Calibrating Activity-based Travel Demand Model System via Microsimulation

by

Siyu Chen

B.S., University of California, Berkeley (2017)

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of

Master of Science in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Signature redacted

Author .....
Department of Civil and Environmental Engineering
May 17, 2019

Signature redacted

Certified by .....
Moshe E. Ben-Akiva
Edmund K. Turner Professor of Civil and Environmental Engineering
Thesis Supervisor

Signature redacted

Accepted by .....
Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee

Calibrating Activity-based Travel Demand Model System via Microsimulation

by

Siyu Chen

Submitted to the Department of Civil and Environmental Engineering
on May 17, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Transportation

Abstract

This thesis addresses the problem of calibrating activity-based travel demand model systems. After estimation, it is common practice to use aggregate measurements to calibrate the estimated model system's parameters. However, calibration of activity-based model systems has received much less attention. Existing calibration approaches are myopic heuristics in the sense that they do not consider inter-dependency among choice-models and do not have a systematic way to adjust model parameters. Also, other simulation-based approaches do not perform well in large-scale applications. In this thesis, we focus on utility-maximizing nested logit activity-based model systems and calibrating count based aggregate statistics like OD flows, mode shares, activity shares and so on. We formulate the calibration problem as a simulation-based optimization problem and propose a stochastic gradient-based solution procedure to solve it.

The solution procedure relies on microsimulation to calculate expected aggregate statistics of interest to the calibration problem. Additionally, we derive approximate analytical expressions for the gradient of the objective function —that are evaluated through microsimulation on mini-batches of the population. The proposed solution procedure is sensitive to the fundamental structure of the activity-based model system and is non-myopic in considering the dependencies across its model components. Finally, we show —through a real-world application— that the proposed solution procedure outperforms other state-of-the-art purely simulation-based optimization approaches in terms of computational efficiency, stability, and convergence. We also compare various gradient-based solution algorithms to determine the best algorithm to update the parameters. This work has the potential to facilitate wider and easier application of activity-based model systems.

Thesis Supervisor: Moshe E. Ben-Akiva
Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my research advisor and thesis supervisor Professor Moshe Ben-Akiva for his patient guidance and useful critiques. Not only he provided me invaluable suggestions on several research projects, but also he set very high standards for me and pushed me to do better.

I would also like to extend my deepest gratitude to Dr. A. Arun Prakash. He played a decisive role in coming up this research idea with me. This thesis would not have been possible without his patient support and numerous advice. Apart from this thesis, he constantly provided invaluable insight into other research projects I involved with.

I must also thank Professor Carlos Lima De Azevedo, Dr. Jimi Oke, and Dr. Ravi Seshadri for their great amount of assistance on research projects.

I also had great pleasure of taking multiple useful courses with amazing professors at MIT. Their knowledge, expertise and wisdom are valuable to me.

I am very grateful to my colleagues: Youssef Medhat Aboutaleb, Mazen Danaf and Yifei Xie. I feel fortunate to study and work with them. I got inspired by them from time to time.

Last but not least, I am deeply indebted to my parents for their endless love and continuous support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Travel models are created to help managers, planners and directors make informed decisions. They produce quantitative information about travel demand and transportation system performance. There are various forms of travel models developed with different modeling assumptions and purposes. Using person-trips as the unit of analysis, trip-based models, developed in the 1950s, are widely used in practice. However, trip-based models suffer from several major limitations. First, trip-based models are not behaviorally realistic. They model demand for trip making rather than for activities. Second, because trip-based models are based on person-trips, they do not capture any dependency among household members. Third, they assume that individuals within the same group share the same characteristics, which causes demographic aggregation errors. Also, they suffer from spatial and temporal aggregation errors because they assume all households in a zone are treated as identical and only differentiate between peak and off-peak periods. Fourth, they cannot model induced travel because each person-trip is independent. Because of aforementioned drawbacks, limited types of policies can be analyzed and activity-based models are developed.

1.1 Activity-based models

Based on the principle that travel demand derives from activities, activity-based models model individual's behaviors sequentially in a continuous domain of time and space. Rasouli and Timmermans [1] identified three activity-based modeling approaches: 1). constraints-based models, 2). utility-maximizing models, and 3). computational process models. They are mainly different in terms of the way to model individual and household activity patterns.

Constraints-based models are able to identify feasible activity schedules under space and time constraints instead of directly predicting individual and household activity-travel patterns. However, they have several limitations, including the limited consideration on household accessibility instead of individual accessibility, the unrealistic assumption of isotropic conditions [2], and lack mechanisms to deal with choice behavior under uncertainty. Regarding these limitations, some generalizations of the classic space-time prism are proposed to address them and are brought together in [3].

On the other hand, utility-maximizing models utilize discrete choice models and other econometric models to maximize individual's utility in choosing among travel patterns. As an advanced model applied widely in practice, the day activity schedule (DAS) model system [4, 5, 6, 7, 8, 9] is a nested logit model system consisting of several nests to model travel choice as a multidimensional choice with shared unobserved elements. The topmost choice is activity pattern choice, in which time of day, mode and destination for primary tour, time of day, mode and destination for secondary tour are modeled. Conditional on the activity pattern choice, lower levels include time-of-day, mode and destination choice for the primary tour, time-of-day, mode and destination choice for the secondary tours. However, it has coarse representation of schedule as combinations of discrete time periods. Apart from the DAS model, there are other activity-based models, like the Prism-Constrained Activity Travel Simulator (PCATS) [10], the Comprehensive Econometric Micro-simulator for Daily Activity-travel Patterns (CEMDAP) [11], and the Comprehensive Utility-based System of

Travel Options Modelling (CUSTOM)[12].

More recently, computational process models are created to relax the strict and behaviorally unrealistic assumption of utility-maximizing behavior. The most comprehensive one is Albatross [13], which controls the scheduling processes in terms of a sequence of 27 steps. However, such scheduling process is based on a priori assumptions of the researchers. Some other applications are SCHEDULER [14], AMOS [15] and ADAPTS [16].

1.2 Thesis motivation and objective

As a result of research in various modeling aspects over the past years, **Activity-Based Model systems (ABMs)** have become a part of integrated travel demand model systems. However, the fundamental task of calibrating ABMs has received limited attention in literature. In this light, this thesis presents a systematic approach to calibrate ABMs that explicitly considers the complex interactions among its components.

The motivation for this study is threefold. First, although ABMs is a powerful and widely accepted modeling framework, there are limited studies on their calibration. Efficiently calibrating these model systems is crucial for their wider and easier application. Second, the existing calibration approaches for ABMs are predominantly myopic heuristics that overlook inter-dependency among choice-models and can result in ill-formed model systems —especially in complex ABMs. Third, relying on "black box" —purely simulation-based— approaches to calibrate ABMs have not been successful in large-scale applications.

In light of the aforementioned motivations, the objectives of this thesis are as follows:

1. To propose a generic mathematical formulation of the ABM calibration problem.

2. To propose an efficient solution procedure, adopting microsimulation and structural relationships within the ABMs.

3. To apply and test the proposed formulation and solution procedure on an existing ABM.

This study contributes to the existing literature in following aspects. First we formulate the ABM calibration problem as a simulation-based optimization problem. Second, for nested logit activity-based models, we derive the expressions for unbiased estimators of the simulated aggregate statistics of interest which are used in the objective function of the calibration problem and can be computed via microsimulation. Third, using the proposed analytical expressions of the objective function we derive an unbiased estimator of the gradient of the objective function —with respect to any parameter of interest in the DAS model system— through microsimulation. Fourth, this proposed generic approach is able to get both objective function and its gradient in a single run of the microsimulation. It explicitly considers the structure of the ABMs to accurately and efficiently calculate the gradient of the objective function without the need to rely on commonly used black-box approaches. Fifth, we propose a stochastic gradient-based approach that uses a sample of the population to iteratively solve the formulated problem. Finally, we demonstrate the correctness and efficiency of the procedure through a real-world application and test various gradient-based solution algorithms. Comparison with the existing simulation based state-of-the-art algorithms —simultaneous perturbation stochastic approximation (SPSA)[17]— showed that the proposed algorithm is computationally efficient, more stable, and results in faster convergence.

1.3 Thesis outline

Chapter 2 examines past travel demand model calibration efforts. Following past development, Chapter 3 formulates the calibration problem as a simulation-based optimization problem and proposes a gradient-based solution procedure to solve the optimization problem. In addition, it also reports derivations of analytical gradient expressions. Chapter 4 contains a illustrative example for better understanding of the solution procedure. Chapter 5 presents the results from experiments done on a

prototype city to demonstrate the efficacy of the calibration approach. The conclusions drawn from the research are reported in Chapter 6, along with future research work.

Chapter 2

Background

Transportation Model Systems have become more complex as a result of readily available computational resources and the need to more realistically model activity-travel planning. Further, various simulation systems are being developed as tools to facilitate analysis and decision-making by researchers, practitioners, and policy-makers [18, 19, 20]. It is increasingly being recognized that calibration and validation of these model systems, specifically the activity-based models, is one of the key aspects that needs to be addressed for a wider and more reliable application of the modeling tools. Yet, very little attention has been devoted to the development and application of robust techniques for calibrating the demand-side parameters of activity-based model systems.

2.1 Calibration of trip-based models

Calibration of traditional trip-based models has a long history and includes approaches that are both manual and algorithmic. The manual approaches involve estimating and constructing of the model system and then adjusting individual models in it to better match the observed statistics. Some of the techniques include (i) adjusting zonal scaling factors, (ii) introducing OD K-factors, (iii) changing alternative specific constant of the mode choice-models. The zonal scaling factors are used for some specific zones to adjust the trips generation and attraction rates [21]. Simi-

larly, OD K-factors are constants that are added to specific origin-destination pairs in the destination choice-models (or trip generation models) to make OD flows of models closer to the observed statistics [22]. Finally, the alternative specific constants of the modes are changed such that mode-shares from the model match the observed statistics [23].

Algorithmic approaches to calibrate trip-based models have been predominantly focused on estimating OD matrix. Three modeling approaches are postulated in [24]: traffic modeling based approaches, statistical inference approaches and gradient based solution techniques. As an example, [25] formulated the problem as an analytical optimization problem where the observations were the sensor counts. Recently, the OD estimation problems have been formulated as simulation-based optimization problems and solved using stochastic algorithms [26, 27, 28]. The trip-based models are considered in the OD estimation problem through a seed matrix that is part of the objective function. [26] concludes that simultaneous perturbation stochastic approximation (SPSA) performs well in real word applications in terms of solution quality and computational efficiency compared to other simulation optimization algorithms like Box-Complex, SNOBFIT, and finite difference stochastic approximation (FDSA).

Both FDSA and SPSA are stochastic approximation (SA) algorithms. With model parameters of interest denoted as $\boldsymbol{\beta}$ and formulated objective function denoted as $L(\boldsymbol{\beta})$, both algorithms use approximate gradient of objective function denoted as $\nabla_{\boldsymbol{\beta}} L'(\boldsymbol{\beta})$ to update model parameters. The update rule at iteration κ can be written as:

$$\boldsymbol{\beta}^{\kappa+1} = \boldsymbol{\beta}^{\kappa} - \alpha^{\kappa} \nabla_{\boldsymbol{\beta}^{\kappa}} L'(\boldsymbol{\beta}^{\kappa}) \tag{2.1}$$

where α^{κ} represents the step size at iteration κ, which becomes smaller as κ increases.

However, FDSA approximates the gradient by perturbing each parameter of $\boldsymbol{\beta}$ in both plus and minus direction separately and takes $2|\boldsymbol{\beta}|$ runs to approximate while

SPSA efficiently estimates the gradient by perturbing all parameters in $\boldsymbol{\beta}$ simultaneously. Thus the approximation of gradient only needs two function evaluations instead of $2|\boldsymbol{\beta}|$ function evaluations in FD scheme. In terms of convergence performance, [17] argues that SPSA performs as good as FDSA while has $|\boldsymbol{\beta}|$-fold simulation saving. In this study, we compare our proposed solution procedure with SPSA and more detail about SPSA algorithm is in section 5.2.

However, the performance of SPSA suffers from gradient approximation error when it is applied to a large-scale system with sparse correlations between parameters and measurements. [27, 28] propose Weighted SPSA (W-SPSA) to overcome this limitation by relying on a weight matrix, which represents the appropriate correlation structure. However, generating the essential ingredient —weight matrix— is not systematic, which considerably limits the applicability of the algorithm.

[29, 30] also propose a computationally efficient metamodel simulation-based optimization approach, which embeds within the simulation-based optimization algorithm information from an analytical yet differentiable and tractable network model. In addition, online estimation of OD matrix for real-time applications has also gained attention [31] with recent work on the problem scalability [32, 33].

2.2 Calibration of activity-based models

Calibration of activity-based model systems has received much less attention in the literature although they are increasingly applied in practice [34, 35, 36, 20]. ABMs also get questioned by many practitioners because it is in many aspects easier to adjust four-step models to fit base level traffic counts exactly [37]. The prevailing practice, which is similar to that of the trip-based models, is to calibrate individual models to better match the observed statistics [38]. Researchers and practitioners have been using top-down model-specific sequential methods, since adjustments to upper level models will tend to have a higher impact on lower level models than vice-versa. [38], for example, calibrated sequentially from the top to the bottom of the DaySim model hierarchy: longer term person and household choices, single

day-long activity pattern choices, tour-level choices, and trip-level choices. As the number of parameters in each level can be quite significant, such calibration generally focuses on the adjustment of alternative specific constants, which tend to capture the effects of variables ignored in the estimation phase [39]. Such adjustment procedure has also been recently proposed when transferring an existing model system and re-calibrating it with local data to a new region [40]. In similar settings, to minimize the burden of the manual iterative procedure at each model level, some researchers and practitioners adopt an incremental alternative-specific constant adjustment method using *dampening factors* [41, 23]. When an aggregate measurement for a given choice dimension is available in terms of shares (e.g.: mode share, activity type shares, etc) constants are updated at each iteration as follows:

$$\beta^{\kappa}_{\pi,i,p} = \beta^{\kappa-1}_{\pi,i,p} + DF_{\pi,i} * ln\left(\frac{S^{\kappa}_{\pi,i}}{A_{\pi,i}}\right)$$

where κ is the iteration index, $\beta^{\kappa}_{\pi,i,p}$ is the alternative specific constant coefficient p in the utility expression of alternative i in a choice-model π to be calibrated, $A_{\pi,i}$ is the observed share of alternative i, $S^{\kappa}_{\pi,i}$ is the estimated share of alternative i at iteration κ. The DF is the *dampening factor* usually set between 0.10 to 0.75 which helps eliminating the oscillating pattern between iterations of the calibration procedure.

It is common practice to normalize the above adjustments by setting one alternative as a baseline and subtracting its adjustment from all the alternatives. The steps described above are repeated till the predicted and the observed distributions are satisfactory close. This state-of-the-practice method is highly limited in the covered degrees of freedom and the considered interactions between the different model system components, thus often forcing modellers to iteratively revisit the estimation phase and slows down the calibration process [42].

[43] proposed that the activity-based models can be calibrated to external OD flow information derived from traffic counts in an indirect approach or direct approach. The indirect approach tries to analyze traffic counts, for example, identify effects of holidays and weather effects, and incorporate findings into the model components

of activity-based models. However, this indirect approach requires changes of model specifications and specifically works for calibrating traffic counts. In addition, the authors do not discuss the influence of such indirect approaches on other models' results.

