (10) is a demand constraint. The right-hand side is the total demand for station i with start time t and duration l. The left hand side considers the set of reservations with a preference for station i start time t and duration l. This summation includes reservations where: (i) the preference was available and was made, (ii) the preference was not available and the reservation was adjusted and made at a neighboring station j (with the same start time t and the same duration l). For a given fleet assignment, the difference between the right-hand side and the left-hand side represents the lost demand for reservations at station i with start time t and duration l. The left-side of the Constraint (11) consists of two terms. The first term represents the total number of reservations at station i that start at time t. The second term represents the total number of reservations at station i that have started prior to time t and are still ongoing. Hence, Constraint (11) ensures that at station i and time t, the number of reserved cars (left-side of the inequality) is bounded above by the number of cars assigned to station i. Constraints (12) and (13) assume non-negative real values for the auxiliary variables (z^i_{tl} and z^{ij}_{tl}). The use of real-valued auxiliary variables, rather than integer variables, contributes to the computational efficiency of this analytical approximation. In this model, the exogenous parameters are d^i_{tl}, r_{tl}, c_i, p^{ij}, $t_{\max}$ and $l_{\max}$. Together they form the exogenous parameter vector $\boldsymbol{q}_2$. The endogenous variables are z^i_{tl}, z^{ji}_{tl} and x_i. The exogenous parameters r_{tl}, c_i, p^{ij} and $l_{\max}$ are directly estimated from the data and in consultation with Zipcar staff. Note that we have $0 \leq p^{ij} \leq 1$ and $p^{ii} = 1$ for all $i \in \mathcal{I}$ and $j \in \mathcal{I}_i$. A discussion on the simplifications of this analytical model compared to the simulator is given in Section 3.5.

The demand parameters (d^i_{tl}) are estimated by sampling from the historical reservation data to generate a set of disaggregate desired reservations (i.e., latent demand), which are then aggregated

to estimate the parameters (d^i_{tl}). A description of the demand sampling step is given in Section 3.5. Unlike the analytical MIPs (metamodel MIP or SP), the simulator uses the disaggregate demand realizations without aggregation. Since this data sampling is stochastic, the case studies of Section 4 consider various experiments based on different latent demand realizations. This serves to evaluate the impact of varying demand on the performance of the proposed method. Similarly, a set of latent demand realizations is used for the stochastic program (SP) benchmark of Section 4.4.

For any station $i \in \mathcal{I}$, if we assume the maximum number of stations in the neighborhood of i, $\mathcal{I}_i$, is smaller than a constant W, i.e., $|\mathcal{I}_i| \leq W$, then the number of auxiliary variables of the metamodel is in the order of $O\left(W|\mathcal{I}||\mathcal{T}||\mathcal{L}|\right)$, and the number of constraints is in the order of $O\left(|\mathcal{I}||\mathcal{T}||\mathcal{L}|\right)$. Hence, by bounding the duration of the planning period and the maximum duration of a reservation, the number of variables and the number of constraints increase linearly with the number of stations. This contributes to the scalability of the model. In summary, the metamodel optimization problem is a mixed-integer linear model, which can be solved in a computationally efficient way with a variety of standard solvers. For the case studies of Section 4.2, the MIP (5)-(7) is solved on average in 2.4 seconds for the the Boston South End network and in 35.1 seconds for the large-scale Boston network of Section 4.3.

3.4 DSO algorithm: MetaAHA

The proposed metamodel is embedded within a general-purpose DSO algorithm. We have chosen the Adaptive Hyperbox Algorithm (AHA) of Xu et al. (2013). As discussed in Section 2.2, AHA has been used to solve high-dimensional problems with a decision vector of dimension 100. AHA is a locally convergent random search algorithm. We use the term *current iterate*, denoted $\boldsymbol{x}_k$, to refer to the point considered to have best performance at iteration k. The name AHA stems from the sampling, at every iteration, from a region which is the intersection of the feasible region and a *hyperbox*. The latter is centered at the current iterate with a size that is updated based on the performance of the current iterate and of its neighbors. For more details, see Appendix A. Let $\mathcal{H}_k$ denote the hyperbox at iteration k. The proposed algorithm, denoted MetaAHA, is an extension of AHA. Algorithm 1 presents MetaAHA.

Each iteration k of the algorithm consists of 4 main steps. Step 1 identifies the set of points to simulate. These can be new points that have not been simulated before or points that have already been simulated and for which we will run additional simulation replications. Step 2 simulates these points. Step 3 checks whether termination criteria are satisfied. Step 4 uses the set of all simulation observations collected so far and updates the fit of the metamodel. Additional algorithmic details and a flowchart summary of MetaAHA are given in Appendix A.

Algorithm AHA is obtained from MetaAHA by omitting Steps 1b, 1c , 3b and 4; and setting r (of Step 1a) to n (while for MetaAHA $r = n-2$). Steps 1b and 1c solve mixed-integer programs. These steps yield solutions to MIPs. Hence, they exploit problem-specific analytical structural information provided by the analytical fleet allocation model. This information enables the algorithm to: (i) identify points with good performance within few, or even no, simulation runs because the analytical fleet allocation model can be solved without available simulation observations, and (ii) become less sensitive to the quality of the initial sample. This sensitivity to the quality of the initial sample has been identified and discussed in past AHA work (Xu et al., 2013). While both Steps 1b and 1c exploit this problem-specific analytical information, Step 1c does so within the hyperbox, leading to the identification of local points with good performance, while Step 1b does so in the entire feasible region, leading to the identification of global points with good performance. In Step 2, we

Algorithm 1 MetaAHA

Initialization:

– Initialize parameters: iteration index $k = 1$, hyperbox $\mathcal{H}_1 = \{\boldsymbol{x} : 0 \leq x_i \leq N_i, \forall i \in \mathcal{I}\}$, set n (the number of solutions to simulate per iteration). Set the number of randomly sampled solution in each iteration $r = n - 2$. For the metamodel parameter vector $\boldsymbol{\beta_1}$, set $\beta_{1,0} = 1$ and $\beta_{1,i} = 0$ $(\forall i \geq 1)$.

Step 1: identify the set of n points to sample

– Step 1a: obtain r points in $\mathcal{F} \cap \mathcal{H}_k$ based on the asymptotically uniform sampling mechanism of AHA.

– Step 1b: obtain 1 point, denoted $\boldsymbol{x}_k^{\text{meta}}$, as the solution to the metamodel optimization Problem (5)-(7).

– Step 1c: obtain 1 point, denoted $\boldsymbol{x}_k^{\text{meta-hyper}}$, as the solution to the metamodel optimization Problem (5)-(7) with the additional constraint that the point belongs to $\mathcal{H}_k$.

Step 2: simulation

– Following the procedure of AHA: simulate the points identified in Step 1; simulate $\boldsymbol{x}_{k-1}$ (for $k > 1$); select the point with best performance $\boldsymbol{x}_k$ (i.e., update the current iterate); update the hyperbox.

Step 3: check for algorithm termination

– Step 3a: test if $\boldsymbol{x}_k$ is a local optimum following the procedure of AHA. If so, stop.

– Step 3b: if the total number of iterations exceeds the maximum number of iterations (i.e., if the computational budget is depleted), stop.

Step 4: metamodel update

– Step 4a: for any simulated point $\boldsymbol{x}$ that has not been evaluated by the analytical network model, evaluate it (i.e., for a given $\boldsymbol{x}$, maximize $g_A(\boldsymbol{x}, \boldsymbol{z})$ of Equation (8) over $\boldsymbol{z}$ subject to Constraints (9)-(13)).

– Step 4b: use all simulation observations collected so far to fit the metamodel parameter $\boldsymbol{\beta}_k$ (i.e., solve the least squares Problem (14) defined in Appendix A).

Step 5: update iteration counter

– Set $k = k + 1$, proceed to Step 1.

determine the number of replications to simulate for each point (this is done based on the approach of AHA, which is also described in this paper in Appendix A), we simulate the points and then update both the hyperbox and the current iterate. In Step 3b, if the computational budget is depleted, then the algorithm is terminated without convergence. This serves to reflect the most common way in which these algorithms are used in practice.

Note that MetaAHA does not change the main building blocks of the basic algorithm AHA. It merely complements it by adding a problem-specific sampling strategy which is based on the use of the metamodel. Hence, AHA's asymptotic local optimality guarantee is preserved. MetaAHA illustrates how a variety of general-purpose DSO algorithms can be complemented with such problem-specific sampling strategies to improve their robustness to the quality of the initial points as well as their short-term (i.e., small sample) performance. For practitioners, who typically use these algorithms under tight computational budgets, this has the potential to improve the performance of these general-purpose algorithms.

3.5 Two-way car-sharing data and simulator

We summarize here the main ideas underlying the simulator. For more details on the specification of the simulator as well as on its validation, see Fields et al. (2017). The simulator takes as input disaggregate historical reservation data, estimated daily demand per station (i.e., total daily number of reservations desired per station), a fleet assignment strategy, and yields as output a set of realized reservations (reservations actually made) with the corresponding network-wide profit.

More specifically, the simulation process consists of two main parts, as summarized in Figure 1. The first step, referred to as the demand sampling step, samples from the data such as to (approximately) obtain a set of desired reservation requests (i.e., reservations that users would ideally desire to make). These reservations can be thought of as realizations of latent demand. Hence, we distinguish between realized demand (an empirical distribution of which is given by the dataset) and latent demand. The second step, referred to as the reservation simulation step, considers a given latent demand (i.e., a given set of desired reservations) and simulates the reservation process as follows. It ranks, and then sequentially processes, the desired reservations by increasing creation time. For a given reservation, if a car is available (at the desired station and during the desired time interval), then the reservation is made. Otherwise, with a given probability the client will either not make a reservation (this is referred to as lost demand) or they will consider an "adjacent" reservation, which is either at a nearby station or at a nearby start time (this is referred to as demand spillover; it accounts for demand censoring). The probability depends on the distance between the initially desired reservation and the considered adjacent reservation. Once a given reservation is made, other users cannot use the same car at any time during this reservation period. This procedure mimics the first-come-first-reserve process.

The most important input to the simulator is the set of historical disaggregate reservation data. In this work, we use Zipcar data. For each reservation observation in the dataset, the following attributes are used: station (this is both the pick-up and the drop-off location), start time, duration and reservation creation time (i.e., the timestamp of when the reservation was made). Additionally, based on information available online we have estimated reservation revenues. The time resolution of the simulator is based on that of the data which is 30 minutes. This means that reservation duration and reservation start times are defined in 30 minute increments.

A main feature of this simulator is that this reservation process simulation is based on a handful of

parameters, which are estimated from the data. Additionally, there are few modeling assumptions, which were made in consultation with Zipcar staff. They include the probability of considering an adjacent reservation and the formulation of a distance metric between reservations. Each of the two steps of the simulation process described above (i.e., demand sampling and reservation simulation) are stochastic. In other words, the generation of a set of desired reservation requests is stochastic and the mapping of a desired reservation to a realized (or even a lost) reservation is also stochastic.

We now present the main simplifications of the analytical metamodel (i.e., of the MIP) compared to the simulator. These simplifications also hold for the stochastic programming model that is used as a benchmark method in the case study of Section 4.4. These simplifications contribute to the formulation of efficient MIP models. First, the analytical model does not enforce the first-come-first-reserve rule of the simulator. In other words, for a given set of reservation requests, they will not be processed by increasing order of reservation creation time. Instead, the set of reservations that leads to highest (metamodel) profit will be realized regardless of their respective creation times. Second, the analytical model allows for reservations to be adjusted in space (i.e., change of station) but not in time (i.e., the start time of a desired reservation cannot change). Third, the adjustment process is simplified. For a given reservation, the simulator checks whether it is available, and if not with a certain probability it considers to either not make any reservation (leading to lost demand) or to attempt a nearby (in space and time) reservation (leading to demand spillover). The simulator iterates on these steps (i.e., a given client may attempt to make several reservations before deciding on a final reservation or before deciding not to make a reservation). In the analytical model, there is no sequential reservation process. Instead, demand spillover is approximated through the discounted revenue parameter, p^{ij} of Eq (8). Fourth, the simulator considers a time resolution of 30 minutes (i.e., reservation start times and durations are defined in 30 minute increments), while the analytical model considers a time resolution of 1 hour.

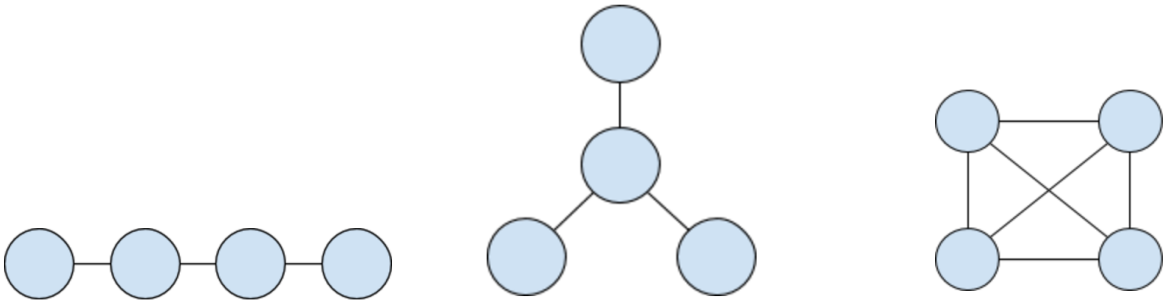
4 Case studies

In this section, we apply MetaAHA to optimize the design of two-way car-sharing systems. Section 4.1 considers a low-dimensional problem with synthetic toy networks. Sections 4.2-4.4 consider high-dimensional problems for Zipcar's Boston market. We study its two-way services for two Boston areas: (i) an area of downtown Boston known as South End (Section 4.2) and (ii) a larger network that includes 23 zipcodes of the Boston metropolitan area (they include Allston, Arlington, Boston, Brighton, Brookline, Cambridge, Charlestown, Chelsea, Medford and Somerville) (Section 4.3 and 4.4). All experiments are conducted on a machine with 125GB RAM with an Intel Xeon E5-2630 v3 processor.

4.1 Synthetic toy networks

The goal of these low-dimensional synthetic experiments is to evaluate the quality of the analytical approximation (g_A of Equation (8), which is provided by the analytical network model) of the simulation-based objective function (g of Equation (1)). We consider 3 networks with topologies that are simple and are representative of subnetwork topologies of Zipcar's Boston network. The 3 networks are displayed in Figure 2. Each circle represents a car-sharing station. Each network has four stations. Recall from Section 3.3 that, in the analytical model, when the desired reservation of a user is not available, he/she may consider a substitute chosen from a set of neighbors that are defined as spatially nearby locations (this set was denoted $\mathcal{I}_i$ in Section 3.3). In other words, these

are stations where the demand can spillover. In Figure 2, for a given station i, its set of neighbors or substitute stations, $\mathcal{I}_i$, is the set of stations that are connected with an edge to station i. Hence, the three network topologies of Figures 2a, 2b and 2c consider, respectively, a loosely connected network of stations, where no stations share any neighbors; a centralized network, where all stations have the center station as a neighbor; and a fully connected network, where all stations are neighbors with all other stations. Each station has a capacity of 6 vehicles (i.e., N_i of Equation (3) equals 6), the fleet size is unlimited (i.e., X of Equation (2) takes any value such that $X \geq 24$). Hence, the feasible region is $\{x \in [0,6]^4 \cap \mathbb{Z}^4\}$, which contains 2401 feasible solutions. The data used for simulation is the Zipcar reservation data related to such a subnetwork. The planning period is an 8-day period in July 2014 (July 10 to July 17).

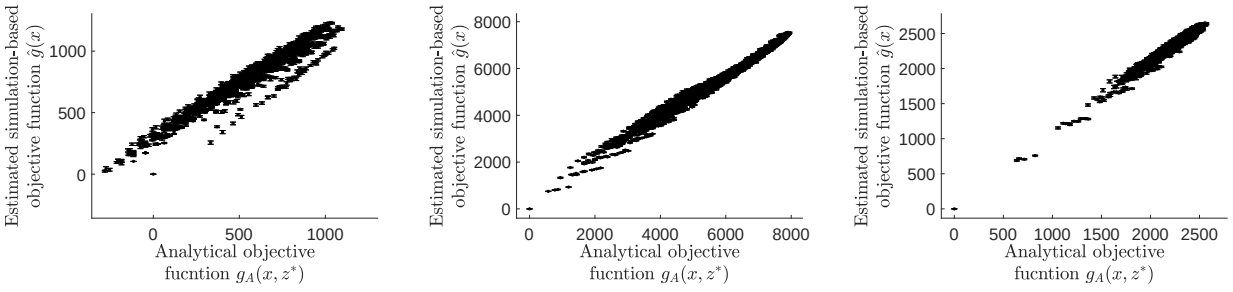


(a) Toy network 1 (b) Toy network 2 (c) Toy network 3

Figure 2: Toy network topologies

For each network, we generate a group of 10 demand scenarios. A demand scenario consists of the desired reservations generated through the demand sampling step described in Section 3.5. The use of various demand scenarios serves to account for demand stochasticity. For a given point, $\boldsymbol{x}$, one simulation replication (i.e., one simulation-based realization of its performance) is defined as the average simulated performance over the 10 demand scenarios. For a given point, $\boldsymbol{x}$, the final estimate of its simulation-based performance, $\hat{g}(\boldsymbol{x})$, is obtained as the average over 50 simulation replications. For the analytical model, we generate a different demand scenario to estimate its exogenous parameters (d^i_{tl} of Equation (10)). For a given point $\boldsymbol{x} \in \mathcal{F}$, the analytical objective function, $g_A(\boldsymbol{x}, \boldsymbol{z}^*)$, is obtained by maximizing Equation (8) over $\boldsymbol{z}$ subject to Constraints (9)-(13).

Each plot of Figure 3 considers one network and displays the analytical objective function, $g_A(\boldsymbol{x}, \boldsymbol{z}^*)$, along the x-axis and the estimated simulation-based objective function, $\hat{g}(\boldsymbol{x})$, with a corresponding 95% interval along the y-axis. The confidence intervals are barely visible. Each plot displays the 2401 feasible solutions.



(a) Toy network 1 (b) Toy network 2 (c) Toy network 3

Figure 3: Comparison of the analytical objective function value with the estimated simulation-based objective function value for toy networks

For all three plots, there is a positive linear correlation between the analytical approximations, $g_A(\boldsymbol{x}, \boldsymbol{z}^*)$, and the simulation-based estimates, $\hat{g}(\boldsymbol{x})$. This indicates that for all three representative network topologies the analytical network model provides a good approximation of the simulation-based objective function.

4.2 Boston South End network

We now consider the South End neighborhood in downtown Boston. A map of the area is displayed in Figure 4. The 23 stations over which we optimize are displayed with red circles. The planning period is July 10-17, 2014. During this period the average fleet size is 101 cars (i.e., $X = 101$). Based on consultation with Zipcar, we set the station capacity, N_i, to 16. We compare the performance of MetaAHA and AHA. This comparison serves to evaluate the added value of complementing AHA with information from the analytical problem-specific network model. The maximum number of algorithm iterations, K, is set to 40. At every iteration, the number of points to be simulated is set to 10 (i.e., $n = 10$).

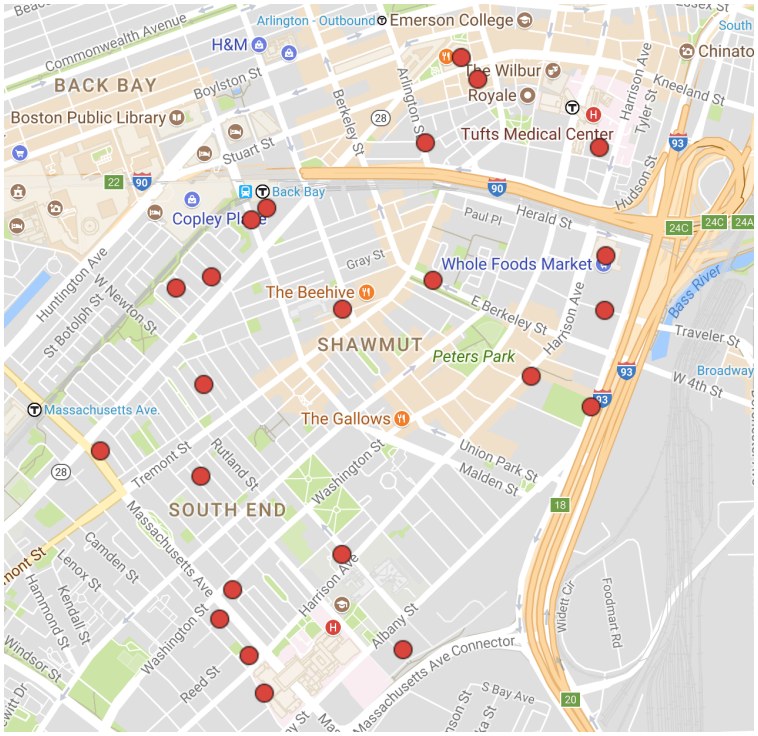
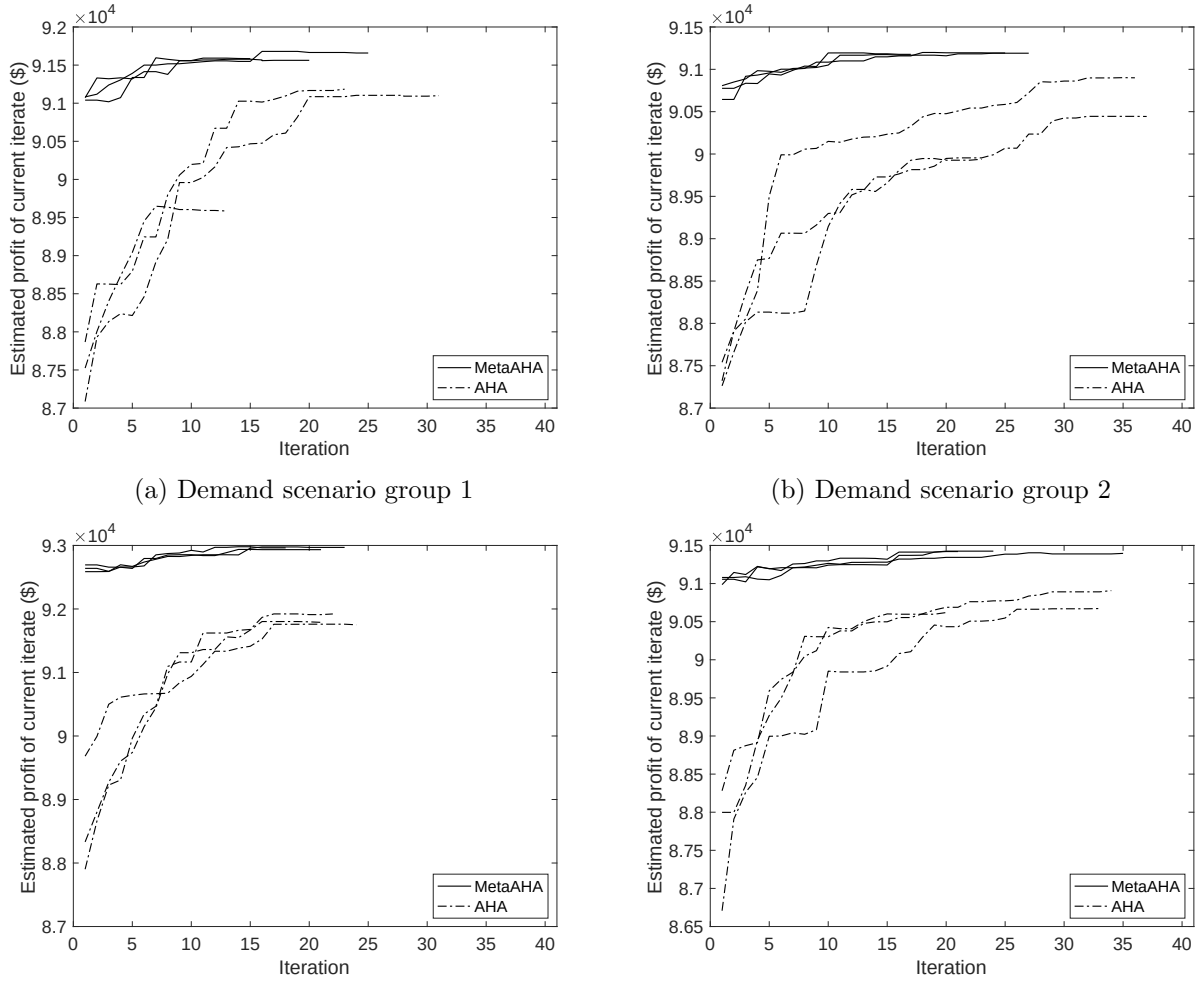


Figure 4: Zipcar stations in Boston South End neighborhood (map data: Google Maps (2017b))

To account for the stochasticity of demand, we proceed as in Section 4.1. We consider a group of 10 demand scenarios. For a given point, $\boldsymbol{x}$, one simulation replication (i.e., one simulation-based realization of its performance) is defined as the average simulated performance over the 10 demand scenarios. Figure 5 contains four plots. Each plot considers a different group of 10 demand scenarios. As in Section 4.1 for each group of demand scenarios, one additional demand scenario is used to estimate the exogenous parameters of the analytical network model.

Each plot displays the iteration index along the x-axis and the performance estimate of the current iterate (i.e., simulation-based estimate of the objective function of the best point) along the y-axis. The range of the y-axis differs across the plots. Each plot illustrates, for a given demand scenario group, the difference in performance of the two methods. Each plot displays 6 lines:



(a) Demand scenario group 1

(b) Demand scenario group 2

(c) Demand scenario group 3

(d) Demand scenario group 4

Figure 5: MetaAHA vs. AHAInit: objective function estimate of the current iterate across iterations

3 solid (resp. dashed) lines that represent 3 MetaAHA (resp. AHA) runs. For all plots, we observe the following main trends. First, MetaAHA identifies points with good performance from the first iteration, while the points initially sampled by AHA do not have good performance. Actually, for all six runs of MetaAHA, the best point identified in the first iteration corresponds to the solution of the analytical fleet allocation problem (i.e., maximize g_A of Equation (8) over both $\boldsymbol{x}$ and $\boldsymbol{z}$ subject to Constraints (2)-(4) and (9)-(13)). This shows the added value of the analytical structural information provided by g_A. Note that the initial points sampled by AHA are obtained from an asymptotically uniform sampling distribution for integral points from compact polyhedrons as defined in Hong and Nelson (2006). This general-purpose sampling method allows AHA to ensure asymptotic convergence properties, yet since it lacks problem-specific information, it is not designed to provide good quality initial solutions. Second, as the iterations advance, AHA identifies points with improved performance. This is consistent with the experiments and observations in Xu et al. (2013), which show that AHA is an efficient algorithm for a broad class of DSO problems. Nonetheless, it is outperformed throughout by MetaAHA. Third, MetaAHA shows a slight improvement across iterations, yet it is not as significant as that of AHA. Fourth, the performance of the final solution derived by MetaAHA (i.e., the current iterate at the final iteration) is similar across the 3 MetaAHA runs, while final solutions have higher variability in

performance for the 3 AHA runs. This indicates that MetaAHA is less sensitive to the stochasticity of the simulator. This may be attributed to the structural analytical information provided by the problem-specific fleet allocation model (g_A).