The direct approach proposed by [43] to fine-tune activity-based models' parameters can be done at four different levels: the data level, the model level, the OD-matrix level and the assignment level. As the data level, the calibration approach attributes weights to the different agents and generate new activities using updated data. However, the weighting procedure can become very computationally intensive as the number of agents increases. In addition, we cannot use activity-based models to do predictions because models' parameters are not adjusted according to observed statistics.

At the model level, without considering route choice and mode choice, they proposed a heuristic approach to adjust only destination choice-models in the model system and demonstrated the approach on a small 10-zone network. Although they recognized the need to calibrate the complete ABMs considering inter-dependency and admitted the complexity would increase significantly, they did not discuss any solution for such situation.

At the OD-matrix level, we have briefly mentioned common approaches in previous section. The problem of applying OD-matrix estimation approaches on calibration is that forecasting future year matrices needs extra attention. The adjustments performed on the OD-matrix in the base year need to be somehow reflected in forecasted matrices. Absolute difference or percentage of change between simulated OD matrices and adjusted matrices for the base year can be applied to future matrices. Another problem is that OD-matrix estimation approaches cannot be used to calibrate other aggregate statistics like certain activity-purpose share or mode share.

At the traffic assignment level, the way of assigning OD flows to the network plays a important role in matching model-based traffic counts with the benchmark measures. However, calibration approaches at this level do not consider if other measures, like the number of individuals traveling, match or not. It also has the

similar prediction issues because model parameters do not adjust.

[44] elucidated the limitations of existing calibration approaches and proposed using Taguchi experimental design and ANOVA to improve the calibration process. They used Taguchi experimental design to identify the models/parameters that have impact on the outputs in a robust manner: ANOVA was applied to determine the effectiveness of models/parameters on the outputs of interest. However, the methodology to adjust the parameters is a heuristic that relies on the outputs from running the simulation with different level of the parameters which might result in premature solutions and intractable number of simulation runs.

Researchers have stepped away of the direct modelling of all these complex interactions in activity-based model systems. Approaches mentioned in previous section like SPSA are applied to calibrate ABMs. In addition, [45] adopted an iterative black-box methodology for the calibration of POLARIS, assuming to only know the inputs and outputs of the entire modelling process. By using a meta-model of the model system surrogate, the authors evaluated and estimated the unknown relationship between the simulated results and observed outputs at a given input set. A Gaussian Process over all potential functions representing this relationship was determined at each iterative step utilizing Bayesian estimation. Active learning methods are used for sample determination in the exploration phase. However, the authors still point out the computational intractability of the proposed process, as it resulted in long and costly simulations for a single experiment alone.

Flötteröd [46] presents an approach to calibrate both DTA simulators and disaggregate demand models' parameters within a Bayesian framework at individual level. Relying on an agent-based dynamic traffic simulator, the analyst's prior knowledge is represented by the simulator and measurements are time dependent traffic counts. Then priori assumption about every individual's choice distribution is combined with available measurements' likelihood from simulator to form an estimated posterior choice distribution. The authors also show that the framework is capable of handling large scenarios [47]. However, this calibration framework only limit to traffic counts data and rely on dynamic traffic network simulator.

2.3 Summary

In summary, studies on calibration of models for demand generation in transportation model systems have focused primarily on trip-based models. Although, activity-based models are increasingly being adopted in practice, effectively calibrating them has been a challenge because it is complex and computationally intractable. This study tries to address this critical gap by proposing an optimization framework and algorithm to effectively calibrate activity-based model systems.

Chapter 3

Methodology

3.1 Overview

As discussed briefly earlier, the activity-based model system in this paper refers to utility-maximizing nested logit ABMs. In this chapter, we discuss calibrating count based simulated aggregate statistics acquired from simulating ABMs against actual aggregate statistics acquired from travel diary, transit on-board surveys, screenline and other counts. Section 3.2 first introduces a generic ABMs structure and then formulates calibration problem as an optimization problem to minimize the distance between simulated aggregate statistics and actual aggregate statistics. In addition, we propose to use expected simulated aggregate statistics from ABMs outputs as estimates of true simulated aggregate statistics to avoid dealing with uncertainties from activity generation and make objective function value deterministic for given choice-model coefficients. Then we derive simple analytical expressions of simulated aggregate statistics in the objective function that can be evaluated through microsimulation utilizing the structure of ABMs and closed form of logit choice-model's probability expression. Following that, section 3.3 explains the complete solution procedure to solve the optimization problem and explains two important aspects of the solution procedure —gradient computation and parameter update— in detail. Important notations are shown in table 3.1.

Notation	Definition
n	Individual/agent[a] n
$\mathcal{N}$	Full set of individuals simulated (complete population)
$\lvert \cdot \rvert$	Cardinality
Π	Set of all choice-models in the ABMs
$\dot{\Pi}$	Set of choice-models to calibrate in the ABMs, $\dot{\Pi} \subseteq \Pi$
π	Choice-model π, $\pi \in \Pi$
$\mathcal{N}_\pi$	Set of individuals on whom the choice-model π will be run
$\gamma_{\pi n}$	Individual n's choices in upper level choice-models lead to current choice-model π
$\mathcal{C}_\pi$	Choice set of choice-model π
$\bar{S}_{\pi i}$	Expected number of individuals from ABMs choosing choice-model π's alternative i
$S_{\pi i}$	Estimated number of individuals from ABMs choosing choice-model π's alternative i
$A_{\pi i}$	Actual number of individuals choosing choice-model π's alternative i
$x_{\pi i n}$	Vector of individual n's characteristics and attributes of choice-model π's alternative i
$U_{\pi i n}$	Individual n's utility of choice-model π's alternative i
$V_{\pi i n}$	Individual n's systematic utility of choice-model π's alternative i
ϵ_n	Individual n's random utility
$\boldsymbol{V}_{\pi n}$	Individual n's systematic utility vector for choice-model π
$P_n(\pi i)$	Individual n's probability of choosing choice-model π's alternative i
$\lambda_{\pi n}$	Individual n's inclusive value/logsum term of choice-model π
$\beta_{\pi i p}$	Coefficient[b] p of choice-model π's alternative i
$\boldsymbol{\beta}^0$	Vector including apriori values of $\beta_{\pi i p}$ we want to calibrate
$\boldsymbol{\beta}$	Vector including current values of $\beta_{\pi i p}$ we want to calibrate
α	Step size used to update coefficients
$w_{\pi i}$	Weight of choice-model π's component i in objective function
η	Number of mini-batches
$\sigma(\cdot)$	Sigmoid function
$\psi(\cdot)$	Softmax function
$f(\cdot)$	Microsimulation function
$L(\cdot)$	Objective function
$L_\pi(\cdot)$	Component of choice-model π in objective function L
$\nabla_{\boldsymbol{\beta}} L(\cdot)$	Gradients of objective function with respect to $\boldsymbol{\beta}$
$\tilde{\cdot}$	Statistics or objective value associated with mini-batch
κ	Iteration number

Table 3.1: Table of important notation

[a] we use individual and agent interchangeably
[b] we use coefficient and paramter interchangeably

3.2 Problem Definition

3.2.1 Generic ABMs structure

In this section, we define a graph $G = (\Pi, \mathcal{R}, \mathcal{D})$ to present an Activity-Based Model system. Π is a set of nodes and each node, $\pi \in \Pi$, represents a choice-model. $\mathcal{R}$ is a set of ordered levels and each level contains at least one choice-model π. A choice-model π located at level r is denoted as $\pi^{[r]}$ and upper level choice-models have smaller

level number. $\mathcal{D}$ is a set of directed edges, which represents inter-dependency among choice-models.

In addition, $\mathcal{E}$ is an adjacency list with size equal to the number of nodes, $|\Pi|$. An entry $\mathcal{E}[\pi]$ represents the list of upper level choice-models —above than choice-model π— dependent on the choice-model π through logsum term or inclusive value [48] of choice-model π. Logsum term of lower level choice-model makes upper level choice-models sensitive to important attributes that are measured directly only at the lower levels of the model.

Without loss of generality, we assume that any choice-model π_p located at level r as $\pi_p^{[r]}$ in the given ABM system, except the root choice-model ($r \neq 1$), is only directly dependent on a choice-model π_q in one level above, $\pi_q^{[r-1]}$. This means that in the ABMs graph G, there is an unique downward path from the root node to a lower level choice-model $\pi_p^{[r]}$. In other words, there is only one possible choice combination from the root choice-model leading to a lower level choice-model.

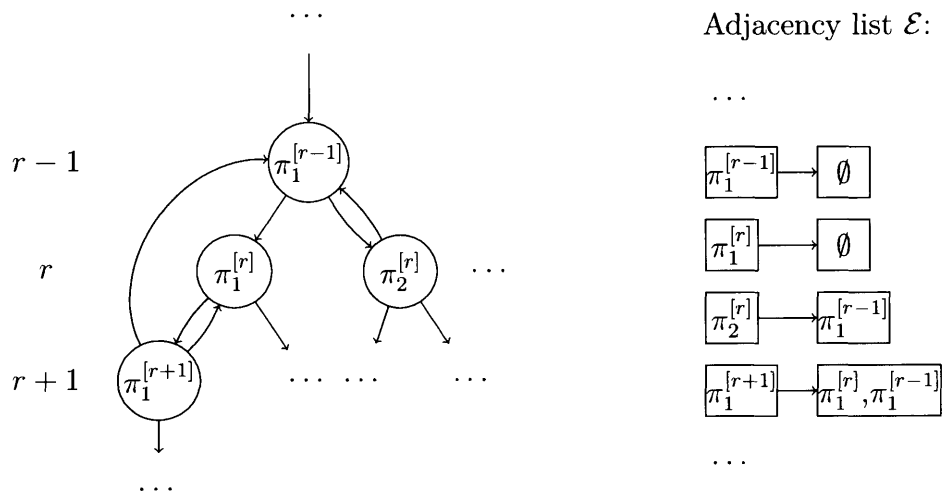


Figure 3-1: A part of Activity-Based Model system graph G. Each circular node represents a choice-model π and each arrow represents the dependency of the end node on the start node. $\Pi = \{\cdots, \pi_1^{[r-1]}, \pi_1^{[r]}, \pi_2^{[r]}, \pi_1^{[r+1]}, \cdots\}$, $\mathcal{R} = \{\cdots, r-1, r, r+1, \cdots\}$

A part of ABMs graph is shown in Figure $3-1$. There are four choice-models located on three different levels as a part of a activity-based model system. The dependency shown in the figure includes the dependency of the lower level choice-

model on the upper level choice-model. Also, it includes the dependency of the upper level choice-model on the lower level choice-model through logsum, which is represented by adjacency list $\mathcal{E}$. For example, an entry $\mathcal{E}[\pi_1^{[r+1]}]$ is a list of two elements: $\pi_1^{[r]}$ and $\pi_1^{[r-1]}$, which represents that upper level choice-models $\pi_1^{[r-1]}$ and $\pi_1^{[r]}$ depend on the choice-model $\pi_1^{[r+1]}$ located on the lower level $r+1$. The following calibration problem formulation and solution procedure illustration are based on these four choice-models. However, they can easily be generalized to calibrate any set of choice-models within a ABM system as we show in section 3.2 and section 3.3.

3.2.2 Problem formulation

To calibrate this generic ABMs, we need the following two categories of variables: 1). actual observed aggregate statistics and 2). simulated aggregate statistics. In this study, we focus on calibrating count based aggregate statistics. The commonly available count based aggregate statistics are the following: number of individuals traveling, number of individuals performing each activity, number of tours corresponding to each activity, number of tours/trips corresponding to each mode, and origin and destination flows at a zonal level. The first four statistics are generally obtained from survey data and expansion factors. The OD flows can generally be obtained from an existing planning model or from an OD estimation procedure performed on the network.

To formulate the problem, let $\dot{\Pi} = \{\pi_1^{[r-1]}, \pi_1^{[r]}, \pi_2^{[r]}, \pi_1^{[r+1]}\}$ represent the set of choice-models needed to be calibrated among all choice-models in the ABMs and let C_π represent the set of alternatives in a choice-model π. Further, let $S_{\pi i}$ denote the estimated number of individuals choosing alternative $i \in C_\pi$ in choice-model π, and $A_{\pi i}$ denote the actual number of alternative $i \in C_\pi$ in choice-model π. Let $\boldsymbol{\beta}_{\pi i}$ represent the coefficients/parameters in the utility expression of alternative $i \in \mathcal{C}_\pi$ in choice-model π. The objective of calibration problem is to determine $\boldsymbol{\beta}_{\pi i}$'s —for the selected choice-models and their alternatives— such that the discrepancies between the simulated aggregate statistics and actual aggregate statistics are minimized. The vector including all $\boldsymbol{\beta}_{\pi i}$'s is denoted as $\boldsymbol{\beta}$. Coefficient p in $\boldsymbol{\beta}_{\pi i}$ is denoted as $\beta_{\pi i p}$.

As we want to minimize the difference between $S_{\pi i}$ and $A_{\pi i}$ for all choice-models in $\dot{\Pi}$ by updating $\boldsymbol{\beta}$ between its lower bound $\boldsymbol{\beta}_{LB}$ and upper bound $\boldsymbol{\beta}_{UB}$, our optimization problem can be written as

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & L(\boldsymbol{\beta}) = \sum_{\pi \in \dot{\Pi}} L_{\pi}(\boldsymbol{\beta}) + L_{o}(\boldsymbol{\beta}) \\ \text{subject to:} \quad & \boldsymbol{\beta}_{LB} \leq \boldsymbol{\beta} \leq \boldsymbol{\beta}_{UB} \end{aligned} \tag{3.1}$$

where we can denote the component of the objective function $L(\boldsymbol{\beta})$ pertaining to a choice-model π as $L_{\pi}(\boldsymbol{\beta})$ and apriori component as $L_{o}(\boldsymbol{\beta})$. These components are expressed below:

$$L_{\pi}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{C}_{\pi}} w_{\pi i} Z(S_{\pi i}(\boldsymbol{\beta}), A_{\pi i}) \quad \forall \pi \in \dot{\Pi} \tag{3.2a}$$

$$L_{o}(\boldsymbol{\beta}) = w_{\boldsymbol{\beta}} Z(\boldsymbol{\beta}^{0}, \boldsymbol{\beta}) \tag{3.2b}$$

where $Z(\cdot)$ is a distance function to calculate the difference between two scalars or two vectors. In the remainder of this paper we use the sum-of-squares distance function: $Z(S_{\pi i}, A_{\pi i})$ is $(S_{\pi i} - A_{\pi i})^2$ and $Z(\boldsymbol{\beta}^{0}, \boldsymbol{\beta})$ is $||\boldsymbol{\beta}^{0} - \boldsymbol{\beta}||^2$. Yet, other differentiable distance functions can be used when appropriate.