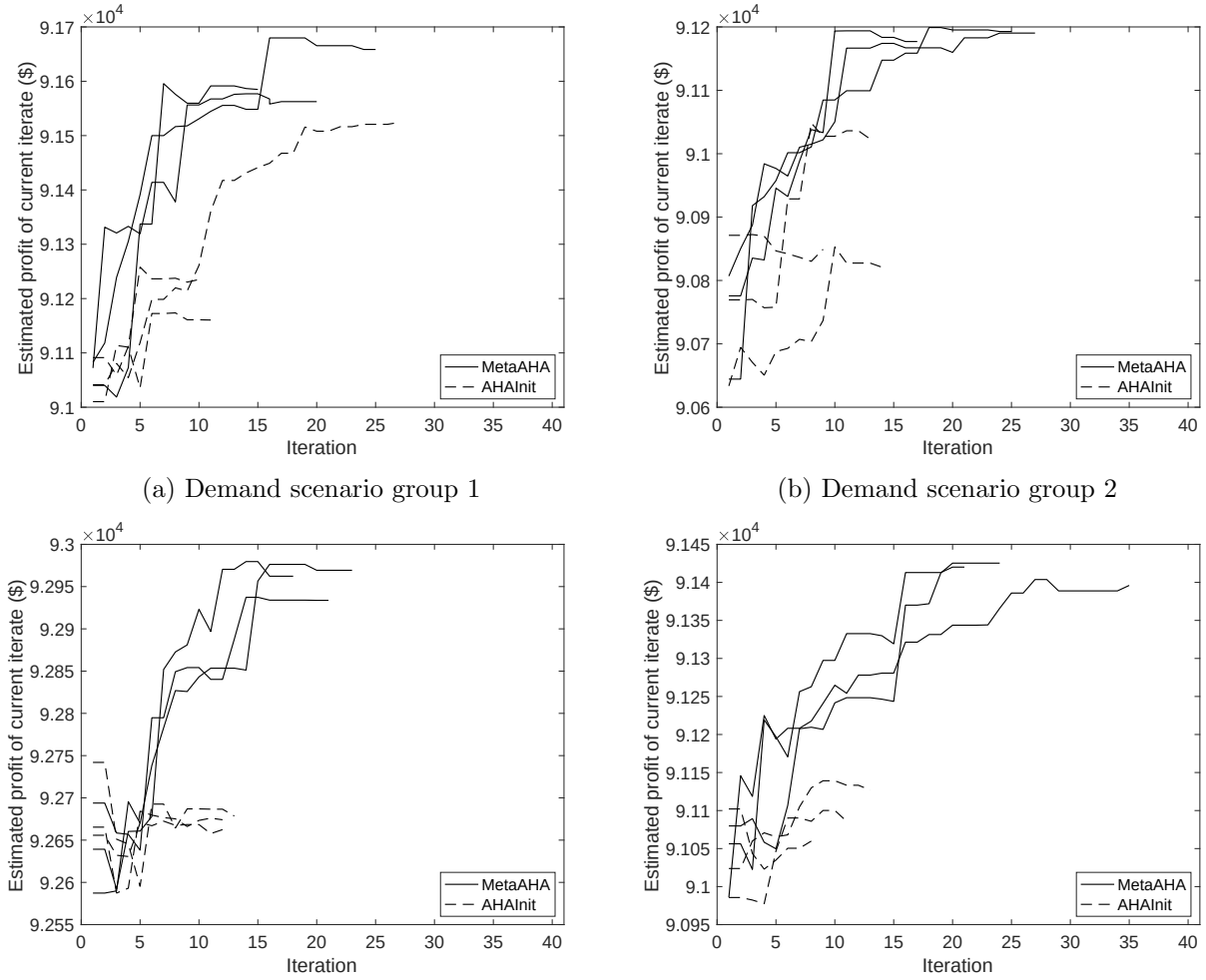
Note that in Figure 5 all lines terminate prior to iteration 40. This occurs if a current iterate is considered to be a local optimum (Step 3a of Algorithm 1). To limit the premature convergence of AHA, Xu et al. (2013) have combined it with the multi-start ISC framework (Xu et al., 2010). Also, most lines are not monotonically non-decreasing. This can occur when running additional simulation replications of the current iterate leads to a lower objective function estimate (which can itself lead to a change of the current iterate).

These results indicate the ability of the metamodel approach to: (i) improve the robustness of the algorithm to the quality of the initial points, (ii) identify good solutions within very few iterations, and (iii) lead to low variability across the performance of the derived final solutions. These are all trends that have been observed in our past metamodel work for continuous SO transportation problems (Zhang et al., 2017; Chong and Osorio, 2017; Chen et al., 2019; Osorio, 2019).

The results of Figure 5 indicate that a suitable approach would be to include in the initial sample of AHA the solution proposed by the analytical fleet allocation problem (i.e., the solution that maximizes g_A of Equation (8) over both $\boldsymbol{x}$ and $\boldsymbol{z}$ subject to Constraints (2)-(4) and (9)-(13)), and then to use the traditional AHA algorithm for all other iterations. Let AHAInit denote this approach. We now carry out a comparison of MetaAHA with AHAInit. This comparison serves to evaluate the added value of using analytical network model information across the iterations of AHA, rather than limiting the use of this analytical model to the first iteration. We use the same experimental design as for Figure 5. Figure 6 display four plots. Each plot considers a given group of 10 demand scenarios for the simulator and one demand scenario for the analytical model. The solid (respectively, dashed) lines represent MetaAHA (resp. AHAInit). The range of the y-axis differs across the plots.

The following trends are common to the four plots. First, MetaAHA outperforms AHAInit across all iterations. This reveals the added value of the metamodel m_k which combines the analytical fleet allocation information g_A with the simulation information. In other words, using the analytical fleet allocation model g_A to initialize a general-purpose algorithm contributes to its efficiency, yet there is even further added value of using the analytical information across iterations. Second, AHAInit tends to converge more quickly to a local optimum. Often, this local optimum has performance that is similar to that of the point obtained by solving the analytical fleet allocation problem (i.e., the point obtained by maximizing g_A subject to Constraints (2)-(4) and (9)-(13)).

For the 12 MetaAHA runs of Figure 6 (i.e., 3 runs for each of the 4 plots), there are a total of 87 instances where the current iterate is updated. Recall that for MetaAHA a current iterate can be of 3 types: (i) it can be a solution to the metamodel optimization problem solved in the entire feasible region (i.e., Step 1b of Algorithm 1, which yields points denoted $\boldsymbol{x}^{\text{meta}}$), (ii) it can be a solution to the metamodel optimization problem solved in the intersection of the entire feasible region and the hyperbox (i.e., Step 1c of Algorithm 1, which yields points denoted $\boldsymbol{x}^{\text{meta-hyper}}$), or (iii) it can be obtained from random sampling (i.e., Step 1a of Algorithm 1, which yields points denoted $\boldsymbol{x}^{\text{sampled}}$). Note that a point can be both of type $\boldsymbol{x}^{\text{meta}}$ and of type $\boldsymbol{x}^{\text{meta-hyper}}$. This occurs when the solution to the metamodel optimization problem in the entire feasible region is located in the hyperbox. Of the 87 different current iterates of the 12 MetaAHA runs in Figure 6, more than two thirds (i.e., 71.3% or 62 points) are of type $\boldsymbol{x}^{\text{meta}}$ or $\boldsymbol{x}^{\text{meta-hyper}}$, while less than one third (28.7% or 25 points) are of type $\boldsymbol{x}^{\text{sampled}}$. In other words, two thirds of the current iterates are obtained by using the structural information of the analytical network model. For the 12 final best



(a) Demand scenario group 1

(b) Demand scenario group 2

(c) Demand scenario group 3

(d) Demand scenario group 4

Figure 6: MetaAHA vs. AHAInit: objective function estimate of the current iterate across iterations

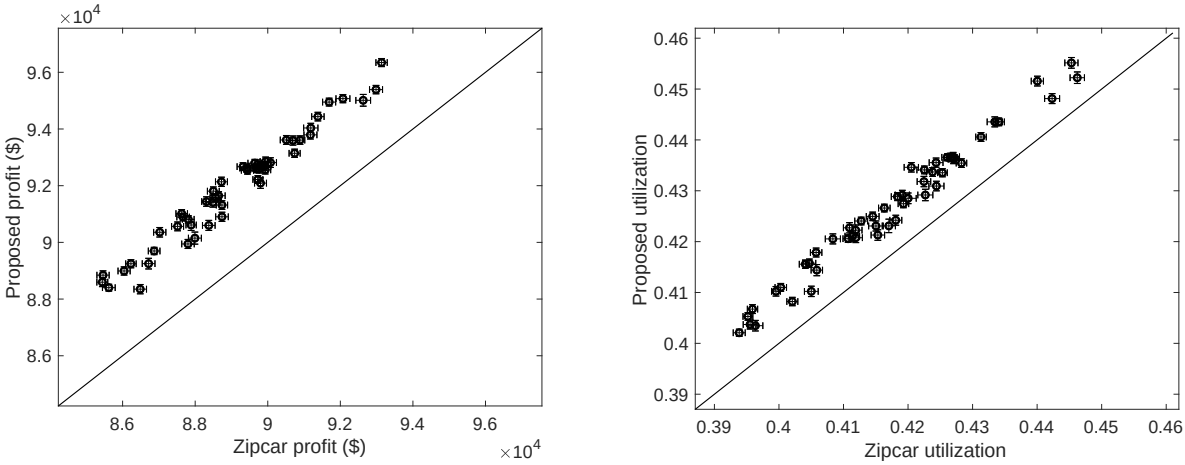
solutions returned by the MetaAHA runs, 9 of them are identified by solving the metamodel and 3 of them by random sampling. Moreover for the 12 runs of MetaAHA, we simulated 2364 points. Only 18.1% of the simulated points are obtained by solving the metamodel (429 points), while the remaining 81.9% are obtained by random sampling. Hence, even though the points derived by metamodel evaluations represent only 18.1% of the total set of sampled points, they lead to 75% of the final solutions and 71.3% of the current iterates. This highlights the added value of the structural information provided by the analytical MIP. Among the 62 current iterates obtained by using structural analytical information, 21 are of type $\boldsymbol{x}^{\text{meta}}$ and 47 are of type $\boldsymbol{x}^{\text{meta-hyper}}$ (note that 6 points are both of type $\boldsymbol{x}^{\text{meta}}$ and $\boldsymbol{x}^{\text{meta-hyper}}$). This shows that both the global (i.e., in the entire feasible region) and the local (i.e., in the hyperbox) information of the analytical network model help to identify points with improved performance. Recall that the metamodel is fitted after every iteration, hence the metamodel optimization problems solved across iterations differ and hence their solutions may differ. It is through this fitting process that the metamodel combines information from the simulator with information from the analytical network model. The high number of distinct current iterates identified by the metamodels illustrates the added value, across iterations, of combining the analytical information with the simulated information.

We created the following Table 2 to present the runtime of AHA, AHAInit, and MetaAHA for the Boston South End case. For each algorithm, we use all 3 runs of the 4 groups of inputs (hence 12 runs in total for each algorithm) to compute the following averages per algorithm run: the average number of iterations, the average total runtime, and the average total simulation runtime. For MetaAHA, we also compute the average total metamodel solution time. We can see that AHAInit has on average smaller number of iterations, which shows that it converges very quickly at a local optimum. Also in the Boston South End case, the metamodel solution time is low.

Table 2: Computational runtime comparison of AHA, AHAInit, and MetaAHA

Algorithm	Agerage number of iterations	Average total algorithm runtime (hr)	Average total simulation time (hr)	Average total metamodel solution time (hr)
AHA	25.41	0.90	0.88	N/A
AHAInit	11.75	0.36	0.35	N/A
MetaAHA	21.58	0.64	0.58	0.03

Figure 7 compares the performance of the best fleet assignment identified by MetaAHA (the proposed strategy) with that used by Zipcar during the planning period of interest. The final proposed (or "best") MetaAHA solution is defined as follows. We consider a set of 50 new demand scenarios. For all 12 solutions derived by MetaAHA (i.e., 3 algorithmic runs for each of the 4 plots of Figure 6), we estimate the average (over the 50 demand scenarios, each scenario is simulated with 50 replications) performance. The proposed solution is that with the best (i.e., largest) average performance.



(a) Profit (b) Utilization

Figure 7: Comparison of the Zipcar fleet assignment with the proposed assignment for the Boston South End network

Figure 7 displays two plots. The left plot compares the profit estimates of the two assignments. The right plot compares them according to vehicle utilization. Both of these metrics are important for Zipcar. For each plot, the x-axis considers the Zipcar assignment and the y-axis considers the MetaAHA proposed assignment. Each plot displays 50 points, which correspond to 50 demand scenarios. For each demand scenario, we estimate the performance based on 50 simulation replications. The performance estimate of each point is displayed along with a, barely visible, 95% confidence interval along each direction. Both the left and the right plots indicate that for all 50 demand scenarios the proposed plan yields improved performance, and this across all 50 demand scenarios. Compared to Zipcar's fleet assignment, the proposed solution yields an average improvement of profit of 3.2% and of vehicle utilization of 2.2%. Recall that these estimates are obtained

via simulation. Hence, they do not state that the proposed method outperforms the Zipcar method when deployed in the field.

4.3 Boston area network - comparison versus AHAInit

In this section, we consider a larger area of the Boston metropolitan area. This serves to evaluate the performance of MetaAHA for a high-dimensional problem. We consider a network of 315 stations distributed throughout 23 zipcodes that span over Allston, Arlington, Boston, Brighton, Brookline, Cambridge, Charlestown, Chelsea, Medford and Somerville. The map of Figure 8 displays the stations as red circles. We consider the same planning period as before. The station capacity, N_i, is set to 16, based on consultation with Zipcar. Historical data indicates that, during this planning period, there are an average of 894 cars assigned to these stations, i.e., $X = 894$. We proceed as before and consider a group of 10 demand scenarios. One additional demand scenario is used to estimate the exogenous parameters of the analytical model. We set the maximum number of iterations to 40 (i.e., $K = 40$) and the number of points to simulate per iteration to 70 (i.e., $n = 70$).

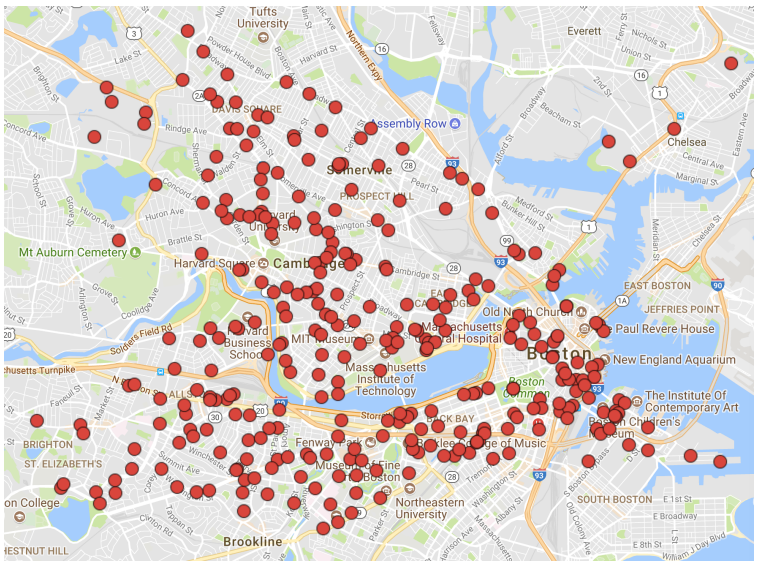


Figure 8: 315 Zipcar stations in Boston area (map data: Google Maps (2017a))

Figure 9 displays the results of 8 MetaAHA runs (solid lines) and 8 AHAInit runs (dashed lines). Only 3 of the 16 runs deplete the computational budget (i.e., they stop at iteration 40). They correspond to 3 MetaAHA runs. More specifically, the 8 MetaAHA runs stop at iterations 13, 14, 24, 33, 33, 40, 40 and 40. Those of AHAInit stop at iterations 14, 15, 15, 19, 20, 24, 33 and 38. All 8 runs of AHAInit yield final solutions with similar objective estimates. Seven out of the 8 MetaAHA final solutions are better than all 8 AHAInit final solutions.

In this work, the maximum number of DSO iterations is fixed a priori. This is standard practice in DSO and continuous SO applications that involve compute-costly simulators. However, it is worth studying the use of more appropriate stopping mechanisms that account for both the performance of the current iterate (i.e., the best point found so far) and the compute cost per iteration.

Figure 10 compares the performance of the best solution identified by MetaAHA with the fleet assignment strategy used by Zipcar. To evaluate the performance of a given fleet assignment strategy (that proposed by MetaAHA or that of Zipcar), we proceed as before. We generate

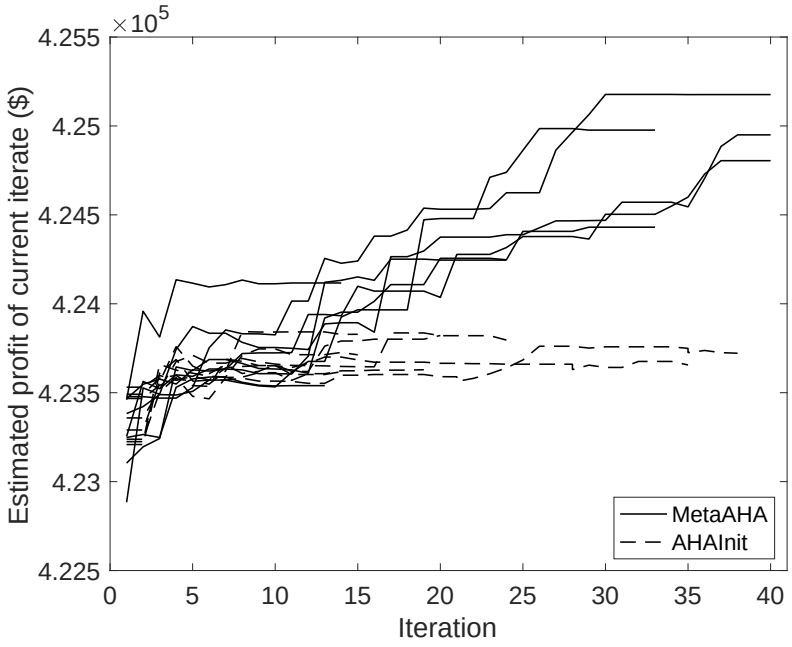
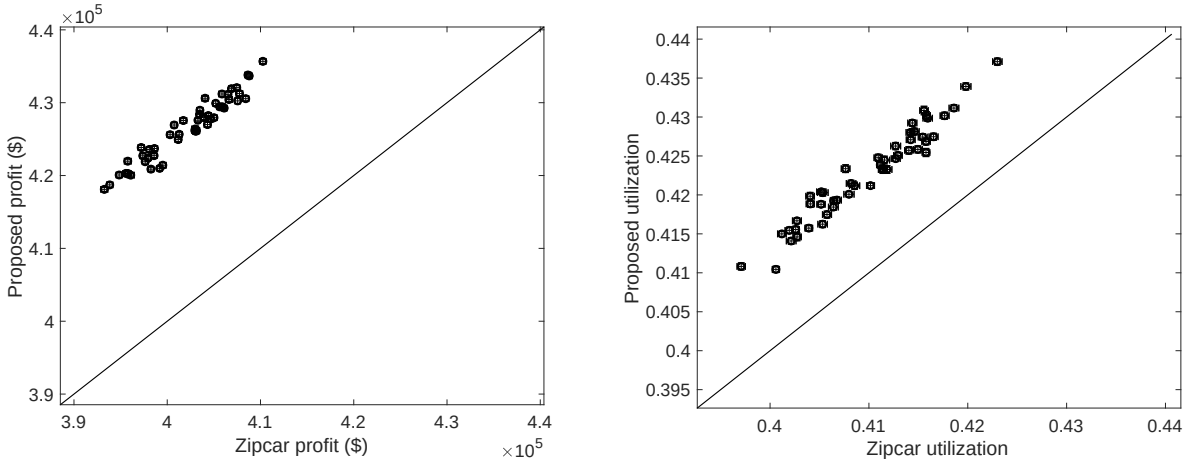


Figure 9: MetaAHA vs. AHAInit: objective function estimate of the current iterate across iterations

50 demand scenarios. For each of the 8 final solutions derived by MetaAHA and for each demand scenario, we run 50 simulation replications to estimate the average profit per solution. The solution with the highest average simulated profit is selected as the proposed solution. Figure 10 displays two plots: the left plot considers profit and the right plot considers vehicle utilization. For each plot, the x-axis considers the Zipcar assignment and the y-axis considers the proposed assignment. Each plot displays 50 points which correspond to the 50 demand scenarios. Each point estimate is displayed along with a 95% confidence interval along both directions. The confidence intervals, which are barely visible, are computed based on 50 replications. For both plots, the 50 points, which represent 50 different demand scenarios, are above the diagonal. Compared to Zipcar's fleet assignment, the proposed solution yields an average improvement of profit of 6% and of vehicle utilization of 3.1%. Moreover, for all 50 demand scenarios, the proposed strategy improves both the profit and the fleet utilization. Again, note that this comparison is based on simulated performance, which may not reflect field performance.

4.4 Boston area network - comparison versus stochastic programming

As mentioned in Section 1, the most common approach to study the fleet assignment problem is the use of analytical optimization methods, such as mathematical programs. In this section, we benchmark the performance of the proposed approach to stochastic programming (SP), which accounts for demand uncertainty. We consider a two-stage SP, where the second stage accounts for demand scenario realizations. The SP formulation is given in Appendix B. We use the same Boston area network as that of Section 4.3. We consider a set of 9 experiments with varying levels of demand and of cost. Demand is scaled by a factor: $\lambda \in \{1, 2, 3\}$, cost (i.e., term c_i of Eq.(1)) is scaled by a factor: $\theta \in \{1, 2, 3\}$. For each experiment (i.e., a given value of (λ, θ)), we proceed as follows. We consider one demand scenario to estimate the exogenous parameters of MetaAHA, and 3 additional demand scenarios for SP and for the simulator. Hence, the SP model and the



(a) Profit (b) Utilization

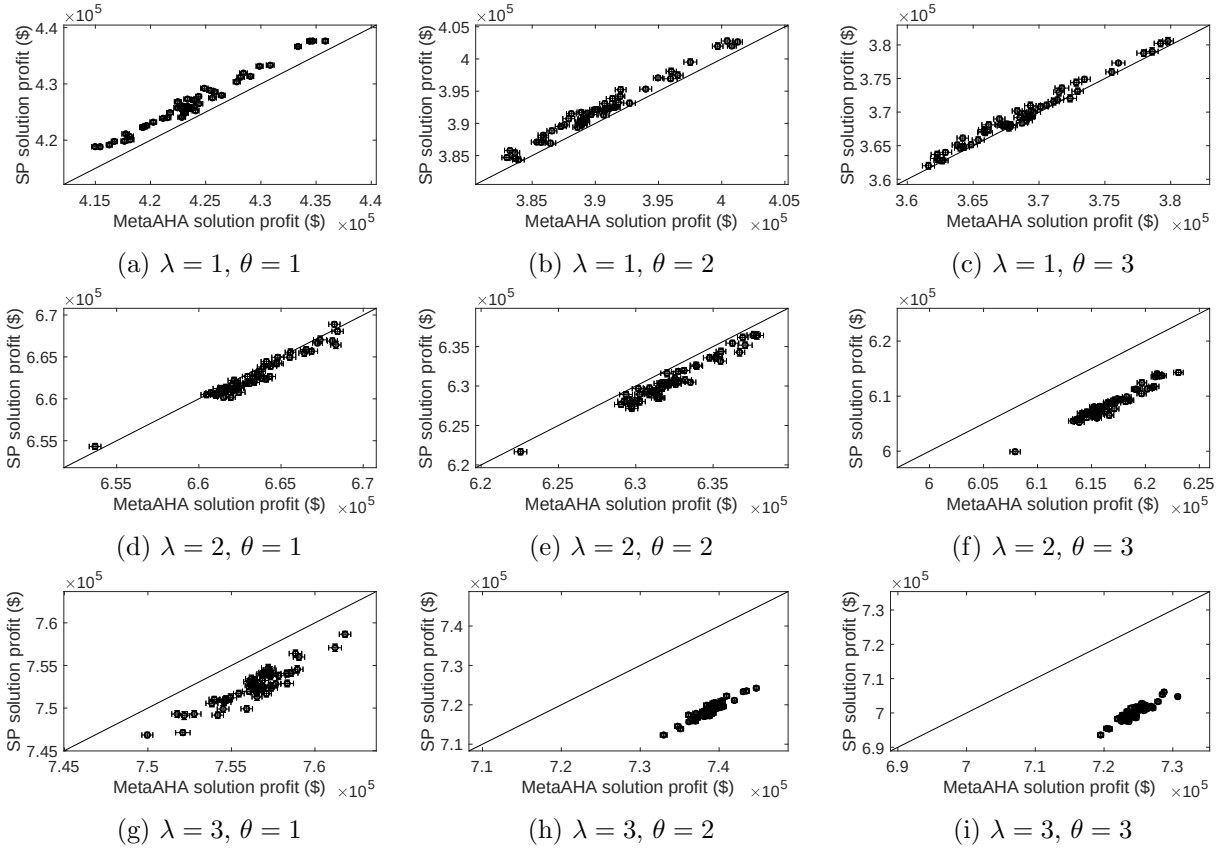
Figure 10: Comparison of the Zipcar fleet assignment with the proposed assignment for the Boston area network

simulator used as part of MetaAHA have the same demand input. When evaluating a solution via simulation (within MetaAHA), the simulated estimate of the objective function is obtained as the average over the 3 demand scenarios. We solve the SP model to obtain one SP solution. We run MetaAHA 3 times (in order to account for the stochasticity of both the simulator and the algorithm) to obtain 3 MetaAHA solutions. We use the same MetaAHA algorithmic parameters as in Section 4.3 (i.e., $K = 40$ and $n = 70$). For each final solution (SP or MetaAHA), we simulate its performance considering 50 demand scenarios and 50 simulation replications per demand scenario (for a total of 2500 simulations per solution). We compare the performance of the SP solution to that of the best MetaAHA solution, which is defined as that with the best simulated profit over the 2500 simulations.