We use different weights $w_{\pi,i}$ for various components of objective function as the scale of the errors among different components can be different. For example, if $\pi_1^{[r-1]}$ choice-model has 4 alternatives and $\pi_1^{[r]}$ choice-model only has 1 alternative, the error of $\pi_1^{[r-1]}$ choice-model will effectively be weighted 4 times more than the error of $\pi_1^{[r]}$ choice-model —if we do not have weight parameters. An approach to set these weights is to ensure the following: (i) errors pertaining to choice-models at a given level must have the same scale and (ii) within a given choice-model, different alternatives in the choice-set should have different scales that depends on the number of individuals choosing the corresponding alternatives. The details on how to appropriately set weights are discussed in the following section.

It should be noted that choice-model π's alternative i in objective function $L(\boldsymbol{\beta})$

can also be a combination of several alternatives in choice set C_π. For example, in $\pi_1^{[r-1]}$ choice-model, we can aggregate some alternatives as one option j to determine $S_{\pi_1^{[r-1]}j}$ and calibrate it against $A_{\pi_1^{[r-1]}j}$. This detail will be shown in section 5.1.

For any choice-model π we want to calibrate and include in objective function $L(\boldsymbol{\beta})$, we also need to include all choice-models in the entry $\mathcal{E}[\pi]$ if any. In other words, all choice-models in $\mathcal{E}[\pi], \forall \pi \in \dot{\Pi}$ have to be in $\dot{\Pi}$ too. This is because choice-model π's any alternative i's coefficients $\boldsymbol{\beta}_{\pi i}$ can influence upper level choice-models through the inclusive value from choice-model π. The reason that we do not need to consider any lower level choice-models —below choice-model π— is that lower level choice-models do not contain any terms having $\boldsymbol{\beta}_{\pi,i}$. For example, in the Figure $3-1$, if we only want to calibrate choice-model $\pi_1^{[r+1]}$, we still need to include components related to $\pi_1^{[r-1]}$ and $\pi_1^{[r]}$ in objective function apart from components related to $\pi_1^{[r+1]}$ because these two choice-models in $\mathcal{E}[\pi_1^{[r+1]}]$. Right now, we are interested in calibrating all four choice-models, which do not have inclusive values flow into any upper level choice-model beyond choice-model $\pi_1^{[r-1]}$. Therefore, we do not need to consider any other upper level choice-model when we are calibrating these four.

3.2.3 Evaluating the objective function via microsimulation

Computing the objective function in equation (3.1) entails obtaining simulated aggregate statistics from the entire activity-based model system. This is because lower level choice-models are dependent on upper level choice-models. As choice-models in ABMs calculate probabilities of choosing available alternatives, we have to rely on simulator to generate activity schedules and every simulation run may generate different simulated aggregate statistics $S_{\pi i}$'s. In order to have a deterministic value to compare against actual aggregate statistics, we use the expected number of individuals choosing choice-model π's alternative i ($\bar{S}_{\pi i}$'s) as estimates for $S_{\pi i}$'s.

In this section, we first derive the analytical expressions of $\bar{S}_{\pi i}$'s for evaluating objective value. Then we utilize microsimulation to avoid simulating every choice-model on every individual. In addition, we apply unbiased Hansen-Hurwitz estimator to derive much simpler analytical expressions of $\bar{S}_{\pi i}$'s, which can be evaluated through

microsimulation. We demonstrate that this approach is simple, straightforward, and ideal for calibration algorithm presented in section 3.3.

As an example, consider the expected number of individuals choosing choice-model $\pi_1^{[r-1]}$'s alternative i, $\bar{S}_{\pi_1^{[r-1]}i}$. Let conditional probability $P_n(\pi_1^{[r-1]}i|\gamma_{\pi_1^{[r-1]}n})$ represent individual n's probability of choosing an alternative $i \in \mathcal{C}_{\pi_1^{[r-1]}}$ through the choice-model $\pi_1^{[r-1]}$ conditional on her choices in upper level choice-models leading to $\pi_1^{[r-1]}$ denoted as $\gamma_{\pi_1^{[r-1]}n}$. Also, let the random variable I_n be equal to 1 if individual n chooses choice-model $\pi_1^{[r-1]}$'s alternative i and equal to 0 otherwise. Then, from the Bernoulli random variable's property, the expected value of I_n is equal to $P_n(\pi_1^{[r-1]}i|\gamma_{\pi_1^{[r-1]}n})P_n(\gamma_{\pi_1^{[r-1]}n})$. Subsequently, we can estimate the number of individuals choosing $\pi_1^{[r-1]}$ choice-model's $\pi_1^{[r-1]}$ alternative i by using the following expression

$$S_{\pi_1^{[r-1]}i} \approx \bar{S}_{\pi_1^{[r-1]}i} = E\left[\sum_{n\in\mathcal{N}} I_n\right] = \sum_{n\in\mathcal{N}} P_n(\pi_1^{[r-1]}i|\gamma_{\pi_1^{[r-1]}n})P_n(\gamma_{\pi_1^{[r-1]}n}) \tag{3.3}$$

Unfortunately, the analytical expressions for $\bar{S}_{\pi i}$'s can quickly get complex as we move to lower-level models because we need to consider individual n's all past choices in upper level choice-models, thus complicating the analytical evaluation of the objective function. As an alternative, we can evaluate $S_{\pi i}$'s and the objective function purely through microsimulation: simulate every individual's activity-travel schedule by running the series of choice-models for her. In other words, we simulate all individuals in the population and directly evaluate the objective function after simulation. However, we still need a preferably analytical expression of the objective function so that we can estimate its gradient to effectively minimize it. In this light, we derive analytical expressions for $S_{\pi i}$ in the objective function that are now evaluated through microsimulation. In the paragraphs below we show the derivation of analytical expression for $S_{\pi_1^{[r-1]}i}$.

In microsimulation of nested logit activity-based model system, the sample of individuals at any choice-model is not a simple random sample because each individual is not necessarily simulated through all choice-models. Therefore, the sample mean is

not an unbiased estimator of $\bar{S}_{\pi i}$. For example, $\sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} P_n(\pi_1^{[r-1]} i | \gamma_{\pi_1^{[r-1]}n}) P_n(\gamma_{\pi_1^{[r-1]}n})$ —where $\mathcal{N}_{\pi_1^{[r-1]}}$ represents the set of individuals simulated through choice-model $\pi_1^{[r-1]}$— is not an unbiased estimator of $\bar{S}_{\pi_1^{[r-1]}i}$ as $\mathcal{N}_{\pi_1^{[r-1]}}$ is not a simple random sample from $\mathcal{N}$. The set $\mathcal{N}_{\pi_1^{[r-1]}} \subseteq \mathcal{N}$ depends on the probabilities of individual n choosing $\gamma_{\pi_1^{[r-1]}n}$ in upper level choice-models, $P_n(\gamma_{\pi_1^{[r-1]}n})$. From the Bernoulli random variable's property, we can estimate the number of individuals simulated through the choice-model $\pi_1^{[r-1]}$ by using the following expression

$$|\mathcal{N}_{\pi_1^{[r-1]}}| \approx \bar{S}_{\pi_1^{[r-1]}} = \sum_{n \in \mathcal{N}} P_n(\gamma_{\pi_1^{[r-1]}n}) \tag{3.4}$$

Therefore, we need to estimate $\bar{S}_{\pi_1^{[r-1]}i}$ from an *unequal probability* sample which can be done using Hansen-Hurwitz estimator.

Hansen-Hurwitz estimator is used to determine an unbiased estimate of population total from a sample, when the sample is drawn with replacement according to prespecified sampling probabilities. For the population set $\mathcal{M}$, let ρ_l denote the probability that the population unit $l \in \mathcal{M}$ will be selected and t_l denote the value-of-interest for the population unit $l \in \mathcal{M}$. Hansen-Hurwitz estimator of the population total $\tau = \sum_{l \in \mathcal{M}} t_l$ from a sample $\mathcal{M}'$ is given by $\hat{\tau} = \frac{1}{|\mathcal{M}'|} \sum_{l \in \mathcal{M}'} \frac{t_l}{\rho_l}$.

Hansen-Hurwitz estimator when applied to determine $\bar{S}_{\pi_1^{[r-1]}i}$ with $\frac{P_n(\gamma_{\pi_1^{[r-1]}n})}{\sum_{n \in \mathcal{N}} P_n(\gamma_{\pi_1^{[r-1]}n})}$ being the selection probability of an individual and $\mathcal{N}_{\pi_1^{[r-1]}}$ being the set of individuals selected is

$$\bar{S}_{\pi_1^{[r-1]}i} \approx \hat{S}_{\pi_1^{[r-1]}i} = \frac{1}{|\mathcal{N}_{\pi_1^{[r-1]}}|} \sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} \left[\frac{P_n(\pi_1^{[r-1]} i | \gamma_{\pi_1^{[r-1]}n}) P_n(\gamma_{\pi_1^{[r-1]}n})}{P_n(\gamma_{\pi_1^{[r-1]}n})} \left(\sum_{n \in \mathcal{N}} P_n(\gamma_{\pi_1^{[r-1]}n}) \right) \right] \tag{3.5}$$

From equation (3.4), $\sum_{n \in \mathcal{N}} P_n(\gamma_{\pi_1^{[r-1]}n})$ is $\bar{S}_{\pi_1^{[r-1]}}$, which is the expected of the number of individuals on whom choice-model $\pi_1^{[r-1]}$ will be applied and is approximately equal to the actual number of individuals simulated through the choice-model $\pi_1^{[r-1]}$, $|\mathcal{N}_{\pi_1^{[r-1]}}|$. Thus the above equation (3.5) can be simplified as

$$\bar{S}_{\pi_1^{[r-1]}i} \approx \hat{S}_{\pi_1^{[r-1]}i} = \frac{\bar{S}_{\pi_1^{[r-1]}}}{|\mathcal{N}_{\pi_1^{[r-1]}}|} \sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} P_n(\pi_1^{[r-1]} i | \gamma_{\pi_1^{[r-1]}n})$$

$$\approx \sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} P_n(\pi_1^{[r-1]} i | \gamma_{\pi_1^{[r-1]}n}) \tag{3.6}$$

Then, we can plug in $\hat{S}_{\pi_1^{[r-1]}i}$ into equation 3.1 as $S_{\pi_1^{[r-1]}i}$. Thus, using microsimulation to evaluate $S_{\pi_1^{[r-1]}i}$ does not need to run choice-model $\pi_1^{[r-1]}$ on every individual. Then applying Hansen-Hurwitz estimator, we have a simple analytical expression to calculate $S_{\pi_1^{[r-1]}i}$ —using equation (3.6)— that does not depend on parameters of any choice-model at a higher level than $\pi_1^{[r-1]}$ choice-model.

Using the same arguments as above, we can construct simple analytical expressions for other components in the objective function that is calculated through microsimulation. In other words, the objective function calculation and expression depend only on running the regular microsimulation on ABMs.

3.3 Solution procedure

3.3.1 Description of the solution procedure

The proposed solution procedure is to minimize the objective function $L(\boldsymbol{\beta})$ in equation (3.1) using a mini-batch, gradient-based approach. We chose a gradient-based approach as it is straightforward and provides insights for further algorithmic developments. In addition, using a subset of complete population as mini-batch in every iteration to estimate the gradient provides a good balance between efficiency and robustness of the procedure. Choice-model parameters are now updated after computing the gradient of objective value with respect to a subset of full population. The two important aspects of the solution procedure are as follows: (i) estimating the gradient of the objective function with respect to the parameters using mini-batches and (ii) updating the parameters for the next iteration based on the 'directions' pro-

vided by the gradient evaluations. The pseudo code of calibration algorithm is shown below

Algorithm 1 ABM calibration approach

Input
$\mathcal{N}$ complete population; $\boldsymbol{\beta}$ coefficients want to calibrate
L objective function; α descent step size
err gradient value threshold; $MaxIter$ maximum number of iterations
η number of mini-batches
Output
$\boldsymbol{\beta}^{\kappa}$ calibrated model coefficients
procedure $Calibrate(\mathcal{N}, \boldsymbol{\beta}, L, \alpha, err, MaxIter, \eta)$
$\kappa = 0$
$BatchSize = \frac{|\mathcal{N}|}{\eta}$
$\boldsymbol{\beta}^0 \leftarrow$ vector of initial weights
$\boldsymbol{\beta}^{\kappa} \leftarrow \boldsymbol{\beta}^0$
while $|\nabla_{\boldsymbol{\beta}^{\kappa}} L| > err$ or $\kappa < MaxIter$ **do** ▷ keep looping until $\boldsymbol{\beta}^{\kappa}$ converges
divide $\mathcal{N}$ into η batches
for $j = 1 \ldots \eta$ **do**
run microsimulation $f(\boldsymbol{\beta}^{\kappa})$ on batch j
acquire batch j's simulated statistics $\tilde{S}_{\pi i}(\boldsymbol{\beta}^{\kappa})$ and gradient $\nabla_{\boldsymbol{\beta}^{\kappa}} \tilde{L}(\boldsymbol{\beta}^{\kappa})$
approximate $S_{\pi i}(\boldsymbol{\beta}^{\kappa})$ as $\eta \tilde{S}_{\pi i}(\boldsymbol{\beta}^{\kappa})$ and $\nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa})$ as $\eta^2 \nabla_{\boldsymbol{\beta}^{\kappa}} \tilde{L}(\boldsymbol{\beta}^{\kappa})$ for complete population (refer to section 3.3.4)
calculate search direction d^{κ} using the gradient $\nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa})$
$\boldsymbol{\beta}^{\kappa+1} \leftarrow \boldsymbol{\beta}^{\kappa} + \alpha d^{\kappa}$
$\kappa = \kappa + 1$
end for
end while
return $\boldsymbol{\beta}^{\kappa}$ ▷ The calibrated weight is $\boldsymbol{\beta}^{\kappa}$
end procedure

We estimate the objective function's gradient by calculating the gradients of its components contributed by each individual —using microsimulation— and summing over them. This approach avoids working with complex analytical expressions over the complete population or a purely simulation-based gradient estimation. Further, it helps us efficiently estimate the objective function's gradient to determine the search direction by running the microsimulation on a mini-batch (sample) of the population. The details of the gradient calculation are provided in section 3.3.2.

After calculating gradients, we can apply any of the gradient-based algorithms —

mini-batch gradient-descent, gradient-descent with momentum, Adam, and BFGS—to acquire different search directions with slight modifications. We iteratively update the parameters of interests —until convergence— with the corresponding search direction and an appropriate step size that reduces objective function's value. The details of updating parameters are provided in section 3.3.4.

In the following three subsections, we first discuss gradient calculation at an individual-level and estimating the complete objective function's gradient from individual-level gradients. Then, we present analytical gradient expressions of objective function with respect to any choice-model's parameter we are interested in calibrating. Finally, in parameter update section, we discuss additional modifications on estimating search directions using estimated gradients while updating the parameters using different gradient-based algorithms.

3.3.2 Flow of gradient computation

The gradient of objective function $L(\boldsymbol{\beta})$, in equation (3.1), with respect to a particular parameter $\beta_{\pi ip}$ in $\boldsymbol{\beta}$ can be expressed as

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} = \frac{\partial L_{\pi_1^{[r-1]}}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} + \frac{\partial L_{\pi_1^{[r]}}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} + \frac{\partial L_{\pi_2^{[r]}}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} + \frac{\partial L_{\pi_1^{[r+1]}}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} + \frac{\partial L_o(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} \tag{3.7}$$

Therefore, the gradient of the objective function can be expressed as the sum of gradients of each of its components. We will discuss the gradient of the objective function's components below.