Figure 11 displays 9 plots, one for each experiment. The demand scaling factor λ (resp. cost scaling factor θ) is constant across columns (resp. rows) and increases across rows (resp. columns). Each plot displays 50 points, which correspond to each of the 50 demand scenarios. The y-axis (resp. x-axis) displays the estimated, via simulation, mean profit over 50 replications using the SP (resp. best MetaAHA) solution. Each point has a 95% confidence interval along both coordinate directions. These intervals are barely visible. Each plot also displays the diagonal line defined by $y = x$.

For $\lambda = 1$ (i.e., top row of plots: Figures 11a, 11b, 11c), all 50 points are above the diagonal line. In other words, the SP solution outperforms that of MetaAHA. More specifically, the SP solution improves, on average, the profit by 0.65%, 0.43% and 0.21%, for $\theta = 1, 2$ and 3, respectively. This trend is reversed for the other 2 rows of plots. For $\lambda = 2$ (i.e., second row of plots), MetaAHA outperforms SP by an average of 0.1%, 0.26% and 1.36%, for $\theta = 1, 2$ and 3, respectively. For $\lambda = 3$ (bottom row of plots), MetaAHA outperforms SP, on average, by 0.51%, 2.80% and 3.48%, for $\theta = 1, 2$ and 3, respectively. For demand levels λ of 2 or 3 (i.e., second or third row of plots), as the cost level θ increases, so does the amount by which MetaAHA outperforms SP. For a given cost level (i.e., a given column of plots), as the demand level increases (i.e., from the top row to the bottom row), so does the amount by which MetaAHA outperforms SP.

Figure 12 considers the same 9 levels of demand and cost. It evaluates the ability of SP to approximate the simulation-based objective function. Each point of Figure 12 considers a given feasible



(a) $\lambda = 1$, $\theta = 1$ (b) $\lambda = 1$, $\theta = 2$ (c) $\lambda = 1$, $\theta = 3$

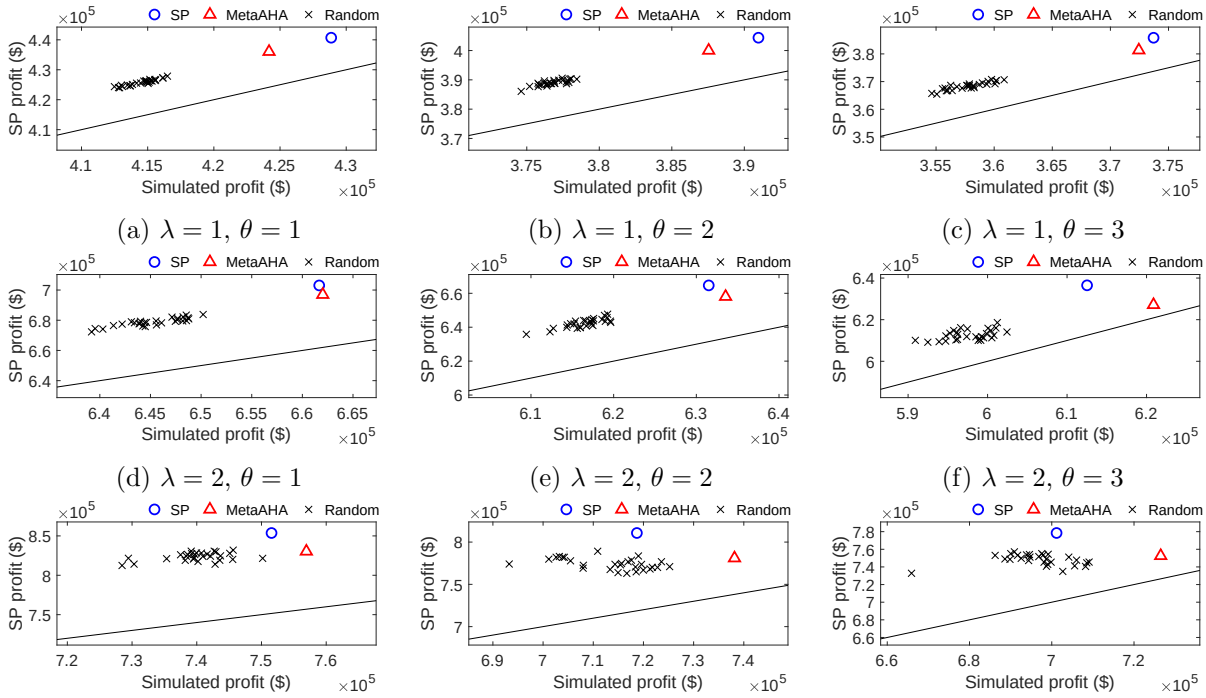
(d) $\lambda = 2$, $\theta = 1$ (e) $\lambda = 2$, $\theta = 2$ (f) $\lambda = 2$, $\theta = 3$

(g) $\lambda = 3$, $\theta = 1$ (h) $\lambda = 3$, $\theta = 2$ (i) $\lambda = 3$, $\theta = 3$

Figure 11: Comparison of the average profit, considering 50 demand scenarios, of the SP solution and of the best MetaAHA solution

solution and displays along the x-axis the simulated estimate of its objective function value and along the y-axis its SP objective function value. Each plot considers a set of the following 30 feasible solutions: the SP optimal solution (displayed as a blue circle), the best MetaAHA solution (displayed as a red triangle), and a set of 28 randomly sampled solutions (black crosses) that are in the neighborhood of the line connecting the SP optimal solution and the best MetaAHA solution (the sampling process is detailed in Appendix C). The diagonal line, $y = x$, is also displayed. For each solution, the SP objective function value and the simulation-based objective function estimate are based on the same 3 demand scenarios. These demand scenarios are the same as those used to derive the SP solution of the previous analysis. The simulation-based objective function estimate is obtained as the average across 50 replications of each of these 3 demand scenarios (i.e., each estimate involves $150 = 50 \times 3$ simulation evaluations). Figure 13 differs from Figure 12 in that the y-axis of each plot displays the metamodel objective function value. The value of the metamodel parameter $\boldsymbol{\beta}$ is that of the last iteration of the MetaAHA run which generated the best MetaAHA solution. Each of the 9 subplots of Figure 12 have the same axis limits as the corresponding subplot in Figure 13. Hence, the subplots are directly comparable across Figures 12 and 13.

For all plots of Figure 12, all points are above the diagonal line, i.e., SP tends to overestimate the simulated profit for all cost and demand levels. For $\lambda = 1$, the SP objective function exhibits a positive linear correlation with the simulated estimate. This also occurs for $(\lambda, \theta) \in \{(2, 1), (2, 2)\}$. Hence, the SP model correctly ranks the performance of the feasible solutions, and hence is an adequate tool for optimization. Nonetheless for high values of demand and of cost (i.e., all plots with $\lambda = 3$, as well as the plot $(\lambda, \theta) = (2, 3)$), there is no longer a positive linear correlation.

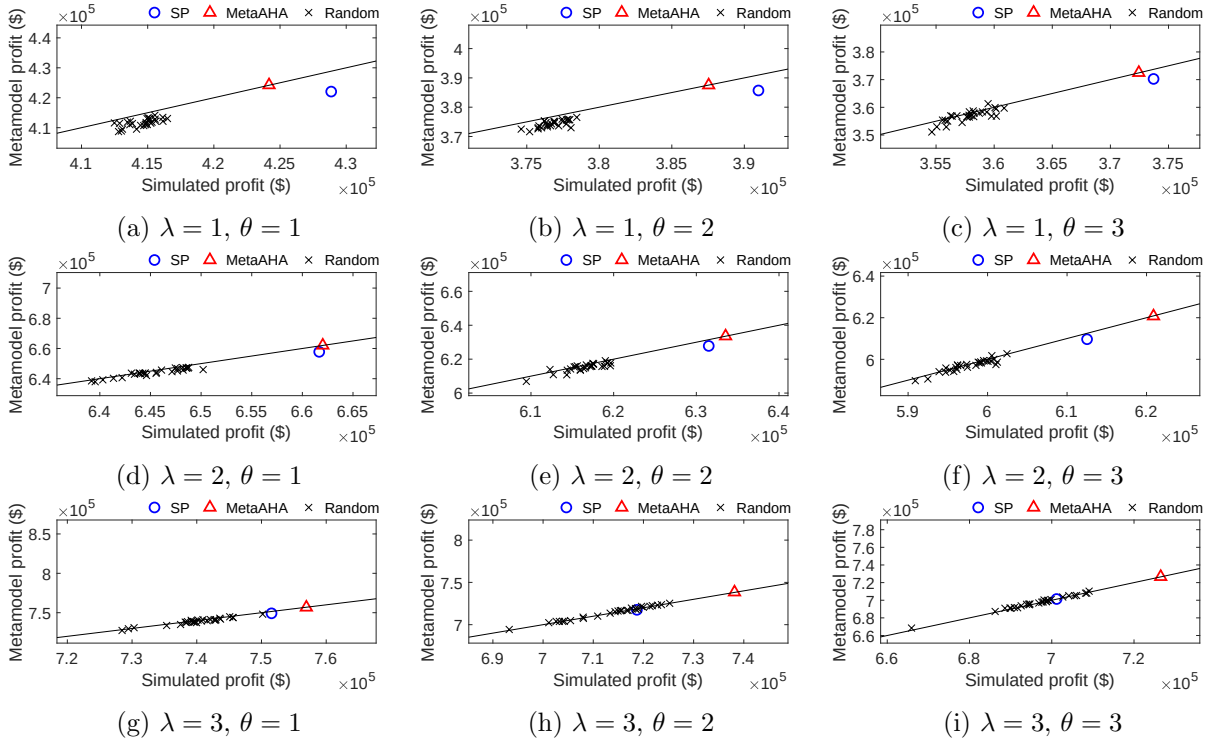


(a) $\lambda = 1, \theta = 1$ (b) $\lambda = 1, \theta = 2$ (c) $\lambda = 1, \theta = 3$

(d) $\lambda = 2, \theta = 1$ (e) $\lambda = 2, \theta = 2$ (f) $\lambda = 2, \theta = 3$

(g) $\lambda = 3, \theta = 1$ (h) $\lambda = 3, \theta = 2$ (i) $\lambda = 3, \theta = 3$

Figure 12: Comparison of the objective functions of the SP model and of the simulation model across various demand levels and cost levels.



(a) $\lambda = 1, \theta = 1$ (b) $\lambda = 1, \theta = 2$ (c) $\lambda = 1, \theta = 3$

(d) $\lambda = 2, \theta = 1$ (e) $\lambda = 2, \theta = 2$ (f) $\lambda = 2, \theta = 3$

(g) $\lambda = 3, \theta = 1$ (h) $\lambda = 3, \theta = 2$ (i) $\lambda = 3, \theta = 3$

Figure 13: Comparison of the objective functions of the metamodel and of the simulation model across various demand levels and cost levels.

For all plots of Figure 13, all points are close to the diagonal line. This indicates that the metamodel approximates well the simulation-based objective function. The positive linear correlation also highlights the ability of the metamodel to correctly rank the performance of the solutions, and hence its adequacy for optimization. Unlike Figure 12, this correlation trend holds even for high levels of demand and of supply.

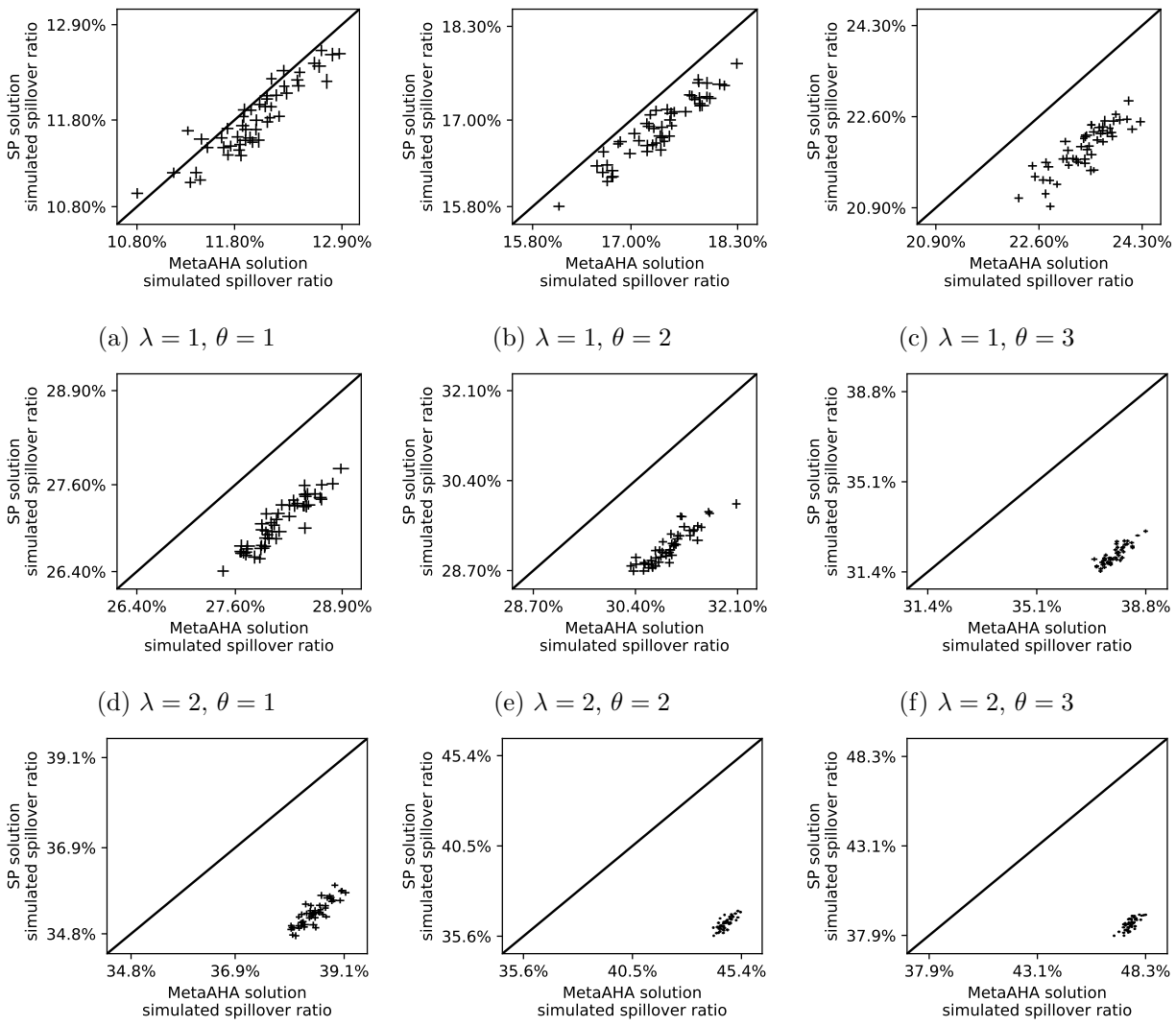
These figures indicate the ability of the metamodel approach to approximate the simulation-based objective function even as the demand-supply interactions become more intricate to model (i.e., when both demand levels and cost levels are high). Recall that the main simplifications of the MIPs (both the metamodel MIP and the SP) compared to the simulator are the lack of the first-come-first-reserve principle, as well as the coarse description of demand spillover (for a description of these simplifications, see the last paragraph of Section 3.5). Hence, we expect the ability of the MIPs to approximate the simulator's objective function to deteriorate as the demand and cost levels increase. This is illustrated by comparing Figures 12 and 13. Moreover, the results of Figure 13 indicate that a simple linear parametric correction to the MIP (through the metamodel parameter $\boldsymbol{\beta}$) suffices to correct for these simplifications. In other words, we need not resort to the formulation of a more intricate analytical optimization problem (e.g., with nonlinear functions to describe spillover in more detail).

Figure 14 illustrates how the coarse MIP modeling of demand spillover impacts the solution quality as congestion increases. The figure considers the same 9 demand and cost combinations as the previous figures. It displays one plot per combination. The y-axis (resp. x-axis) displays the estimated, via simulation, mean spillover ratio over 50 replications using the SP (resp. best MetaAHA) solution. The spillover ratio is the proportion of reservations that incurred spillover (either spatial and/or temporal), meaning that the first choice of and/or reservation time was not available and the user opted for an alternate station and/or reservation time. The spillover ratio is estimated by simulating the performance of the solution. Each point has a 95% confidence interval along both coordinate directions. These intervals are barely visible. Each plot also displays the diagonal line defined by $y = x$. Points along the diagonal line indicate that the SP and the MetaAHA solutions yield a similar proportion of simulated spillovers. As congestion increases (i.e., going from the top to the bottom row plot and/or going from the left-most to the right-most plot), the MetaAHA solution yields increasingly higher spillover ratios compared to the SP solution. In other words, as congestion increases, the solutions identified by MetaAHA allow for a larger proportion of users to spillover compared to the solutions identified by the SP. Recall from Figure 11, that as congestion increases so does the amount by which the MetaAHA solution outperforms the SP solution. Hence, as congestion increases, the MIP modeling of spillover becomes increasingly inaccurate, and combining this MIP modeling with information from the more detailed simulator becomes increasingly important to identify good quality solutions.

4.4.1 Compute times

Table 3 displays, for each demand and cost combination, the computational runtimes for SP and for the MetaAHA run that led to the best solution. All runtimes are reasonable, or even fast, for this problem which is solved offline on an approximately weekly basis. SP runtimes are lower than the MetaAHA runtimes by 1-2 orders of magnitude. Note that there are many straightforward opportunities to parallelize the MetaAHA code and substantially reduce the compute times. The current implementation is not parallelized.

Figure 15 provides a more detailed computational runtime analysis of MetaAHA. It considers the



(a) $\lambda = 1$, $\theta = 1$ (b) $\lambda = 1$, $\theta = 2$ (c) $\lambda = 1$, $\theta = 3$

(d) $\lambda = 2$, $\theta = 1$ (e) $\lambda = 2$, $\theta = 2$ (f) $\lambda = 2$, $\theta = 3$

(g) $\lambda = 3$, $\theta = 1$ (h) $\lambda = 3$, $\theta = 2$ (i) $\lambda = 3$, $\theta = 3$

Figure 14: Comparison of the average spillover ratio, considering 50 demand scenarios, of the SP solution and of the best MetaAHA solution

steps of Algorithm 1 that are the most computationally costly: Steps 1b, 1c, 2, and 4a. The top 9 boxplots consider the computational runtime of Step 4a (which is referred to as "Analytical evaluation") for each of the 9 demand-cost scenarios. The middle 9 boxplots consider the computational runtime of Steps 1b and 1c (which are referred to as "Metamodel solution") for each of the 9 scenarios. The bottom 9 boxplots consider the computational runtime of Step 2 (which is referred to as "Simulation") for each of the 9 scenarios. Each boxplot considers compute time of the given algorithmic step(s) per algorithmic iteration. Since we have run MetaAHA 3 times for each demand-cost scenario, each boxplot considers the compute times per iteration of all iterations of the 3 runs.

The simulation runtimes dominate the runtime per iteration. For a given algorithmic step, the higher the demand (i.e., higher λ), the higher the compute times. For Step 1b and 1c, a higher demand leads to a higher number of MIP constraints, which can increase the compute times. For Step 2, a higher demand leads to a higher number of potential reservations that need to be processed sequentially in time, thus the higher compute times.

Table 3: Computational runtime comparison of SP and of MetaAHA

λ	θ	Solution time of SP (hr)	Run time of MetaAHA (hr)
1	1	0.07	16.92
1	2	0.07	10.89
1	3	0.07	16.97
2	1	0.43	33.84
2	2	0.86	22.18
2	3	0.49	36.78
3	1	1.19	51.33
3	2	1.04	59.14
3	3	1.04	41.9

The high variability of the runtimes for Step 4a ("Analytical evaluation" boxplots) is due to the varying number of points for which the corresponding linear program needs to be solved. More specifically, at a given iteration, an LP is solved for every simulated point for which the analytical network model has not been evaluated. The number of such points can vary substantially across iterations. Note that we solve these LPs sequentially, they could be evaluated in parallel. This is an example of a simple parallelization step that can contribute to reduce compute times if this is of interest.

Figure 15 indicates that, for a given demand level, the time to solve the metamodel decreases as the cost level increases. However, the change in cost level does not have any significant impact on the compute time for simulation or for analytical model evaluation. Since the main compute time bottleneck per iteration is the simulation runtime, the change in cost level does not impact the overall algorithm runtime.

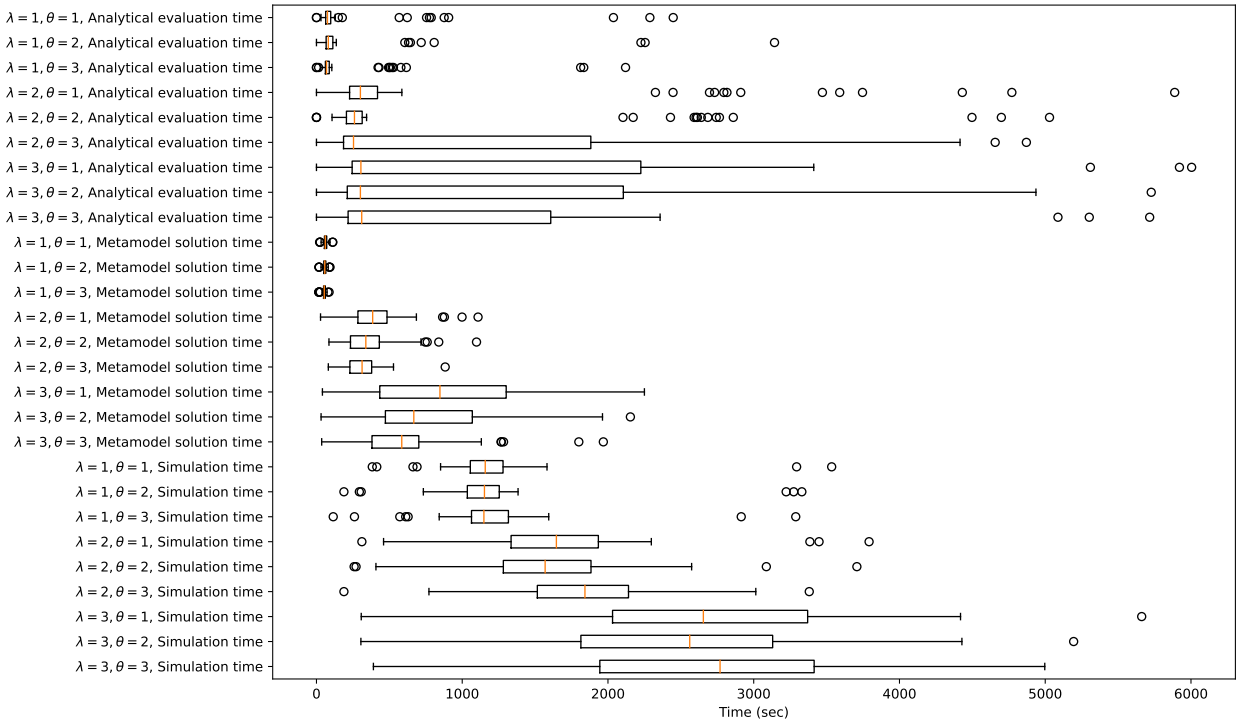


Figure 15: All type of computational time for all demand scenarios

5 Conclusions

This paper formulates a DSO algorithm for a family of large-scale car-sharing fleet allocation problems. The approach is a metamodel SO approach. A novel metamodel is formulated, which is based on a MIP formulation. The metamodel is embedded within a general-purpose DSO algorithm. The proposed algorithm is validated with synthetic toy network experiments. The metamodel approximations of profit are shown to have a positive linear correlation with the higher resolution simulation-based profit estimates. The algorithm is then applied to several Boston case studies using Zipcar car-sharing reservation data, including a high-dimensional problem. The method is first benchmarked versus two types of algorithms that differ only in their use of the analytical MIP information: one benchmark algorithm (AHA) does not use any analytical network information (i.e., no MIP information), the second benchmark algorithm (AHAInit) uses the MIP information only to identify an initial solution but not throughout the optimization process. The experiments indicate that the analytical network model information is useful both at the first iteration of the algorithm and across iterations. The solutions derived by the proposed method are also benchmarked versus the Zipcar deployed solution. Via simulation, the proposed solutions outperform those deployed, both in terms of profit and vehicle utilization. This holds for all considered demand scenarios. We also benchmark MetaAHA versus stochastic programming (SP). SP outperforms the proposed approach for low levels of demand and of cost. As demand and cost levels increase, so does the occurrence of demand spillback and the importance of accounting for the first-come-first-reserve principle. In these cases, the SP approach is outperformed by the metamodel approach.

The combination of the problem-specific analytical MIP with a general-purpose SO algorithm enables the DSO algorithm to: (i) address high-dimensional problems, (ii) become computationally efficient (i.e., it can identify good quality solutions within few simulation observations), (iii) become robust to the quality of the initial points and to the stochasticity of the simulator. More generally, the information provided by the MIP to the SO algorithm enables it to exploit problem-specific structural information. Hence, the simulator is no longer treated as a black box.

We view this general idea of combining analytical MIP formulations with general-purpose SO algorithms, or more broadly with general-purpose sampling strategies, as an innovative and promising area of future research. With the increase in the availability and the resolution of transportation data comes the potential to address more intricate formulations of traditional transportation optimization problems (e.g., formulations with a more detailed probabilistic data-driven description of demand). This paper illustrates how the traditional MIP formulations that exist can be coupled with high-resolution data, a sampling (or simulation) strategy, and a general-purpose SO algorithm, to address this next generation of transportation problems.

There is a wide variety of general-purpose DSO algorithms. As general-purpose algorithms, they can be used to address a broad class of problems. Nonetheless, they are rarely designed such as to achieve good short-term performance (i.e., good performance within a few simulation runs). This paper illustrates how the scalability, computational efficiency, and robustness of these SO algorithms can be enhanced such as to enable them to address realistic transportation problems.