In section 3.2.3, we showed that the estimated number of individuals (from population) choosing choice-model π's alternative i is the sum over each individual's probability of choosing choice-model π's alternative i given that the choice-model was evaluated for her —i.e., she belongs to $\mathcal{N}_\pi$. The equation is shown below:

$$S_{\pi i}(\boldsymbol{\beta}) = \sum_{n \in \mathcal{N}_\pi} P_n(\pi i | \gamma_{\pi n}) \tag{3.8}$$

where $\gamma_{\pi n}$ represents individual n's choices in upper level choice-models leading to current choice-model π.

Note that, for each component of the objective function $L_\pi(\boldsymbol{\beta})$, only $S_{\pi i}(\boldsymbol{\beta}), \forall i \in \mathcal{C}_\pi$ terms depend on $\boldsymbol{\beta}$. Therefore, $\frac{\partial L_\pi(\boldsymbol{\beta})}{\partial \beta_{\pi ip}}$, essentially can be expressed as a function of $\frac{\partial S_{\pi i}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}}$ which —from equation (3.8)— is a function of $\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi ip}}$ for $i \in \mathcal{C}_\pi, n \in \mathcal{N}_\pi$. We explicitly derive this relationship below for 3.2a and 3.2b.

$$\begin{aligned}\frac{\partial L_\pi(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} =& 2\sum_{i\in\mathcal{C}_\pi} w_{\pi i}(S_{\pi i}(\boldsymbol{\beta}) - A_{\pi i})\frac{\partial S_{\pi i}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} \\ =& 2\sum_{i\in\mathcal{C}_\pi} w_{\pi i}(S_{\pi i}(\boldsymbol{\beta}) - A_{\pi i})\sum_{n\in\mathcal{N}_\pi}\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi ip}} \qquad \forall \pi \in \dot{\Pi} \end{aligned} \tag{3.9a}$$

$$\frac{\partial L_o(\boldsymbol{\beta})}{\partial \beta_{\pi ip}} = 2w_{\boldsymbol{\beta}}(\beta^0_{\pi ip} - \beta_{\pi ip}) \tag{3.9b}$$

Thus, from above equations, to calculate the gradient of the objective function, we need to calculate the gradient of $P_n(\pi i|\gamma_{\pi n})$ with respect to $\beta_{\pi ip}$ for each individual $n \in \mathcal{N}_\pi$. We do this by calculating $\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi ip}}$ —details of which are discussed in section 3.3.3— along with $P_n(\pi i|\gamma_{\pi n})$ in the nested logit model system's microsimulation framework. The process is shown as a flowchart in Figure 3-2 for a single individual, assuming in microsimulation she is only going through choice-models $\pi_1^{[r-1]}$, $\pi_1^{[r]}$ and $\pi_1^{[r+1]}$. For each choice-model π, we calculate individual n's conditional probability of choosing choice i, $P_n(\pi i|\gamma_{\pi n})$, and $\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi ip}}$ at the same time.

3.3.3 Derivation of the gradient of the objective function with respect to choice-model specific coefficients

In this section, we use L to represent objective function listed in equation (3.1) and calculate the gradient of L with respect to coefficients $\beta_{\pi ip}$ we are interested in from different selected choice-models. First of all, we can divide choice-models of interest

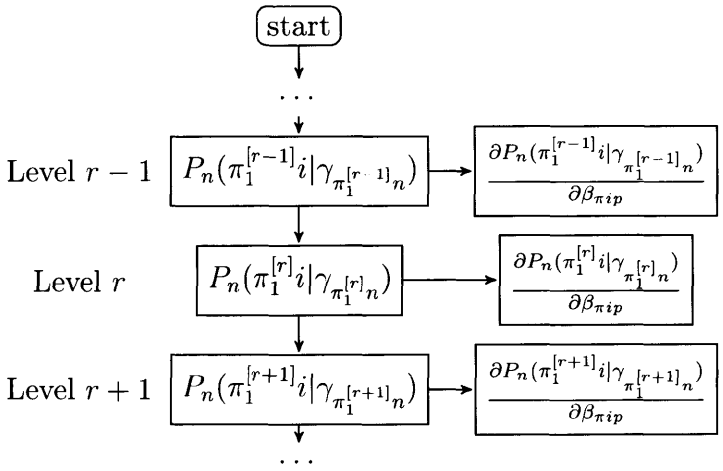


Figure 3-2: Flow of gradient computation. In each choice-model π, we are able to calculate each individual $n \in \mathcal{N}_\pi$'s conditional probability $P_n(\pi i|\gamma_{\pi n})$ and its gradient $\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi ip}}$. According to Hansen-Hurwitz estimator, choice-model π needs to be evaluated only on individuals $n \in \mathcal{N}_\pi$ to evaluate both $S_{\pi i}(\boldsymbol{\beta})$ and $\frac{\partial S_{\pi i}(\boldsymbol{\beta})}{\partial \beta_{\pi ip}}$. Also, each individual n is not necessarily simulated through every choice-model. For example, choice-model $\pi_2^{[r]}$ is not applied to individual n in this plot because we assume that her choice in choice-model $\pi_1^{[r-1]}$ leads to choice-model $\pi_1^{[r]}$. Similarly, she is only subject to certain lower level choice-models according to her selected alternative in choice-model $\pi_1^{[r+1]}$.

$\dot{\Pi}$ into two mutually exclusive sets, $\dot{\Pi}_*$ and $\dot{\Pi}_*^*$ according to adjacency list $\mathcal{E}$, such that $\dot{\Pi}_* \cup \dot{\Pi}_*^* = \dot{\Pi}$. $\dot{\Pi}_*$ is a set of choice-models only having downward linkage to lower level choice-models. In other words, $\dot{\Pi}_* = \{\pi | \mathcal{E}[\pi] = \emptyset, \pi \in \dot{\Pi}\}$. $\dot{\Pi}_*^*$ contains the rest, which are choice-models of interest having upward linkage as logsums to upper level choice-models of interest, $\dot{\Pi}_*^* = \{\pi | \mathcal{E}[\pi] \neq \emptyset, \pi \in \dot{\Pi}\}$. As shown in Figure $3-1$, $\dot{\Pi}_* = \{\pi_1^{[r-1]}, \pi_1^{[r]}\}$ and $\dot{\Pi}_*^* = \{\pi_2^{[r]}, \pi_1^{[r+1]}\}$. In next following subsections, we derive generic gradient expressions of the objective function with respect to coefficients corresponding to choice-models in $\dot{\Pi}_*$ and $\dot{\Pi}_*^*$ separately. Then we present actual analytical gradient expressions with respect to coefficients corresponding to four choice-models of interest shown in Figure 3-1.

However, before that, we introduce notation to represent some functions used extensively in the derivations below —like sigmoid function, $\sigma(\cdot)$, and softmax function, $\psi(\cdot)$. We also present the expressions for their gradient here in order to avoid repetitive notation in following sections.

If z be a real valued scalar input to $\sigma(\cdot)$, the gradient of $\sigma(z)$ with respect to z is:

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z)) \tag{3.10}$$

If $\boldsymbol{z} = [z_1, \ldots z_K]^T$ represents a K-dimensional vector with real valued scalars and $\psi(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$, then the gradient of $\psi(\boldsymbol{z})_j$ with respect to z_k is:

$$\frac{\partial \psi(\boldsymbol{z})_j}{\partial z_k} = \begin{cases} \psi(\boldsymbol{z})_j(1 - \psi(\boldsymbol{z})_j), & \text{if } j = k \\ -\psi(\boldsymbol{z})_j \psi(\boldsymbol{z})_k, & \text{otherwise} \end{cases} \tag{3.11}$$

In addition, if $\lambda_{\pi n} = \log(\sum_{i \in \mathcal{C}_\pi} e^{V_{\pi in}})$ represents the logsum term of choice-model π for individual n, where $V_{\pi in}$ represents systematic utility of choice-model π's alternative i for individual n. Then the gradient of $\lambda_{\pi n}$ with respect to coefficient p of any choice-model π's alternative i, $\beta_{\pi ip}$ is:

$$\frac{\partial \lambda_{\pi n}}{\partial \beta_{\pi ip}} = \frac{\sum_{j \in \mathcal{C}_\pi} e^{V_{\pi jn}} \frac{\partial V_{\pi jn}}{\partial \beta_{\pi ip}}}{\sum_{i \in \mathcal{C}_\pi} e^{V_{\pi in}}} \tag{3.12}$$

where $\frac{\partial V_{\pi jn}}{\partial \beta_{\pi ip}}$ can be easily calculated according to actual systematic utility specification of choice-model π's alternative j.

Finally, we can calculate the gradient of apriori component L_o in objective function L with respect to coefficient p of choice-model π's alternative i, $\beta_{\pi ip}$ as.

$$\frac{\partial L_o}{\partial \beta_{\pi ip}} = 2w_\beta(\beta^0_{\pi ip} - \beta_{\pi ip}) \tag{3.13}$$

Generic gradient with respect to coefficients of choice-model $\pi_* \in \dot{\Pi}_*$

We now derive the expression for the gradient of the objective function L with respect to coefficient p of choice-model π_*'s alternative i ($\pi_* \in \dot{\Pi}_*$), represented as $\beta_{\pi_* ip}$. From equation (3.7), this expression can be written as follows

$$\frac{\partial L}{\partial \beta_{\pi_* ip}} = \frac{\partial L_{\pi_*}}{\partial \beta_{\pi_* ip}} + \frac{\partial L_o}{\partial \beta_{\pi_* ip}} \tag{3.14}$$

As all lower level choice-models below π_* do not contain any terms related to $\beta_{\pi_* ip}$, the gradients of components of these choice-models with respect to $\beta_{\pi_* ip}$ are 0. Also, all upper level choice-models above π_* do not contain any longsum term related to $\beta_{\pi_* ip}$, the gradients of components of these choice-models with respect to $\beta_{\pi_* ip}$ are 0. As we have already calculated $\frac{\partial L_o}{\partial \beta_{\pi_* ip}}$ in equation (3.9b) we only need to calculate $\frac{\partial L_{\pi_*}}{\partial \beta_{\pi_* ip}}$. From equation 3.9$a$, the expression $\frac{\partial L_{\pi_*}}{\partial \beta_{\pi_* ip}}$ can be written as

$$\frac{\partial L_{\pi_*}}{\partial \beta_{\pi_* ip}} = 2 \sum_{i \in} w_{\pi_* i} (S_{\pi_* i}(\boldsymbol{\beta}) - A_{\pi_* i}) \sum_{n \in \mathcal{N}_{\pi_*}} \frac{\partial P_n(\pi_* i | \gamma_{\pi_* n})}{\partial \beta_{\pi_* ip}} \tag{3.15}$$

As choice-model π_* is a logit model, the probability of individual n choosing alternative i is

$$P_n(\pi_* i | \gamma_{\pi_* n}) = \frac{e^{V_{\pi_* in}}}{\sum_{i \in \mathcal{C}_{\pi_*}} e^{V_{\pi_* in}}} = \psi(\boldsymbol{V}_{\pi_* n})_i \tag{3.16}$$

where $\boldsymbol{V}_{\pi_* n} = [V_{\pi_* 1n}, \ldots, V_{\pi_* in}, \ldots]^T$ represents a $|\mathcal{C}_{\pi_*}|$-dimensional systematic utility vector for individual n and $V_{\pi_* in}$ denotes the systematic utility of alternative i in the choice-model π_* for individual n. Subsequently, we can write

$$\frac{\partial P_n(\pi_* i | \gamma_{\pi_* n})}{\partial \beta_{\pi_* ip}} = \sum_{j \in \mathcal{C}_{\pi_*}} \frac{\partial \psi(\boldsymbol{V}_{\pi_* n})_i}{\partial V_{\pi_* jn}} \frac{\partial V_{\pi_* jn}}{\partial \beta_{\pi_* ip}} \tag{3.17}$$

where $\frac{\partial \psi(\boldsymbol{V}_{\pi_* n})_i}{\partial V_{\pi_* jn}}$ is expressed in equation (3.11). $\frac{\partial V_{\pi_* jn}}{\partial \beta_{\pi_* ip}}$ can be calculated easily according to the actual systematic utility specification of choice-model π_*.

Finally, by combining the expressions in equations (3.14),(3.16) and (3.17) we have

$$\frac{\partial L}{\partial \beta_{\pi_* ip}} = 2 \sum_{i \in \mathcal{C}_{\pi_*}} w_{\pi_* i} (S_{\pi_* i}(\boldsymbol{\beta}) - A_{\pi_* i}) \sum_{n \in \mathcal{N}_{\pi_*}} \sum_{j \in \mathcal{C}_{\pi_*}} \frac{\partial \psi(\boldsymbol{V}_{\pi_* n})_i}{\partial V_{\pi_* jn}} \frac{\partial V_{\pi_* jn}}{\partial \beta_{\pi_* ip}}$$
$$+2 w_{\boldsymbol{\beta}} (\beta^0_{\pi_* ip} - \beta_{\pi_* ip}) \tag{3.18}$$

Generic gradient with respect to coefficients of choice-model $\pi^*_* \in \dot{\Pi}^*_*$

From equation (3.7), the gradient of objective function L with respect to coefficient p of choice-model π^*_*'s alternative i, $\beta_{\pi^*_* ip}$ can be written as

$$\frac{\partial L}{\partial \beta_{\pi^*_* ip}} = \sum_{\pi \in \mathcal{E}[\pi^*_*]} \left(\frac{\partial L_\pi}{\partial \beta_{\pi^*_* ip}} \right) + \frac{\partial L_{\pi^*_*}}{\partial \beta_{\pi^*_* ip}} + \frac{\partial L_o}{\partial \beta_{\pi^*_* ip}} \tag{3.19}$$

Note that all lower level choice-models below π^*_* do not contain any terms related to $\beta_{\pi^*_* ip}$, the gradients of components of these choice-models with respect to $\beta_{\pi^*_* ip}$ are 0. However, there are some upper level choice-models $\{\pi | \pi \in \mathcal{E}[\pi^*_*]\}$ containing logsum related to $\beta_{\pi^*_* ip}$, the gradients of components of these choice-models with respect to $\beta_{\pi^*_* ip}$ are not equal to 0. The expression $\frac{\partial L_o}{\partial \beta_{\pi^*_* ip}}$ can be calculated as shown in equation (3.9b). From equations (3.9a), the expressions $\frac{\partial L_\pi}{\partial \beta_{\pi^*_* ip}}$ and $\frac{\partial L_{\pi^*_*}}{\partial \beta_{\pi^*_* ip}}$ can be written as

$$\frac{\partial L_\pi}{\partial \beta_{\pi^*_* ip}} = 2 \sum_{i \in \mathcal{C}_\pi} w_{\pi,i} (S_{\pi,i}(\boldsymbol{\beta}) - A_{\pi,i}) \sum_{n \in \mathcal{N}_\pi} \frac{\partial P_n(\pi i | \gamma_{\pi n})}{\partial \beta_{\pi^*_* ip}} \quad \forall \pi \in \mathcal{E}[\pi^*_*] \tag{3.20a}$$

$$\frac{\partial L_{\pi^*_*}}{\partial \beta_{\pi^*_* ip}} = 2 \sum_{i \in \mathcal{C}_{\pi^*_*}} w_{\pi^*_* i} (S_{\pi^*_* i}(\boldsymbol{\beta}) - A_{\pi^*_* i}) \sum_{n \in \mathcal{N}_{\pi^*_*}} \frac{\partial P_n(\pi^*_* i | \gamma_{\pi^*_* n})}{\partial \beta_{\pi^*_* ip}} \tag{3.20b}$$

Thus, to compute the above expressions, we need to derive $\frac{\partial P_n(\pi i | \gamma_{\pi n})}{\partial \beta_{\pi^*_* ip}}$ and $\frac{\partial P_n(\pi^*_* i | \gamma_{\pi^*_* n})}{\partial \beta_{\pi^*_* ip}}$ separately.

As choice-model π^*_* is a logit model, the probability of individual n choosing

alternative i is

$$P_n(\pi_*^* i|\gamma_{\pi_*^* n}) = \frac{e^{V_{\pi_*^* in}}}{\sum_{i\in\mathcal{C}_{\pi_*^*}} e^{V_{\pi_*^* in}}} = \psi(\boldsymbol{V}_{\pi_*^* n})_i \tag{3.21}$$

where $\boldsymbol{V}_{\pi_*^* n} = \left[V_{\pi_*^* 1n}, \ldots, V_{\pi_*^* in}, \ldots\right]^T$ represents a $|\mathcal{C}_{\pi_*^*}|$-dimensional systematic utility vector and $V(\pi_*^* in)$ denotes the systematic utility of alternative i in the choice-model π_*^* for individual n. Subsequently, we can write

$$\frac{\partial P_n(\pi_*^* i|\gamma_{\pi_*^* n})}{\partial \beta_{\pi_*^* ip}} = \sum_{j\in\mathcal{C}_{\pi_*^*}} \frac{\partial \psi(\boldsymbol{V}_{\pi_*^* n})_i}{\partial V_{\pi_*^* jn}} \frac{\partial V_{\pi_*^* jn}}{\partial \beta_{\pi_*^* ip}} \tag{3.22}$$

where $\frac{\partial \psi(\boldsymbol{V}_{\pi_*^* n})_i}{\partial V_{\pi_*^* jn}}$ is expressed in equation (3.11). $\frac{\partial V_{\pi_*^* jn}}{\partial \beta_{\pi_*^* ip}}$ can be calculated easily according to the actual systematic utility specification of choice-model π_*^*.

As choice-model π is a logit model, the expression for $\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi_*^* ip}}$ is

$$\frac{\partial P_n(\pi i|\gamma_{\pi n})}{\partial \beta_{\pi_*^* ip}} = \sum_{j\in\mathcal{C}_\pi} \frac{\partial \psi(\boldsymbol{V}_{\pi,n})_i}{\partial V_{\pi jn}} \frac{\partial V_{\pi jn}}{\partial \beta_{\pi_*^* ip}} \qquad \forall \pi \in \mathcal{E}[\pi_*^*] \tag{3.23}$$

where $\frac{\partial \psi(\boldsymbol{V}_{\pi,n})_i}{\partial V_{\pi jn}}$ can be calculated using equation (3.11). The term $\frac{\partial V_{\pi jn}}{\partial \beta_{\pi_*^* ip}}$ in the above equation is not equal to zero as $V_{\pi jn}$ contains a inclusive value (logsum) term from choice-model π_*^*, $\lambda_{\pi_*^* n}$ —defined as $\log(\sum_{i\in\mathcal{C}_{\pi_*^*}} e^{V_{\pi_*^* in}})$— which is a function of $\beta_{\pi_*^* ip}$. As a result, $\frac{\partial V_{\pi jn}}{\partial \beta_{\pi_*^* ip}}$ can be calculated easily according to the actual systematic utility specification of choice-model π and equation 3.12.

Finally, by combining the expressions in equations (3.19), (3.20), (3.22) and (3.23) we have

$$\begin{aligned}\frac{\partial L}{\partial \beta_{\pi_*^* ip}} = & \sum_{\pi\in\mathcal{E}[\pi_*^*]} \left(2 \sum_{i\in\mathcal{C}_\pi} w_{\pi,i}(S_{\pi,i}(\boldsymbol{\beta}) - A_{\pi,i}) \sum_{n\in\mathcal{N}_\pi} \sum_{j\in\mathcal{C}_\pi} \frac{\partial \psi(\boldsymbol{V}_{\pi,n})_i}{\partial V_{\pi jn}} \frac{\partial V_{\pi jn}}{\partial \beta_{\pi_*^* ip}} \right) \\ & +2 \sum_{i\in\mathcal{C}_{\pi_*^*}} w_{\pi_*^* i}(S_{\pi_*^* i}(\boldsymbol{\beta}) - A_{\pi_*^* i}) \sum_{n\in\mathcal{N}_{\pi_*^*}} \sum_{j\in\mathcal{C}_{\pi_*^*}} \frac{\partial \psi(\boldsymbol{V}_{\pi_*^* n})_i}{\partial V_{\pi_*^* jn}} \frac{\partial V_{\pi_*^* jn}}{\partial \beta_{\pi_*^* ip}} \\ & +2 w_{\boldsymbol{\beta}}(\beta^0_{\pi_*^* ip} - \beta_{\pi_*^* ip})\end{aligned} \tag{3.24}$$

Actual gradient with respect to coefficients of choice-model $\pi \in \dot{\Pi}$

Now we are able to apply generic analytical expressions we derived to get analytical expressions of gradient with respect to coefficients of choice-model we want to calibrate in $\dot{\Pi}$.

Among choice-models $\pi \in \dot{\Pi}$, choice-model $\pi_1^{[r-1]}$ and choice-model $\pi_1^{[r]}$ are in the set $\dot{\Pi}_*$. According to the expression in equation (3.18), $\frac{\partial L}{\partial \beta_{\pi_1^{[r-1]}ip}}$ can be written as

$$\frac{\partial L}{\partial \beta_{\pi_1^{[r-1]}ip}} = 2 \sum_{i \in \mathcal{C}_{\pi_1^{[r-1]}}} w_{\pi_1^{[r-1]}i} (S_{\pi_1^{[r-1]}i}(\boldsymbol{\beta}) - A_{\pi_1^{[r-1]}i}) \sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} \sum_{j \in \mathcal{C}_{\pi_1^{[r-1]}}} \frac{\partial \psi(\boldsymbol{V}_{\pi_1^{[r-1]}n})_i}{\partial V_{\pi_1^{[r-1]}jn}} \frac{\partial V_{\pi_1^{[r-1]}jn}}{\partial \beta_{\pi_1^{[r-1]}ip}}$$
$$+2w_{\boldsymbol{\beta}}(\beta^0_{\pi_1^{[r-1]}ip} - \beta_{\pi_1^{[r-1]}ip}) \tag{3.25}$$

$\frac{\partial L}{\partial \beta_{\pi_1^{[r]}ip}}$ can be written as

$$\frac{\partial L}{\partial \beta_{\pi_1^{[r]}ip}} = 2 \sum_{i \in \mathcal{C}_{\pi_1^{[r]}}} w_{\pi_1^{[r]}i} (S_{\pi_1^{[r]}i}(\boldsymbol{\beta}) - A_{\pi_1^{[r]}i}) \sum_{n \in \mathcal{N}_{\pi_1^{[r]}}} \sum_{j \in \mathcal{C}_{\pi_1^{[r]}}} \frac{\partial \psi(\boldsymbol{V}_{\pi_1^{[r]}n})_i}{\partial V(\pi_1^{[r]}jn)} \frac{\partial V(\pi_1^{[r]}jn)}{\partial \beta_{\pi_1^{[r]}ip}}$$
$$+ 2w_{\boldsymbol{\beta}}(\beta^0_{\pi_1^{[r]}ip} - \beta_{\pi_1^{[r]}ip}) \tag{3.26}$$

Among choice-models $\pi \in \dot{\Pi}^*_*$, choice-model $\pi_2^{[r]}$ and choice-model $\pi_1^{[r+1]}$ are in the set $\dot{\Pi}^*_*$. According to the expression in equation (3.24), $\frac{\partial L}{\partial \beta_{\pi_2^{[r]}ip}}$ can be written as

$$\frac{\partial L}{\partial \beta_{\pi_2^{[r]}ip}} = \frac{\partial L_{\pi_1^{[r-1]}}}{\partial \beta_{\pi_2^{[r]}ip}} + \frac{\partial L_{\pi_2^{[r]}}}{\partial \beta_{\pi_2^{[r]}ip}} + \frac{\partial L_o}{\partial \beta_{\pi_2^{[r]}ip}} \tag{3.27}$$
$$= 2 \sum_{i \in \mathcal{C}_{\pi_1^{[r-1]}}} w_{\pi_1^{[r-1]}i} (S_{\pi_1^{[r-1]}i}(\boldsymbol{\beta}) - A_{\pi_1^{[r-1]}i}) \sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} \sum_{j \in \mathcal{C}_{\pi_1^{[r-1]}}} \frac{\partial \psi(\boldsymbol{V}_{\pi_1^{[r-1]},n})_i}{\partial V_{\pi_1^{[r-1]}jn}} \frac{\partial V_{\pi_1^{[r-1]}jn}}{\partial \beta_{\pi_2^{[r]}ip}}$$
$$+2 \sum_{i \in \mathcal{C}_{\pi_2^{[r]}}} w_{\pi_2^{[r]},i} (S_{\pi_2^{[r]},i}(\boldsymbol{\beta}) - A_{\pi_2^{[r]},i}) \sum_{n \in \mathcal{N}_{\pi_2^{[r]}}} \sum_{j \in \mathcal{C}_{\pi_2^{[r]}}} \frac{\partial \psi(\boldsymbol{V}_n(\pi_2^{[r]}))_i}{\partial V_{\pi_2^{[r]}jn}} \frac{\partial V_{\pi_2^{[r]}jn}}{\partial \beta_{\pi_2^{[r]}ip}}$$
$$+2w_{\boldsymbol{\beta}}(\beta^0_{\pi_2^{[r]}ip} - \beta_{\pi_2^{[r]}ip}) \tag{3.28}$$

$\frac{\partial L}{\partial \beta_{\pi_1^{[r+1]}ip}}$ can be written as

$$\frac{\partial L}{\partial \beta_{\pi_1^{[r+1]}ip}} = \frac{\partial L_{\pi_1^{[r-1]}}}{\partial \beta_{\pi_1^{[r+1]}ip}} + \frac{\partial L_{\pi_1^{[r]}}}{\partial \beta_{\pi_1^{[r+1]}ip}} + \frac{\partial L_{\pi_1^{[r+1]}}}{\partial \beta_{\pi_1^{[r+1]}ip}} + \frac{\partial L_o}{\partial \beta_{\pi_1^{[r]}ip}} \tag{3.29}$$

$$\begin{aligned}
&= 2 \sum_{i \in \mathcal{C}_{\pi_1^{[r-1]}}} w_{\pi_1^{[r-1]}i} (S_{\pi_1^{[r-1]}i}(\boldsymbol{\beta}) - A_{\pi_1^{[r-1]}i}) \sum_{n \in \mathcal{N}_{\pi_1^{[r-1]}}} \sum_{j \in \mathcal{C}_{\pi_1^{[r-1]}}} \frac{\partial \psi(\boldsymbol{V}_{\pi_1^{[r-1]},n})_i}{\partial V_{\pi_1^{[r-1]}jn}} \frac{\partial V_{\pi_1^{[r-1]}jn}}{\partial \beta_{\pi_1^{[r+1]}ip}} \\
&+2 \sum_{i \in \mathcal{C}_{\pi_1^{[r]}}} w_{\pi_1^{[r]}i} (S_{\pi_1^{[r]}i}(\boldsymbol{\beta}) - A_{\pi_1^{[r]}i}) \sum_{n \in \mathcal{N}_{\pi_1^{[r]}}} \sum_{j \in \mathcal{C}_{\pi_1^{[r]}}} \frac{\partial \psi(\boldsymbol{V}_n(\pi_1^{[r]}))_i}{\partial V_{\pi_1^{[r]}jn}} \frac{\partial V_{\pi_1^{[r]}jn}}{\partial \beta_{\pi_1^{[r+1]}ip}} \\
&+2 \sum_{i \in \mathcal{C}_{\pi_1^{[r+1]}}} w_{\pi_1^{[r+1]}i} (S_{\pi_1^{[r+1]}i}(\boldsymbol{\beta}) - A_{\pi_1^{[r+1]}i}) \sum_{n \in \mathcal{N}_{\pi_1^{[r+1]}}} \sum_{j \in \mathcal{C}_{\pi_1^{[r+1]}}} \frac{\partial \psi(\boldsymbol{V}_{\pi_1^{[r+1]},n})_i}{\partial V_{\pi_1^{[r+1]}jn}} \frac{\partial V_{\pi_1^{[r+1]}jn}}{\partial \beta_{\pi_1^{[r+1]}ip}} \\
&+2 w_{\boldsymbol{\beta}} (\beta^0_{\pi_1^{[r+1]}ip} - \beta_{\pi_1^{[r+1]}ip})
\end{aligned} \tag{3.30}$$

3.3.4 Updating the parameters

After estimating the gradient of objective function —as detailed above— we can directly use it to update the parameter using any of the gradient-based algorithms. For example, in the straightforward gradient-descent algorithm, the estimated gradient will form the search direction and that is combined with an appropriate step size to iteratively update the coefficients $\boldsymbol{\beta}$, until convergence. However, estimating the gradient by running the simulation on the complete synthetic population might be computationally intensive. As an alternative and is a trade-off between robustness and efficiency, we use mini-batch approach which uses a subset of the full population to estimate gradient and facilitates faster improvement of the coefficients.

Mini-batch gradient estimation

To apply the mini-batch algorithms, we need to modify the objective function and its gradient so that the estimates from mini-batches are representative of the actual objective function and its gradient.