The proposed approach performs optimization preserving the disaggregate information in the data (rather than limiting its use to fitting aggregate demand parameters). This leads to a data-driven approach that exploits the rich information of demand and demand-supply interactions embedded in the data. Nonetheless, this also limits its use for car-sharing markets where data is unavailable or unreliable. In particular, it is not directly applicable to new markets where data has not yet been collected.

In the proposed approach the analytical MIP is solved at every iteration of the DSO algorithm. This setting is reasonable in the case of two-way car-sharing service design mainly because solving the MIP is relatively fast compared to the simulation time. For future work, it is of interest to formulate a learning mechanism that determines, at every DSO iteration, whether to solve the MIP or not.

In past work for continuous SO problems (Tay and Osorio, 2022), we have studied the added value of using information from the physical model to devise more efficient exploration (e.g., to define global sampling distributions) or more efficient exploitation (e.g., optimization) techniques. We have evaluated the added value of combining these ideas. The extension of this work to a discrete optimization setting is an area of future work. Extensions of ongoing interest include the use of MIPs to enable the design of real-time DSO problems.

In recent years, reinforcement learning (RL) has been used to optimize on-demand transportation services. Examples of such studies include Zhu et al. (2021) and Xie et al. (2023), and also illustrated by the recent review of Qin et al. (2022). Using MIPs to enhance the sample efficiency of RL is an area of future work. More generally, exploiting information from low-resolution compute-efficient MIPs to boost the small-sample performance of RL techniques is of interest.

Acknowledgement

This research was partially funded by the project "Car-sharing services: optimal network expansions to integrate automotive with mass-transit systems and to mitigate congestion at the metropolitan scale" of the Ford-MIT Alliance. The authors thank the Ford project members, including Perry MacNeille and Paul Beer. The authors also thank the Zipcar team: Arvind Kannan, Stephen Moseley, Lauren Alexander and James Hardison for the data, as well as their valuable input and feedback.

Appendix A Algorithmic details

In this section, we present algorithmic details of MetaAHA. The algorithmic steps refer to Algorithm 1. In Step 2 of the algorithm, the number of simulation replications to run for a given point $\boldsymbol{x}$ up until and including iteration k, denoted $N_k(\boldsymbol{x})$, is computed based on the approach of AHA (Xu et al., 2013). It is given by $N_k(\boldsymbol{x}) = \min\{5, \lceil 5(\log k)^{1.01} \rceil\}$. If at a given iteration k, the number of simulation replications of point $\boldsymbol{x}$ obtained from previous iterations is greater or equal to $N_k(\boldsymbol{x})$, then we do not evaluate additional replications.

In Step 2 of the algorithm, the hyperbox is updated based on the following AHA approach (Xu et al., 2013). Let $\boldsymbol{x}_k$ denote the current iterate at iteration k, with the ith element denoted $x_{k,i}$. Let $\mathcal{E}(k)$ denote the set of points that have been simulated up until and including iteration k. The hyperbox is defined (or updated) at iteration k as $\mathcal{H}_k = \{\boldsymbol{x} : l_{k,i} \leq x_i \leq u_{k,i}, \forall i \in \mathcal{I}\}$. The bounds $l_{k,i}$ and $u_{k,i}$ are defined as follows.

$$l_{k,i} = \max_{\boldsymbol{x} \in \mathcal{E}(k) \setminus \{\boldsymbol{x}_k\}} \{x_i : x_i < x_{k,i}\}, \forall i \in \mathcal{I}.$$

If $l_{k,i}$ is empty, then set $l_{k,i} = 0$. Similarly,

$$u_{k,i} = \min_{\boldsymbol{x} \in \mathcal{E}(k) \setminus \{\boldsymbol{x}_k\}} \{x_i : x_i > x_{k,i}\}, \forall i \in \mathcal{I}.$$

If $u_{k,i}$ is empty, then set $u_{k,i} = N_i$.

Step 4b of the algorithm fits the metamodel parameter $\boldsymbol{\beta}_k$ by solving the below least squares problem, which is formulated and discussed in more detail in Osorio and Bierlaire (2013).

$$\min_{\boldsymbol{\beta}_k} \sum_{\boldsymbol{x} \in \mathcal{E}(k)} \left[w_k(\boldsymbol{x}) \left(\hat{g}(\boldsymbol{x}; \boldsymbol{q}_1) - m_k(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\beta}_k, \boldsymbol{q}_2) \right) \right]^2 + \left(w_0(\beta_{k,0} - 1) \right)^2 + \sum_{i=1}^{|\mathcal{I}|+1} \left(w_0 \beta_{k,i} \right)^2, \tag{14}$$

where w_0 is a fixed scalar weight, $\hat{g}(\boldsymbol{x}; \boldsymbol{q}_1)$ represents the simulated estimate of the profit function for point $\boldsymbol{x}$, and the weight $w_k(\boldsymbol{x})$ function is defined as $w_k(\boldsymbol{x}) = 1/(1 + \|\boldsymbol{x} - \boldsymbol{x}_k\|_2)$. The least squares problem minimizes a weighted distance between the simulated profit estimates $\hat{g}$ and the metamodel predictions m_k. Each point is weighted by a distance function that gives more weights to points that are closer to the current iterate, such as to improve the local (i.e., close to the current iterate) fit of the metamodel. The additional terms in the least squares problem are included such as to ensure a full rank least-squares matrix.

Figure 16 provides a flowchart summary of the MetaAHA algorithm.

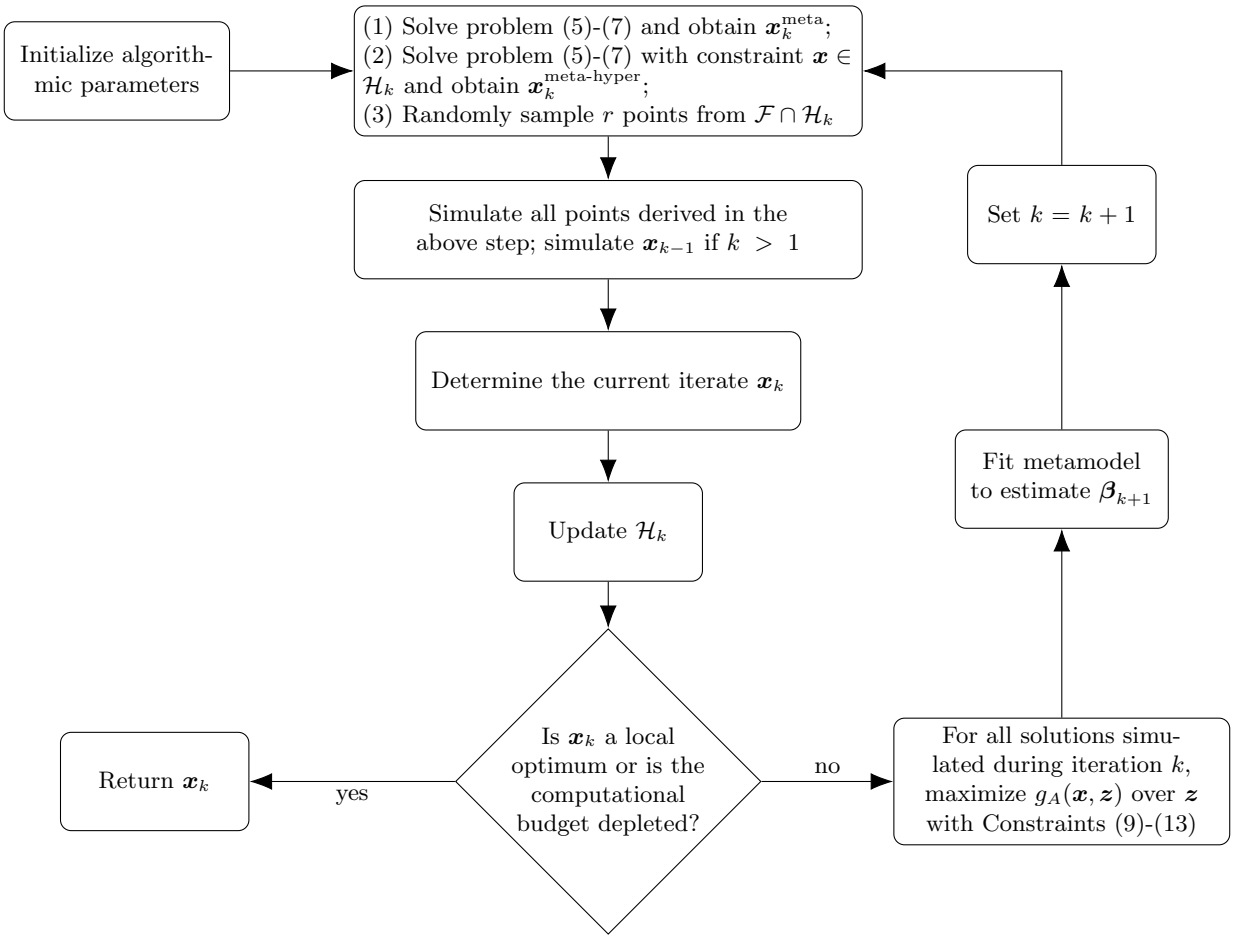


Figure 16: MetaAHA Steps

Appendix B Stochastic programing (SP) formulation

To formulate the SP model, we use the notation of the main manuscript and introduce the following notation.

Q	number of demand scenarios;
$d_{tl}^{i(q)}$	number of customers that desire a reservation at station i with start time t and duration l in demand scenario q;
$z_{tl}^{i(q)}$	number of customers that make a reservation at station i with start time t and duration l in demand scenario q;
$z_{tl}^{ji(q)}$	number of customers that desire to make a reservation at station j with start time t and duration l but make an adjusted reservation at station i with start time t and duration l in demand scenario q;
$\boldsymbol{z}$	vector that combines all variables $\{z_{tl}^{i(q)}\}$ and $\{z_{tl}^{ji(q)}\}$;
$\pi^{(q)}$	probability of scenario q, set to $1/Q$;
$\boldsymbol{q}_3$	vector of exogenous parameters;
g_{SP}	analytical approximation of g (Equation (1)) derived by the SP model.

We view the fleet allocation strategy $\boldsymbol{x}$ as the first-stage decision variables, and the demand-supply interaction $\boldsymbol{z}$ as the second-stage decision variables. The SP problem is formulated as follows.

$$\max_{\boldsymbol{x},\boldsymbol{z}} \quad g_{SP}(\boldsymbol{x},\boldsymbol{z};\boldsymbol{q}_3) = \sum_{q=1}^{Q} \pi^{(q)} \left(\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}_i} \sum_{t\in\mathcal{T}} \sum_{l\in\mathcal{L}} p^{ij} r_{tl} z_{tl}^{ij(q)} \right) - \sum_{i\in\mathcal{I}} c_i x_i, \tag{15}$$

subject to

$$\sum_{j\in\mathcal{I}_i} z_{tl}^{ji(q)} = z_{tl}^{i(q)} \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall l\in\mathcal{L}, \forall q\in\{1,2,\ldots,Q\}, \tag{16}$$

$$\sum_{j\in\mathcal{I}_i} z_{tl}^{ij(q)} \le d_{tl}^{i(q)} \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall l\in\mathcal{L}, \forall q\in\{1,2,\ldots,Q\}, \tag{17}$$

$$\sum_{l\in\mathcal{L}} z_{tl}^{i(q)} + \sum_{l\in\mathcal{L}} \sum_{t'\in\mathcal{T}_1(t,l)} z_{t'l}^{i(q)} \le x_i \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall q\in\{1,2,\ldots,Q\}, \tag{18}$$

$$z_{tl}^{i(q)} \in \mathbb{R}_+ \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall l\in\mathcal{L}, \forall q\in\{1,2,\ldots,Q\}, \tag{19}$$

$$z_{tl}^{ij(q)} \in \mathbb{R}_+ \qquad \forall i\in\mathcal{I}, \forall j\in\mathcal{I}_i, \forall t\in\mathcal{T}, \forall l\in\mathcal{L}, \forall q\in\{1,2,\ldots,Q\}, \tag{20}$$

$$\boldsymbol{x}\in\mathcal{F}, \tag{21}$$

where $\mathcal{T}_1(t,l) = \{t'\in\mathcal{T} : t'+1 \le t \le t'+l-1\}$. In this model, the exogenous parameters are $d_{tl}^{i(q)}$ and $\pi^{(q)}$, as well as r_{tl}, c_i, p^{ij}, $t_{\max}$ and $l_{\max}$ defined in Section 3.3, represented by the vector $\boldsymbol{q}_3$. The objective function (15) is the expected revenue over all scenarios minus the cost. Constraints (16)-(20) are the equivalent of their MIP counterparts Constraints (9)-(13), respectively. Constraint (21) is equivalent to Constraint (7).