Let η denote the number of disjoint mini-batches that comprise the population. Also, let $\tilde{A}_{\pi i}$ represent the actual number of individuals choosing choice-model π's al-

ternative i in the mini-batch. As mini-batches are constructed via random sampling, we can estimate $\tilde{A}_{\pi i}$ as $\frac{A_{\pi i}}{\eta}$. Using a similar argument, we can estimate the $S_{\pi i}(\boldsymbol{\beta})$ of the full population from $\tilde{S}_{\pi i}(\boldsymbol{\beta})$ — which represents simulated number of individuals choosing choice-model π's alternative i in the mini-batch— as $\tilde{S}_{\pi i}(\boldsymbol{\beta})\eta$. Thus, the objective function of the mini-batch $\tilde{L}(\boldsymbol{\beta})$ can be related to the objective function of the complete population $L(\boldsymbol{\beta})$. As an example, consider $\pi_1^{[r-1]}$ choice-model component of the mini-batch objective function, $\tilde{L}(\boldsymbol{\beta})$

$$
\begin{aligned}
\tilde{L}_{\pi_1^{[r-1]}}(\boldsymbol{\beta}) &= \sum_{i \in \mathcal{C}_{\pi_1^{[r-1]}}} \tilde{w}_{\pi_1^{[r-1]}{}_i} (\tilde{S}_{\pi_1^{[r-1]}{}_{,i}}(\boldsymbol{\beta}) - \tilde{A}_{\pi_1^{[r-1]}{}_{,i}})^2 \\
&\approx \sum_{i \in \mathcal{C}_{\pi_1^{[r-1]}}} w_{\pi_1^{[r-1]}{}_i} (\frac{S_{\pi_1^{[r-1]}{}_i}(\boldsymbol{\beta})}{\eta} - \frac{A_{\pi_1^{[r-1]}{}_i}}{\eta})^2 \\
&= \sum_{i \in \mathcal{C}_{\pi_1^{[r-1]}}} \frac{w_{\pi_1^{[r-1]}{}_i}}{\eta^2} (S_{\pi_1^{[r-1]}{}_i}(\boldsymbol{\beta}) - A_{\pi_1^{[r-1]}{}_i})^2 \\
&= \frac{L_{\pi_1^{[r-1]}}(\boldsymbol{\beta})}{\eta^2}
\end{aligned}
$$

where $\tilde{w}_{\pi_1^{[r-1]}{}_i}$ is equal to $w_{\pi_1^{[r-1]}{}_i}$ because of random sampling. As a consequence, the objective function and its gradient for a mini-batch can be written in terms of the population objective function as follows

$$L(\boldsymbol{\beta}) \approx \tilde{L}(\boldsymbol{\beta})\eta^2 \tag{3.31a}$$

$$\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) \approx \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta})\eta^2 \tag{3.31b}$$

Therefore, by scaling gradient from a mini-batch by η^2 we can estimate the complete objective function's gradient and appropriately update the coefficients $\boldsymbol{\beta}$. All the discussions that follow pertain to the mini-batch approach presented here. In the reminder of this section, we discuss some of the specific parameter update schemes —in addition to the standard gradient descent— that were used as the part of the

study.

Gradient descent with momentum (GM)

Gradient descent with momentum [49] is a method that usually provides faster convergence by accelarating gradients in the descent direction. It calculates descent search direction as an exponentially moving average of previous search directions, $m^{\kappa-1}$, and current gradients, $\nabla_{\boldsymbol{\beta}^\kappa} L(\boldsymbol{\beta}^\kappa)$. as follows.

$$m^\kappa = \theta_1 * m^{\kappa-1} + (1-\theta_1) * \nabla_{\boldsymbol{\beta}^\kappa} L(\boldsymbol{\beta}^\kappa) \tag{3.32}$$

where m^0 is initialized as (vectors of) 0's. Since there are not enough values to average over in the first several iterations, we need to adopt the bias correction as follows:

$$\hat{m}^\kappa = \frac{m^\kappa}{1-\theta_1^\kappa} \tag{3.33}$$

Then, we would use bias-corrected first moment estimates, $\tilde{m}^\kappa$, as search direction of current iteration κ and update coefficients $\boldsymbol{\beta}$ with an appropriate step size, α_{GM}. As a result, the parameter update equation can be written as

$$\boldsymbol{\beta}^{\kappa+1} = \boldsymbol{\beta}^\kappa - \alpha_{GM} \tilde{m}^\kappa \tag{3.34}$$

Adam

Adam [50] computes individual adaptive learning rates for different parameters using the estimates of first and second moments of the gradients. The first and second moments estimates are initialized as (vectors of) 0's; which are then updated using exponential moving averages of the gradients $\nabla_{\boldsymbol{\beta}^\kappa} L(\boldsymbol{\beta}^\kappa)$ and the squared gradients $\nabla_{\boldsymbol{\beta}^\kappa} L^2(\boldsymbol{\beta}^\kappa)$ (elementwise square). As before, we need to correct for the initialization

bias. The complete set of update equations at iteration κ are shown below.

$$m^{\kappa} = \theta_1 * m^{\kappa-1} + (1-\theta_1) * \nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa}) \tag{3.35a}$$

$$g^{\kappa} = \theta_2 * g^{\kappa-1} + (1-\theta_2) * \nabla_{\boldsymbol{\beta}^{\kappa}} L^2(\boldsymbol{\beta}^{\kappa}) \tag{3.35b}$$

$$\hat{m}^{\kappa} = \frac{m^{\kappa}}{1-\theta_1^{\kappa}} \tag{3.35c}$$

$$\hat{g}^{\kappa} = \frac{g^{\kappa}}{1-\theta_2^{\kappa}} \tag{3.35d}$$

The search direction of Adam at iteration κ is $\frac{\hat{m}^{\kappa}}{\sqrt{\hat{g}^{\kappa}}+\delta}$ and δ is a very small number (usually set at 10^{-8}). Finally, we update coefficients $\boldsymbol{\beta}$ using the above search direction and an appropriate step size, α_{adam} as

$$\boldsymbol{\beta}^{\kappa+1} = \boldsymbol{\beta}^{\kappa} - \alpha_{adam} \frac{\hat{m}^{\kappa}}{\sqrt{\hat{g}^{\kappa}} + \delta} \tag{3.36}$$

BFGS

Broyden-Fletcher-Goldfarb-Shanno (BFGS) [51] algorithm is a quasi Newton method, which approximates the Hessian matrix B using updates specified by gradient evaluations. Given an approximate the inverse of Hessian matrix B^{κ} at iteration κ and gradients $\nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa})$, we can obtain search direction as $-B^{\kappa}\nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa})$ and update the coefficients $\boldsymbol{\beta}$ with an appropriate step size, α_{BFGS}. The approximate inverse of the Hessian matrix B is initialized as an identity matrix. Updating equations of the inverse of approximate Hessian matrix at iteration κ are shown below.

$$s^{\kappa} = \boldsymbol{\beta}^{\kappa+1} - \boldsymbol{\beta}^{\kappa} \tag{3.37a}$$

$$y^{\kappa} = \nabla_{\boldsymbol{\beta}^{\kappa+1}} L(\boldsymbol{\beta}^{\kappa+1}) - \nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa}) \tag{3.37b}$$

$$B^{\kappa+1} = B^{\kappa} + \frac{(s^{\kappa\top} y^{\kappa} + y^{\kappa\top} B^{\kappa} y^{\kappa})(s^{\kappa} s^{\kappa\top})}{(s^{\kappa\top} y^{\kappa})^2} - \frac{B^{\kappa} y^{\kappa} s^{\kappa\top} + s^{\kappa} y^{\kappa\top} B^{\kappa}}{s^{\kappa\top} y^{\kappa}} \tag{3.37c}$$

As a result, the update equation in iteration κ can be written as

$$\boldsymbol{\beta}^{\kappa+1} = \boldsymbol{\beta}^{\kappa} + \alpha_{BFGS} B^{\kappa} \nabla_{\boldsymbol{\beta}^{\kappa}} L(\boldsymbol{\beta}^{\kappa}) \tag{3.38}$$

Chapter 4

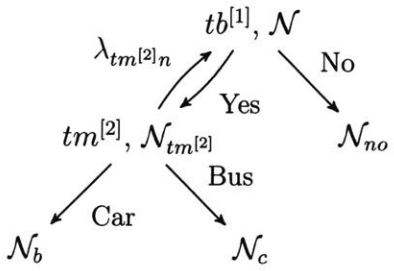
Illustrative example

4.1 Overview

In this section, we present the following small illustrative example to provide a work-through of proposed solution procedure. Figure 4-1 presents the simplified model system's structure. In this example, we only include two choice-models at two levels: travel binary choice-model ($tb^{[1]}$) at level 1 and travel mode choice-model ($tm^{[2]}$) at level 2. Choice-model $tb^{[1]}$ models if an individual travels or not; choice-model $tm^{[2]}$ models if an individual travels by bus or by car conditional on her travelling. In addition, lower level choice-model $tm^{[2]}$ influence upper level choice-model $tb^{[1]}$'s choice through logsum.

As we did in section 3.2, we can construct a ABM graph $G = (\Pi, \mathcal{R}, \mathcal{D})$ for this iilustrative example. Choice-model set $\Pi = \{tb^{[1]}, tm^{[2]}\}$; level set $\mathcal{R} = \{1, 2\}$; Directed edge set $\mathcal{D} = \{(tb^{[1]}, tm^{[2]}), (tm^{[2]}, tb^{[1]})\}$. In addition, we can have an adjacency list $\mathcal{E}$ such that $\mathcal{E}[tb^{[1]}] = \{\emptyset\}$ and $\mathcal{E}[tm^{[2]}] = \{tb^{[1]}\}$ because there is a logsum term from choice-model $tm^{[2]}$ flows into choice-model $tb^{[1]}$. Since we are going to calibrate both two choice-models, choice-model of interest set $\dot{\Pi} = \Pi$. As we mentioned in section 3.3, we can have two mutually exclusive sets: set of choice-models without logsum flowing out $\dot{\Pi}_* = \{tb^{[1]}\}$ and set of choice-models with logsum flowing out $\dot{\Pi}^*_* = \{tm^{[2]}\}$.

The complete population set is $\mathcal{N}$ and the population size, which is also the



(a) Model system's structure

Coefficient	Initial value ($\beta^0_{\pi p}$)
$\beta_{tb^{[1]}y}$	4
$\beta_{tb^{[1]}\lambda_{tm^{[2]}}}$	0.3
$\beta_{tm^{[2]}ctt}$	-0.3
$\beta_{tm^{[2]}ccost}$	-1
$\beta_{tm^{[2]}b}$	-1
$\beta_{tm^{[2]}btt}$	-0.5
$\beta_{tm^{[2]}bcost}$	-1

(b) Coefficient values

Figure 4-1: Illustrative example. We have two choice-models: $tb^{[1]}$ and $tm^{[2]}$, and only those individuals choose traveling go into $tm^{[2]}$ choice-model. In each choice-model, we can calculate individual n's (conditional) probability of choosing alternative i and its gradient with respect to coefficients. According to Hansen-Hurwitz estimator, the estimated number of individuals traveling by bus is $\sum_{n\in\mathcal{N}_{tm^{[2]}}} P_n(tm^{[2]}b|\gamma_{tb^{[1]}n})$.

number of individuals on whom choice-model $tb^{[1]}$ is applied, is $|\mathcal{N}|$. The number of individuals on whom choice-model $tm^{[2]}$ is applied is denoted as $|\mathcal{N}_{tm^{[2]}}|$. The choice set of choice-model $tb^{[1]}$ is $\mathcal{C}_{tb^{[1]}} = \{y(yes), no\}$ and the choice set of choice-model is $\mathcal{C}_{tm^{[2]}} = \{b(bus), c(car)\}$. We assume everyone can access full choice sets for simplicity. Also, we assume coefficients are not individual specific and choice-model π's coefficient p is denoted as $\beta_{\pi p}$.

The systematic utility equations the were used in this example for the two choice-models are detailed below:

$$V_{tb^{[1]},no,n} = 0 \tag{4.1a}$$

$$V_{tb^{[1]}yn} = \beta_{tb^{[1]}y} + \beta_{tb^{[1]}\lambda_{tm^{[2]}}}\lambda_{tm^{[2]}n} \tag{4.1b}$$

$$V_{tm^{[2]}cn} = \beta_{tm^{[2]}ctt}CarTT_n + \beta_{tm^{[2]}ccost}CarCost_n \tag{4.1c}$$

$$V_{tm^{[2]}bn} = \beta_{tm^{[2]}b} + \beta_{tm^{[2]}btt}BusTT_n + \beta_{tm^{[2]}bcost}BusCost_n \tag{4.1d}$$

Individual's cost and travel time attributes are generated from normal distributions. $CarTT_n$ has mean as 6 minutes and standard deviation as 2 minutes; $CarCost_n$ has mean as 4 dollars and standard deviation as 2 dollars. $BusTT_n$ has mean as 10 minutes and standard deviation as 4 minutes; $BusCost_n$ has mean as 2 dollars and standard deviation as 0.5 dollars. The logsum term from choice-model $tm^{[2]}$ is denoted as $\lambda_{tm^{[2]}n}$, which is equal to $\log(e^{V_{tm^{[2]}bn}} + e^{V_{tm^{[2]}cn}})$.

The probability of travelling for each individual n is $P_n(tb^{[1]}y) = \frac{e^{V_{tb^{[1]}yn}}}{1+e^{V_{tb^{[1]}yn}}}$; the probability of traveling by car conditional on having travelled for each individual n is $P_n(tm^{[2]}c|\gamma_{tb^{[1]}n}) = \frac{e^{V_{tm^{[2]}cn}}}{e^{V_{tm^{[2]}cn}}+e^{V_{tm^{[2]}bn}}}$ and the probability of traveling by bus conditional on having travelled for each individual n is similar.

4.2 Solution procedure application

The population size in this toy example is 5,000. We first generate car travel time, car cost, bus travel time and bus cost by drawing from normal distributions. Then we are able to calculate logsum for each individual. After that, we can calculate utilities and probabilities of each alternative by above equations. Finally, we do microsimulation to get initial simulated aggregate statistics. Once we have initial values, we can start calibration using the algorithm described in Algorithm 1.

For simplicity, we only calibrate alternative specific constants and logsum coefficient, which are $\beta_{tb^{[1]}y}$, $\beta_{tm^{[2]}b}$ and $\beta_{tb^{[1]}\lambda_{tm^{[2]}}}$, in this example.

Thus $\boldsymbol{\beta} = \left[\beta_{tb^{[1]}y}, \beta_{tm^{[2]}b}, \beta_{tb^{[1]}\lambda_{tm^{[2]}}}\right]^T$. Also, we don't consider upper and lower bound on $\boldsymbol{\beta}$ and don't include a penalty term of $\boldsymbol{\beta}$ in the objective function. Then, the optimization problem of this illustrative example can be written as

$$\begin{aligned}\min_{\boldsymbol{\beta}} \quad & L(\boldsymbol{\beta}) = L_{tb^{[1]}}(\boldsymbol{\beta}) + L_{tm^{[2]}}(\boldsymbol{\beta}) \\ & = (A_{tb^{[1]}y} - S_{tb^{[1]}y}(\boldsymbol{\beta}))^2 + (A_{tm^{[2]}c} - S_{tm^{[2]}c}(\boldsymbol{\beta}))^2\end{aligned}$$

As illustrated in section 3.2.3, simulated aggregate statistics $S_{tb^{[1]}y}$ and $S_{tm^{[2]}c}$ can

be written as

$$S_{tb^{[1]}y} = \sum_{n \in \mathcal{N}} P_n(tb^{[1]}y) \tag{4.2a}$$

$$S_{tm^{[2]}c} = \sum_{n \in \mathcal{N}_{tm^{[2]}}} P_n(tm^{[2]}c|\gamma_{tb^{[1]}n}) \tag{4.2b}$$

According to section 3.3 and Equation 3.18, since choice-model $tb^{[1]} \in \dot{\Pi}_*$, the gradient of objective function L with respect to coefficient $\beta_{tb^{[1]}y}$ can be written as

$$\begin{aligned} \frac{\partial L}{\partial \beta_{tb^{[1]}y}} &= \frac{\partial L_{tb^{[1]}}}{\partial \beta_{tb^{[1]}y}} = -2(A_{tb^{[1]}y} - S_{tb^{[1]}y}) \sum_{n \in \mathcal{N}} \frac{\partial P_n(tb^{[1]}y)}{\partial \beta_{tb^{[1]}y}} \\ &= -2(A_{tb^{[1]}y} - S_{tb^{[1]}y}) \sum_{n \in \mathcal{N}} \sigma(V_{tb^{[1]}yn})(1 - \sigma(V_{tb^{[1]}yn})) \end{aligned} \tag{4.3}$$

where the expression $\frac{\partial V_{tb^{[1]}yn}}{\partial \beta_{tb^{[1]}y}} = 1$ according to equation 4.1b and the expression $\frac{\partial L_{tm^{[2]}}}{\partial \beta_{tb^{[1]}y}} = 0$ because choice-model $tm^{[2]}$ does not have any term dependent on coefficient $\beta_{tb^{[1]}y}$.

The gradient of objective function L with respect to coefficient $\beta_{tb^{[1]}\lambda_{tm^{[2]}}}$ can be written as

$$\begin{aligned} \frac{\partial L}{\partial \beta_{tb^{[1]}\lambda_{tm^{[2]}}}} &= \frac{\partial L_{tb^{[1]}}}{\partial \beta_{tb^{[1]}\lambda_{tm^{[2]}}}} = -2(A_{tb^{[1]}y} - S_{tb^{[1]}y}) \sum_{n \in \mathcal{N}} \frac{\partial P_n(tb^{[1]}y)}{\partial \beta_{tb^{[1]}y}} \\ &= -2(A_{tb^{[1]}y} - S_{tb^{[1]}y}) \sum_{n \in \mathcal{N}} \sigma(V_{tb^{[1]}yn})(1 - \sigma(V_{tb^{[1]}yn}))\lambda_{tm^{[2]}n} \end{aligned} \tag{4.4}$$

where the expression $\frac{\partial V_{tb^{[1]}yn}}{\partial \beta_{tb^{[1]}\lambda_{tm^{[2]}}}} = \lambda_{tm^{[2]}n}$ according to equation 4.1b and the expression $\frac{\partial L_{tm^{[2]}}}{\partial \beta_{tb^{[1]}y}} = 0$ because choice-model $tm^{[2]}$ does not have any term dependent on coefficient $\beta_{tb^{[1]}\lambda_{tm^{[2]}}}$.

However, since choice-model $tm^{[2]} \in \dot{\Pi}^*_*$, the gradient of objective function L with respect to coefficient $\beta_{tm^{[2]}b}$ can be written as

$$\begin{aligned}\frac{\partial L}{\partial \beta_{tm^{[2]}b}} &= \frac{\partial L_{tb^{[1]}}}{\partial \beta_{tm^{[2]}b}} + \frac{\partial L_{tm^{[2]}}}{\partial \beta_{tm^{[2]}b}} \\ &= -2(A_{tb^{[1]}y} - S_{tb^{[1]}y}) \sum_{n \in \mathcal{N}} \frac{\partial \sigma(V_{tb^{[1]}yn})}{\partial V_{tb^{[1]}yn}} \frac{\partial V_{tb^{[1]}yn}}{\partial \beta_{tm^{[2]}b}} \\ &+(-2)(A_{tm^{[2]}c} - S_{tm^{[2]}c}) \sum_{n \in \mathcal{N}} \frac{\partial \psi(\boldsymbol{V}_{tm^{[2]}n})_b}{\partial V_{tm^{[2]}bn}} \frac{\partial V_{tm^{[2]}bn}}{\partial \beta_{tm^{[2]}b}} \end{aligned} \tag{4.5a}$$

where the expression $\frac{\partial V_{tm^{[2]}bn}}{\partial \beta_{tm^{[2]}b}} = 1$ according to Equation 4.1d and the expression $\frac{\partial V_{tb^{[1]}yn}}{\partial \beta_{tm^{[2]}b}}$ is not equal to zero since choice-model $tb^{[1]}$ contains a logsum term from choice-model $tm^{[2]}$, which is a function of $\beta_{tm^{[2]}b}$. As a results, it can be written as

$$\frac{\partial V_{tb^{[1]}yn}}{\partial \beta_{tm^{[2]}b}} = \beta_{tb^{[1]}\lambda_{tm^{[2]}}} \frac{\partial \lambda_{tm^{[2]}n}}{\partial \beta_{tm^{[2]}b}} = \beta_{tb^{[1]}\lambda_{tm^{[2]}}} \frac{e^{V_{tm^{[2]}bn}}}{e^{V_{tm^{[2]}bn}} + e^{V_{tm^{[2]}cn}}} \tag{4.5b}$$

Therefore, by combining above equations, we have

$$\begin{aligned}\frac{\partial L}{\partial \beta_{tm^{[2]}b}} &= -2(A_{tb^{[1]}y} - S_{tb^{[1]}y}) \sum_{n \in \mathcal{N}} \sigma(V_{tb^{[1]}yn})(1 - \sigma(V_{tb^{[1]}yn})) \frac{\beta_{tb^{[1]}\lambda_{tm^{[2]}}} e^{V_{tm^{[2]}bn}}}{e^{V_{tm^{[2]}bn}} + e^{V_{tm^{[2]}cn}}} \\ &+(-2)(A_{tm^{[2]}c} - S_{tm^{[2]}c}) \sum_{n \in \mathcal{N}} \psi(\boldsymbol{V}_{tm^{[2]}n})_c(-\psi(\boldsymbol{V}_{tm^{[2]}n})_b) \end{aligned} \tag{4.5c}$$

In this experiment, we use mini-batch gradient descent algorithm to minimize the objective function L, with the number of mini-batches η as 5. Therefore, in every iteration, we cannot evaluate the simulated aggregate statistics of full population, $S_{tb^{[1]}y}$ and $S_{tm^{[2]}c}$. However, as shown in section 3.3.4, we can estimate these two values through scaling the simulated aggregate statistics of mini-batch by η. As a result, $S_{tb^{[1]}y} = 5\tilde{S}_{tb^{[1]}y}$ and $S_{tm^{[2]}c} = 5\tilde{S}_{tm^{[2]}c}$.

Similarly, we also need to estimate the actual aggregate statistics of mini-batch,

$\tilde{A}_{tb^{[1]}y}$ and $\tilde{A}_{tm^{[2]}c}$, through scaling the actual aggregate statistics of full population by η. As a result, $A_{tb^{[1]}y} = 5\tilde{A}_{tb^{[1]}y}$ and $A_{tm^{[2]}c} = 5\tilde{A}_{tm^{[2]}c}$.

Finally, we need to estimate the objective function value of full population by scaling the objective function value of mini-batch. As a result, $L(\boldsymbol{\beta}) = 25\tilde{L}(\boldsymbol{\beta})$. Then, in order to estimate the gradients of full population, we can scale the gradients of mini-batch by η^2 following the same arguments. In order to update parameters, we can take estimated gradients of full population as our search direction and multiply it with a proper step size —$\alpha = 1 \times 10^{-8}$ in this example— to iteratively update coefficients $\boldsymbol{\beta}$ until convergence. The expression of updating coefficients in iteration κ can be written as

$$\boldsymbol{\beta}^{\kappa+1} = \boldsymbol{\beta}^{\kappa} - \alpha \nabla_{\boldsymbol{\beta}^{\kappa}} \tilde{L}(\boldsymbol{\beta}^{\kappa}) \eta^2 \tag{4.6}$$

4.3 Results

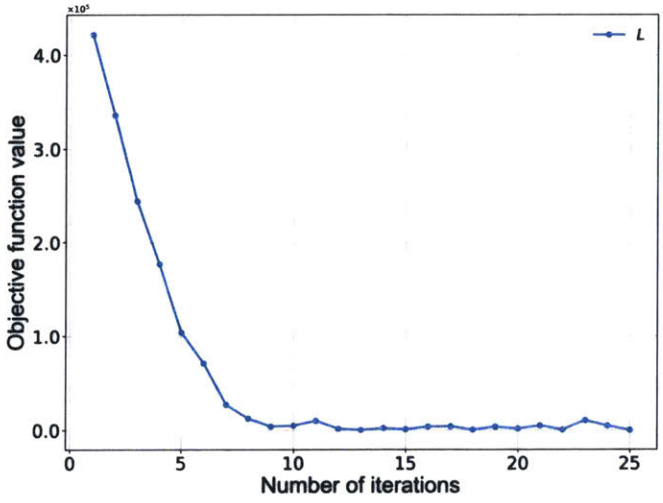


Figure 4-2: Illustrative example mini-batch gradient descent objective value plot. The estimated objective value of full population drops close to 0 in the first few iterations. It is possible to have a faster and more stable convergence if use some diminshing stepsize rule.

We run 25 iterations in total, which include 5 master iterations. We assume that the actual number of individuals travelling, $A_{tb^{[1]}y}$, is 4000 and the actual number of individuals travelling by car, $A_{tm^{[2]},bus}$, is 1000. The estimated objective function value of full population, the estimated simulated aggregate statistics of full population, the coefficients values, and the estimated gradients of full population are shown in Table 4.1.

We also plot the estimated objective function value of full population in Figure 4-2. As we can see, the estimated objective value of full population drops quickly in the first few iterations and converges close to near zero in the end. The final estimated number of individuals traveling is 3995, which is very close to $A_{tb^{[1]}y}$; and the final estimated number of individuals traveling by bus is 970, which is also close to $A_{tm^{[2]},bus}$.

This illustrative example shows the complete procedure to calibrate utility-maximizing nested logit ABMs model system using our proposed solution procedure. In next section, we will show that our solution procedure is applicable on a real and complex ABMs.

Iteration		$L(\boldsymbol{\beta})$	$S_{tb^{[1]}y}(\boldsymbol{\beta})$	$S_{tm^{[2]}b}(\boldsymbol{\beta})$	$\beta_{tb^{[1]}y}$	$\beta_{tm^{[2]}b}$	$\beta_{tb^{[1]}\lambda_{tm^{[2]}}}$	$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_{tb^{[1]}y}}$	$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_{tm^{[2]}b}}$	$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_{tb^{[1]}\lambda_{tm^{[2]}}}}$
Master 1	Batch 1	421772.791	4540	1114	3.996	−0.997	0.327	440602.201	−272527.339	−2672033.222
	Batch 2	336058.616	4486	1149	3.991	−0.995	0.353	435464.617	−215877.364	−2594984.595
	Batch 3	244175.489	4396	1079	3.987	−0.993	0.377	403767.759	−194934.164	−2444033.950
	Batch 4	177351.069	4341	1128	3.983	−0.992	0.399	373286.440	−166130.674	−2208706.819
	Batch 5	104257.754	4263	1043	3.980	−0.990	0.418	12482.549	−4569.712	−74116.501
...	...	...	...	...	...	...	...	...	...	...
Master 5	Batch 1	4911.701	3956	1004	3.974	−0.988	0.456	−2628.493	1371.811	15942.884
	Batch 2	671.665	4025	1025	3.974	−0.988	0.458	36210.797	7685.062	−215188.993
	Batch 3	10498.25	4042	996	3.973	−0.988	0.462	60447.170	−58751.824	−354238.954
	Batch 4	4816.172	3986	1020	3.973	−0.988	0.461	−19138.727	48278.094	114179.310
	Batch 5	231.587	3995	970	3.973	−0.988	0.460	−6567.549	9698.641	39181.522

Table 4.1: Illustrative example result table. At the end of 25 iterations, the estimated objective value of full population decreases to a small number; the estimated simulated aggregate statistics of full population are close to actual aggregate statistics; estimated gradients of full population are much less than those in the beginning of iterations.

Chapter 5

Case Study

This chapter reports the results from experiments done on a prototype city to demonstrate the efficacy of the calibration approach discussed above. In this case study, we use Day Activity-Schedule (DAS) model system —a type of nested logit ABM that utilizes utility-maximizing choice-models. We compare the proposed solution procedure with different algorithms with simultaneous perturbation stochastic approximation (SPSA), a typical black-boxed based algorithm for simulation-based optimization. Section 5.1 presents ABM structure used, an overview of simulation set up, and the details of the problem formulation for this case study. Section 5.2 introduces SPSA algorithm, implementation details of solution procedure with different gradient-based algorithms and compare them when applied to the prototype city.

5.1 Overview

ABMs used in the case study is implemented for a prototype city that represents a public transportation oriented city like Singapore or London. It follows the same DAS model system illustrated in section 5.1.1. The number of persons in population of this prototype city is $351,518$. The prototype city is divided in to 24 TAZ zones and its network is comprised on 95 nodes, 254 links, and 286 segments.

5.1.1 Day Activity-Schedule

DAS based ABM utilizes a deep nested logit structure to represent different travel choices as a multidimensional choices with shared unobserved elements. Specifically, we consider a recent and more complex DAS model system detailed in [36] in this case study. However, the formulation and solution procedure can be extended to any other nested logit ABM architecture.

As shown in Figure 5-1 and detailed in Table 5.1, the considered DAS model system consists of interconnected discrete choice-models that are characterized into three levels: (i) Day Pattern Level, which determines if an individual will travel and, if so, the types of primary activities and intermediate stops she will do; (ii) Tour Level, which determines the mode, destination, and time-of-day of the different primary activities; and (iii) Intermediate Stop Level, which generates intermediate stops within the tours —including their mode, destination, and time-of-day.

The choice-models in lower levels pertain to activity-travel decisions conditional on the upper level decisions; the models across these levels are interconnected through inclusive values, which are also known as logsums. In Figure 5-1, the conditional dependence between the choice-models is shown through solid arrows and the inclusive values (bottom-up) relationships are shown through dashed arrows. For example, *dpb* choice-model considers the inclusive values of *dpt* and *dps* choice-models as inputs to decide if an individual travels or not.

For illustration, consider the day pattern tour choice-model (*dpt*) in the DAS model system in Figure 5-1. The actual number of individuals who chose a particular pattern i among the alternatives in the *dpt* choice-model is A_{dpti}. A simulated individual will be exposed to the *dpt* model and its choice set $\mathcal{C}_{dpt}$, if she chooses to travel which is decided by the upper *dpb* model. The simulated number of individuals who choose the pattern i in the choice-model *dpt*, i.e. S_{dpti}, is then determined by running the simulation on the synthetic population and counting the number of individuals who choose the pattern i in choice-model *dpt*. Then, S_{dpti} and A_{dpti} are compared to determine the changes to the parameters being calibrated.