Appendix C Sampling of feasible solutions for the experiments of Figures 12 and 13

For a given demand scenario, let $\boldsymbol{x}^{(1)}$ be the SP solution and $\boldsymbol{x}^{(2)}$ be the best MetaAHA solution of the experiments of Section 4.4. We use the following procedure to generate a solution near the line connecting the SP optimal solution and the best MetaAHA solution:

- Step 1: generate $u \sim U(0,1)$, where $U(0,1)$ is the standard uniform distribution.
- Step 2: let $\tilde{\boldsymbol{x}} = \boldsymbol{x}^{(1)} + u\left(\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}\right)$.
- Step 3: build a hyperbox $\mathcal{H}(\tilde{\boldsymbol{x}}) = \{\boldsymbol{x} : \tilde{x}_i - 2 \leq x_i \leq \tilde{x}_i + 2, \forall i \in \mathcal{I}\}$.
- Step 4: randomly sample a point $\boldsymbol{x}$ from $\mathcal{H}(\tilde{\boldsymbol{x}}) \cap \mathcal{F}$ using the uniform sampling distribution of AHA (Xu et al., 2013).

References

Ankenman, B., Nelson, B. L., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research*, 58(2):371–382.

Balac, M., Ciari, F., and Axhausen, K. W. (2016). Evaluating the influence of parking space on the quality of service and the demand for one-way carsharing: a Zürich area case study. In *Proceedings of the 95th Annual Meeting of the Transportation Research Board*.

Balac, M., Ciari, F., and Axhausen, K. W. (2017). Modeling the impact of parking price policy on free-floating carsharing: case study for Zürich, Switzerland. *Transportation Research Part C*, 77:207–225.

Barton, R. R. and Meckesheimer, M. (2006). Metamodel-based simulation optimization. In Henderson, S. G. and Nelson, B. L., editors, *Handbooks in operations research and management science: simulation*, volume 13, pages 535–574. Elsevier, Amsterdam, Netherlands.

Becker, H., Ciari, F., and Axhausen, K. W. (2017). Comparing car-sharing schemes in Switzerland: User groups and usage patterns. *Transportation Research Part A*, 97:17–29.

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

Boyacï, B., Zografos, K. G., and Geroliminis, N. (2017). An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. *Transportation Research Part B*, 95:214–237.

Brandstätter, G., Gambella, C., Leitner, M., Malaguti, E., Masini, F., Puchinger, J., Ruthmair, M., and Vigo, D. (2016). Overview of optimization problems in electric car-sharing system design and management. In Dawid, H., Doerner, K. F., Feichtinger, G., Kort, P. M., and Seidl, A., editors, *Dynamic Perspectives on Managerial Decision Making: Essays in Honor of Richard F. Hartl*, volume 22, pages 441–471. Springer International Publishing, Cham, Switzerland.

Cepolina, E. M. and Farina, A. (2012). A new shared vehicle system for urban areas. *Transportation Research Part C*, 21(1):230–243.

Chen, X., Osorio, C., and Santos, B. F. (2019). Simulation-based travel time reliable signal control. *Transportation Science*.

Chen, X., Zhu, Z., He, X., and Zhang, L. (2015). Surrogate-based optimization for solving a mixed integer network design problem. *Transportation Research Record*, 2497(1):124–134.

Chick, S. E. and Inoue, K. (2001). New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 49(5):732–743.

Chiraphadhanakul, V. (2013). *Large-scale analytics and optimization in urban transportation: Improving public transit and its integration with vehicle-sharing services*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Chong, L. and Osorio, C. (2017). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*, 52(3):637–656.

Ciari, F., Balac, M., and Axhausen, K. W. (2016a). Modeling carsharing with the agent-based simulation MATSim: state of the art, applications, and future developments. *Transportation Research Record*, 2564(1):14–20.

Ciari, F., Balac, M., and Balmer, M. (2015). Modelling the effect of different pricing schemes on free-floating carsharing travel demand: a test case for Zurich, Switzerland. *Transportation*, 42(3):413–433.

Ciari, F., Balmer, M., and Axhausen, K. W. (2009). Concepts for large-scale carsharing system: Modeling and evaluation with agent-based approach. In *Proceedings of the 88th Annual Meeting of the Transportation Research Board*.

Ciari, F., Bock, B., and Balmer, M. (2014). Modeling station-based and free-floating carsharing demand: test case study for Berlin. *Transportation Research Record*, 2416(1):37–47.

Ciari, F., Schuessler, N., and Axhausen, K. W. (2013). Estimation of carsharing demand using an activity-based microsimulation approach: model discussion and some results. *International Journal of Sustainable Transportation*, 7(1):70–84.

Ciari, F., Weis, C., and Balac, M. (2016b). Evaluating the influence of carsharing stations' location on potential membership: a Swiss case study. *EURO Journal on Transportation and Logistics*, 5(3):345–369.

Coll, M.-H., Vandersmissen, M.-H., and Thériault, M. (2014). Modeling spatio-temporal diffusion of carsharing membership in Québec City. *Journal of Transport Geography*, 38:22–37.

Correia, G. H. A. and Antunes, A. P. (2012). Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E*, 48(1):233–247.

Correia, G. H. A., Jorge, D. R., and Antunes, D. M. (2014). The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: an application in Lisbon, Portugal. *Journal of Intelligent Transportation Systems*, 18(3):299–308.

De Lorimier, A. and El-Geneidy, A. M. (2013). Understanding the factors affecting vehicle usage and availability in carsharing networks: A case study of communauto carsharing system from Montréal, Canada. *International Journal of Sustainable Transportation*, 7(1):35–51.

Deng, Y. (2015). *Design and management of mobility-on-demand (MOD) transportation systems considering demand uncertainty and flexibility - a simulation-based approach*. PhD thesis, National University of Singapore, Singapore.

Duncan, M. (2011). The cost saving potential of carsharing in a US context. *Transportation*, 38(2):363–382.

Fields, E., Osorio, C., and Zhou, T. (2017). A data-driven car sharing simulator for inferring latent demand. Technical report, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA. Available at: `http://web.mit.edu/osorioc/www/papers/fields17Sim.pdf`.

Fields, E., Osorio, C., and Zhou, T. (2021). A data-driven method for reconstructing a distribution from a truncated sample with an application to inferring car-sharing demand. *Transportation Science*, 55(3):616–636.

Firnkorn, J. and Müller, 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):1519–1528.

Frazier, P. I., Powell, W. B., and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439.

Google Maps (2017a). 315 Zipcar stations in Boston area. `https://drive.google.com/open?id=1HggY_4-dOuYU71lhnudL8WqeSJA&usp=sharing`. Accessed: 2017-09-22.

Google Maps (2017b). Zipcar stations in Boston South End neighborhood. `https://drive.google.com/open?id=1hOvbRIjfZJjF5L3OfoAThiq0_p8&usp=sharing`. Accessed: 2017-09-22.

Greenhall, A. (2016). Experimentation in a ridesharing marketplace. `https://eng.lyft.com/https-medium-com-adamgreenhall-simulating-a-ridesharing-marketplace-36007a8a31f2`. Accessed: 2018-08-15.

He, L., Mak, H.-Y., Rong, Y., and Shen, Z.-J. M. (2017). Service region design for urban electric vehicle sharing systems. *Manufacturing & Service Operations Management*, 19(2):309–327.

Hong, L. J. and Nelson, B. L. (2006). Discrete optimization via simulation using COMPASS. *Operations Research*, 54(1):115–129.

Hong, L. J., Nelson, B. L., and Xu, J. (2015). Discrete optimization via simulation. In Fu, M. C., editor, *Handbook of simulation optimization*, volume 216, pages 9–44. Springer, New York, NY, USA.

Jian, N., Freund, D., Wiberg, H. M., and Henderson, S. G. (2016). Simulation optimization for a large-scale bike-sharing system. In Roeder, T. M. K., Frazier, P. I., Szechtman, R., Zhou, E., Huschka, T., and Chick, S., editors, *Proceedings of the 2016 Winter Simulation Conference*, pages 602–613, Piscataway, NJ, USA. IEEE Press.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

Jorge, D., Barnhart, C., and Correia, G. H. A. (2015). Assessing the viability of enabling a round-trip carsharing system to accept one-way trips: Application to Logan Airport in Boston. *Transportation Research Part C*, 56:359–372.

Jorge, D. and Correia, G. H. A. (2013). Carsharing systems demand estimation and defined operations: A literature review. *European Journal of Transport and Infrastructure Research*, 13(3):201–220.

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

Kleijnen, J. P., Van Beers, W., and Van Nieuwenhuyse, I. (2010). Constrained optimization in expensive simulation: Novel approach. *European Journal of Operational Research*, 202(1):164–174.

Lu, M., Chen, Z., and Shen, S. (2017). Optimizing the profitability and quality of service in carshare systems under demand uncertainty. *Manufacturing & Service Operations Management*, 20(2):162–180.

MATSim (2018). MATSim: multi-agent transport simulation. `https://www.matsim.org`. Accessed: 2018-08-14.

Millard-Ball, A., Murray, G., Ter Schure, J., Fox, C., and Burkhardt, J. (2005). Car-sharing: Where and how it succeeds. Technical Report TCRP Report 108, Transportation Research Board of the National Academies, Washington, D.C., USA.

Nagaraj, K. S. (2014). *Stochastically constrained simulation optimization on mixed-integer spaces*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA.

Nair, R. and Miller-Hooks, E. (2014). Equilibrium network design of shared-vehicle systems. *European Journal of Operational Research*, 235(1):47–61.

NCSL (2017). Car sharing - state laws and legislation. `http://www.ncsl.org/research/transportation/car-sharing-state-laws-and-legislation.aspx`. NCSL stands for National Conference of State Legislatures. Accessed: 2018-08-12.

Nelson, B. L. (2010). Optimization via simulation over discrete decision variables. In Hasenbein, J. J., Gray, P., and Greenberg, H. J., editors, *Tutorials in Operations Research: Risk and Optimization in an Uncertain World*, volume 7, pages 193–207. INFORMS, Hanover, MD, USA.

O'Mahony, E. D. (2015). *Smarter tools for (Citi) bike sharing*. PhD thesis, Cornell University, Ithaca, New York, USA.

Osorio, C. (2019). Dynamic origin-destination matrix calibration for large-scale network simulators. *Transportation Research Part C: Emerging Technologies*, 98:186–206.

Osorio, C. and Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems. *Operations Research*, 61(6):1333–1345.

Osorio, C. and Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science*, 49(3):623–636.

Osorio, C. and Nanduri, K. (2015). Energy-efficient urban traffic management: A microscopic simulation-based approach. *Transportation Science*, 49(3):637–651.

Qin, Z. T., Zhu, H., and Ye, J. (2022). Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144:103852.

Salemi, P. L. (2014). *Gaussian Markov random fields and moving least squares for metamodeling and optimization in stochastic simulation*. PhD thesis, Northwestern University, Evanston, Illinois, USA.

Schmöller, S., Weikl, S., Müller, J., and Bogenberger, K. (2015). Empirical analysis of free-floating carsharing usage: The Münich and Berlin case. *Transportation Research Part C*, 56:34–51.

Sebastiani, M. T., Lüders, R., and Fonseca, K. V. O. (2016). Evaluating electric bus operation for a real-world BRT public transportation using simulation optimization. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2777–2786.

Shaheen, S. and Chan, N. (2016). Mobility and the sharing economy: Potential to facilitate the first-and last-mile public transit connections. *Built Environment*, 42(4):573–588.

Shaheen, S. A. and Cohen, A. P. (2013). Carsharing and personal vehicle services: Worldwide market developments and emerging trends. *International Journal of Sustainable Transportation*, 7(1):5–34.

Stillwater, T., Mokhtarian, P., and Shaheen, S. (2009). Carsharing and the built environment: Geographic information system-based study of one US operator. *Transportation Research Record*, 2110(1):27–34.

Sun, L., Hong, L. J., and Hu, Z. (2014). Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Operations Research*, 62(6):1416–1438.

Swisher, J. R., Jacobson, S. H., and Yücesan, E. (2003). Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Transactions on Modeling and Computer Simulation*, 13(2):134–154.

Tay, T. and Osorio, C. (2022). Bayesian optimization techniques for high-dimensional simulation-based transportation problems. *Transportation Research Part B*, 164:210–243.

Wang, H., Pasupathy, R., and Schmeiser, B. W. (2013). Integer-ordered simulation optimization using R-SPLINE: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM Transactions on Modeling and Computer Simulation*, 23(3):1–24.

Wild, S. M., Regis, R. G., and Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219.

Xie, J., Frazier, P. I., and Chick, S. E. (2016). Bayesian optimization via simulation with pairwise sampling and correlated prior beliefs. *Operations Research*, 64(2):542–559.

Xie, J., Liu, Y., and Chen, N. (2023). Two-sided deep reinforcement learning for dynamic mobility-on-demand management with mixed autonomy. *Transportation Science*, 0(0).

Xu, J. (2012). Efficient discrete optimization via simulation using stochastic Kriging. In Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., and Uhrmacher, A. M., editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 1–12, Piscataway, NJ, USA. IEEE Press.

Xu, J., Nelson, B. L., and Hong, J. L. (2010). Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 20(1):1–29.

Xu, J., Nelson, B. L., and Hong, L. J. (2013). An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS Journal on Computing*, 25(1):133–146.

Zhang, C., Osorio, C., and Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B*, 97:214–239.

Zhou, T. (2015). Network design for integrated vehicle-sharing and public transportation service. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Zhu, Z., Ke, J., and Wang, H. (2021). A mean-field markov decision process model for spatial-temporal subsidies in ride-sourcing markets. *Transportation Research Part B: Methodological*, 150:540–565.

Zipcar (2017). Zipcar officially launches in Worcester, Massachusetts. `http://www.zipcar.com/press/releases/worcester`. Accessed: 2017-09-26.