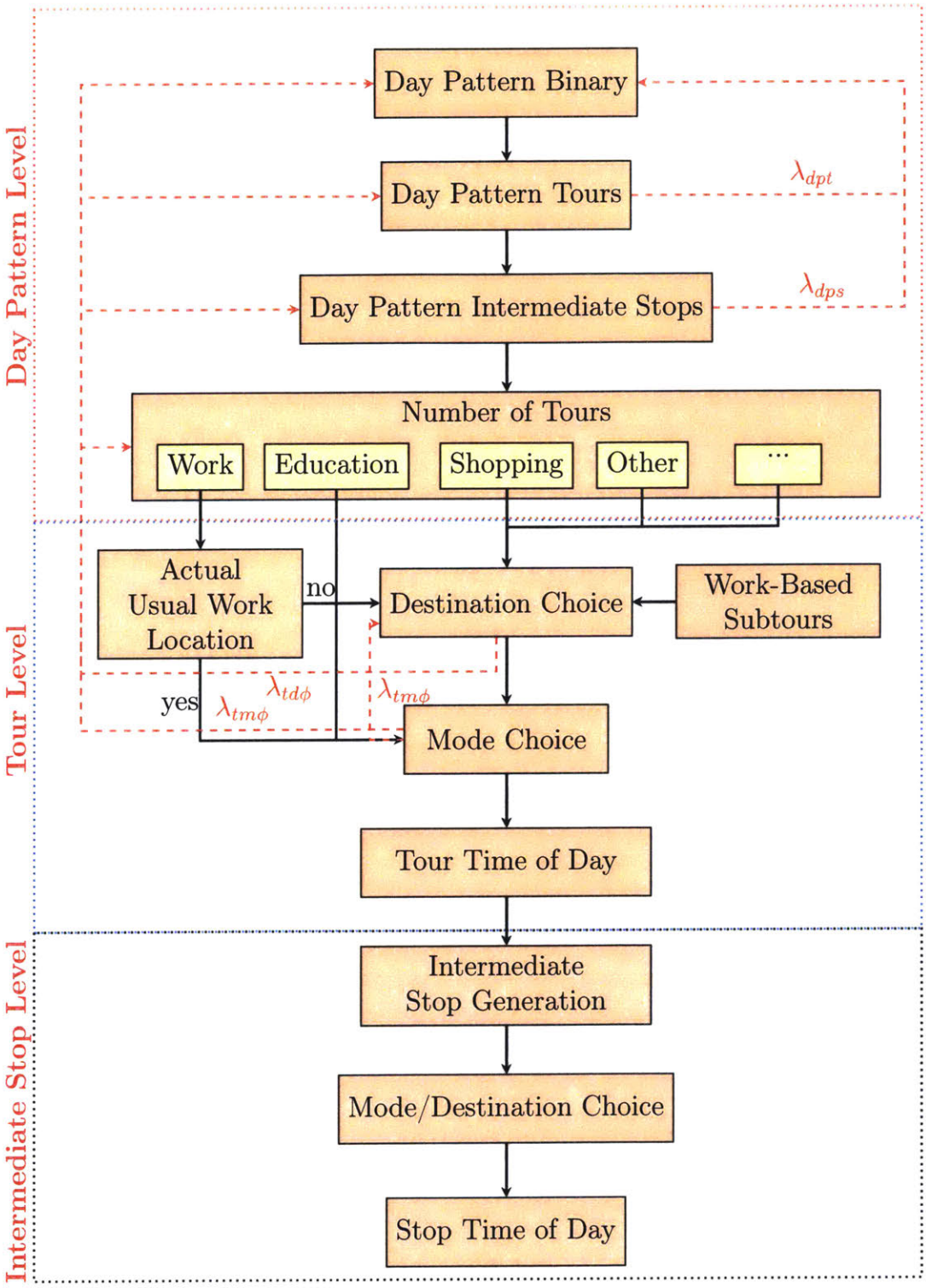


Figure 5-1: Day Activity-Schedule (DAS) model system structure. Each node represents a choice-model. Solid arrows represent the conditional dependence between the choice-models and dashed arrows represent the inclusive values λ_π (bottom-up) relationships. In the figure, 'usual' work tours refer to work tours with fixed destinations and 'unusual' work tours refer to work tours needed to estimate destinations.

Level	Choice-Model Name (π)	Choice Set ($\mathcal{C}_\pi$) Description
Day Pattern Level	Day Pattern Binary (*dpb*)	Binary choice for traveling or not
	Day Pattern Tours (*dpt*)	Combinations of different tour purposes
	Day Pattern Stops (*dps*)	Combinations of different intermediate stop purposes
	Number of Tours ($nt\phi$)	Number of tours for a selected purpose, ϕ ($\phi \in \Phi$)
Tour Level	Usual Work (*tuw*)	Binary choice if attending usual work
	Tour Mode ($tm\phi$)	Choice of modes for usual work or a selected purpose, ϕ
	Tour Destination ($td\phi$)	Choice of destinations for a selected purpose, ϕ, other than usual work/education
	Tour Time of Day (*ttd*)	1176 time window combinations for 30 min intervals in 24 hours
Intermediate Stop Level	Intermediate Stop Generation (*isg*)	Quit + number of different tour purposes
	Intermediate Stop Mode/Destination (*imd*)	Combinations of different modes and destinations
	Intermediate Stop Time of Day (*itd*)	Choice of arrival time for 30 min intervals in 24 hrs

Table 5.1: DAS model system. The selected DAS model system contains three levels. Choice-models in each level and their choice set descriptions are listed.

5.1.2 Simulation set up

For the sake of brevity and for the purpose of this case study, we calibrate only the aggregate statistics from *dpb*, *dpt*, $nt\phi$, and $tm\phi$ choice-models. The complete set of purposes Φ was $\{work, edu, shop, other\}$. Further, there were nine modes in the choice set $\mathcal{C}_{tm\phi}$ for $tm\phi$ choice-model: *bus*, *rail*, *private bus*, *SOV*, *carsharing*2, *carsharing*3, *motorcycle*, *walk* and *taxi*. *carsharing*2 represents two individuals sharing one car and *carsharing*3 represents three or more individuals sharing one car; *private bus* represents regular shuttle service like school bus.

5.1.3 Problem formulation

As explained in section 3.2, we can formulate the objective function $L(\boldsymbol{\beta})$ as the sum of the component pertaining to each choice-model π, denoted as L_π. In this case

study, we do not have apriori term in our objective function for simplicity. As a result, the complete objective function $L(\boldsymbol{\beta})$ for the purpose of this case study can be written as

$$L(\boldsymbol{\beta}) = L_{dpb}(\boldsymbol{\beta}) + L_{dpt}(\boldsymbol{\beta}) + \sum_{\phi \in \Phi} L_{nt\phi}(\boldsymbol{\beta}) + \sum_{\phi \in \Phi} L_{tm\phi}(\boldsymbol{\beta}) \tag{5.1}$$

For *dpb* choice-model, we only calibrate the number of individuals traveling. The estimated number of individuals traveling from *dpb* choice-model is S_{dptyes}. Thus the objective function $L_{dpb}(\boldsymbol{\beta})$ is:

$$L_{dpb}(\boldsymbol{\beta}) = w_{dpbyes}(S_{dptyes}(\boldsymbol{\beta}) - A_{dptyes})^2 \tag{5.2}$$

where w_{dpbyes} should be equal to 1 because within *dpb* choice-model level there is only one term in objective function.

As briefly mentioned in section 3.2, we aggregate 14 different day pattern tour choices in *dpt* choice-model to form 4 different aggregate choices according to 4 different tour purposes: Work, Education, Shopping and Other. Then, we minimize differences between actual number of individuals and estimated number of individuals choosing each of these 4 aggregate choices. The estimated number of individuals choosing purpose $\phi \in \Phi$ in *dpt* choice-model, $S_{dpt\phi}$, is equal to the sum of the estimated number of individuals choosing alternative i, S_{dpti}, —across all alternatives— such that the alternative i contains purpose ϕ tour. As a result, the component of the objective function pertaining to *dpt* choice-model $L_{dpt}(\boldsymbol{\beta})$ is:

$$\begin{aligned} L_{dpt}(\boldsymbol{\beta}) = & \, w_{dptwork}(S_{dptwork}(\boldsymbol{\beta}) - A_{dptwork})^2 \\ & + w_{dptedu}(S_{dptedu}(\boldsymbol{\beta}) - A_{dptedu})^2 \\ & + w_{dptshop}(S_{dptshop}(\boldsymbol{\beta}) - A_{dptshop})^2 \\ & + w_{dptother}(S_{dptother}(\boldsymbol{\beta}) - A_{dptother})^2 \end{aligned} \tag{5.3}$$

where within *dpt* choice-model level, there are $|\Phi|$ components in objective function, so $\sum_{\phi \in \Phi} w_{dpt\phi}$ should be equal to 1 to make sure *dpb* choice-model error term and *dpt*

choice-model error terms have the same scale in objective function. An intuitive way to set values of the weights $w_{dpt\phi}$ is according to the number of individuals choosing purpose ϕ.

$$w_{dpt\phi} = \frac{A_{dpt\phi}}{\sum_{\phi \in \Phi} A_{dpt\phi}} \tag{5.4}$$

For the next level, we have 4 independent number of tours choice-models for 4 different tour purposes which are as follows: number of work tours (ntw), number of education tours (nte), number of shopping tours (nts), and number of other tours (nto). The choice-set $\mathcal{C}_{nt\phi}$ is $\{1, 2\}$, which only contains two possible choices and is the same for all $\phi \in \Phi$. Thus the objective function $L_{nt}(\boldsymbol{\beta})$ will be:

$$\begin{aligned} L_{nt\phi}(\boldsymbol{\beta}) = w_{ntw1}(S_{ntw1}(\boldsymbol{\beta}) - A_{ntw1})^2 + w_{nte1}(S_{nte1}(\boldsymbol{\beta}) - A_{nte1})^2 \\ + w_{nts1}(S_{nts1}(\boldsymbol{\beta}) - A_{nts1})^2 + w_{nto1}(S_{nto1}(\boldsymbol{\beta}) - A_{nto1})^2 \end{aligned} \tag{5.5}$$

where $w_{nt\phi 1} = \frac{A_{nt\phi 1}}{\sum_{\phi \in \Phi} A_{nt\phi 1}}$ using the similar argument as in the discussion regarding dpt choice-model above.

The $tm\phi$ and $td\phi$ choice-models are combined together in the current experiments. The population of workers are divided two categories: ones with usual work location and ones without usual work location. For the education tours and usual work tours, the destination is fixed; therefore we need only to calibrate mode choice-models. The education tours mode choice-model is tme, and usual work tours mode choice-model is tmw. For unusual work tours, shopping tours, and other tours, the mode and destination choices are combined into a single multidimentional choice-models which are denoted as $tmdw$, $tmds$, and $tmdo$ respectively. As we calibrate only mode choices in this case study, we consider these choice-models separately.

For tour mode choice-models, although there are 9 modes available, we only calibrate three of them, which are bus, rail and SOV, as these three modes are major modes in the prototype city. In other words, the full choice set of $tm\phi$ choice-model, $\mathcal{C}_{tm\phi}$, has 9 modes and the subset choice set $\tilde{\mathcal{C}}_{tm\phi} \subseteq \mathcal{C}_{tm\phi}$ is $\{bus,\ rail,\ SOV\}$. Based

on the above discussion the objective function $L_{tm\phi}(\boldsymbol{\beta})$ will be:

$$\begin{aligned} L_{tm\phi}(\boldsymbol{\beta}) = & \sum_{i \in \tilde{\mathcal{C}}_{tmw}} w_{tmwi}(S_{tmwi}(\boldsymbol{\beta}) - A_{tmwi})^2 \\ & + \sum_{i \in \tilde{\mathcal{C}}_{tme}} w_{tmei}(S_{tmei}(\boldsymbol{\beta}) - A_{tmei})^2 \\ & + \sum_{i \in \tilde{\mathcal{C}}_{tmdw}} w_{tmdwi}(S_{tmdwi}(\boldsymbol{\beta}) - A_{tmdwi})^2 \\ & + \sum_{i \in \tilde{\mathcal{C}}_{tmds}} w_{tmdsi}(S_{tmdsi}(\boldsymbol{\beta}) - A_{tmdsi})^2 \\ & + \sum_{i \in \tilde{\mathcal{C}}_{tmdo}} w_{tmdoi}(S_{tmdoi}(\boldsymbol{\beta}) - A_{tmdoi})^2 \end{aligned} \tag{5.6}$$

where $w_{tm\phi/tmd\phi i} = \frac{A_{tm\phi/tmd\phi i}}{\sum_{\phi \in \Phi} \sum_{i \in \tilde{\mathcal{C}}_{tm\phi/tmd\phi}} A_{tm\phi/tmd\phi i}}$ for $i \in \tilde{\mathcal{C}}_{tm\phi/tmd\phi}$

The estimated number of individuals choosing choice-model π's alternative i, $S_{\pi i}$, is calculated through microsimulation as explained in section 3.2.3. The initial number of individuals, $S_{\pi i}(\boldsymbol{\beta^0})$, actual number of individuals, $A_{\pi i}$, and calibrated number of individuals, $S_{\pi i}(\boldsymbol{\beta})$, choosing each alternative i under each choice-model π are shown in Table 5.2. The numbers we are actually calibrating are bold. The number of choices within each choice set $\mathcal{C}_\pi$ is also shown in Table 5.2. From Table 5.2, we can see that the majority of individuals are traveling and most of them have work tours. Among all four different types of tours, majority of individuals conduct only a single tour of a given purpose.

Level	Choice-Model Name	Choice ($\mathcal{C}_{\pi,i}$)	i	$S_{\pi i}(\beta^0)$	$A_{\pi i}$	GD $S_{\pi i}(\beta)$	GM $S_{\pi i}(\beta)$	Adam $S_{\pi i}(\beta)$	BFGS $S_{\pi i}(\beta)$	SPSA $S_{\pi i}(\beta)$
Day Pattern Level	Day Pattern Binary (*dpb*)	2	**Travel**	**292079**	**309335**	**309755**	**309355**	**312155**	**308170**	**309540**
			Not Travel	59439	42183	41763	42163	39363	43348	41978
	Day Pattern Binary (*dpb*)	14	**Work**	**221684**	**247468**	**238610**	**244200**	**242090**	**238770**	**232730**
			Edu	**66621**	**61867**	**67945**	**60925**	**68770**	**67465**	**66885**
			Shop	**89**	**117**	**100**	**54**	**10**	**78**	**3000**
			Other	**26270**	**37120**	**36470**	**43005**	**31895**	**55415**	**54735**
	# of Work Tours (*ntw*)	(1,2)	**1**	**196720**	**215297**	**214915**	**214020**	**214040**	**214305**	**204989**
			2	24964	32171	23695	30180	28050	24465	27741
	# of Edu Tours (*nte*)	(1,2)	**1**	**66185**	**60629**	**67385**	**60470**	**61095**	**65860**	**48670**
			2	436	1238	560	455	7675	1605	18215
	# of Shop Tours (*nts*)	(1,2)	**1**	**74**	**93**	**85**	**49**	**8**	**70**	**2901**
			2	15	24	15	5	2	8	99
	# of Other Tours (*nto*)	(1,2)	**1**	**24898**	**35635**	**34265**	**40195**	**31850**	**41455**	**48482**
			2	1372	1485	2205	2810	45	13960	6253
Tour Level	Usual Work Tour Mode (*tmw*)	9	**Bus**	**49302**	**55592**	**51040**	**51900**	**51970**	**49885**	**52715**
			Rail	**66395**	**74771**	**69700**	**71275**	**71335**	**70655**	**68970**
			SOV	**45569**	**51323**	**43595**	**43625**	**40255**	**45230**	**43150**
	Education Tour Mode (*tme*)	9	**Bus**	**21789**	**20502**	**21850**	**19805**	**21385**	**24849**	**28785**
			Rail	**12241**	**11516**	**12620**	**11070**	**12140**	**18185**	**12560**
			SOV	**1615**	**1515**	**1745**	**1175**	**1695**	**1530**	**1625**
	Unusual Work Tour Mode (*tmdw*)	9*24	**Bus**	**4953**	**5684**	**5790**	**8425**	**5990**	**5395**	**6665**
			Rail	**6136**	**7160**	**7315**	**10415**	**8315**	**6735**	**5838**
			SOV	**2122**	**2432**	**2095**	**2410**	**2100**	**1768**	**2238**
	Shopping Tour Mode (*tmds*)	9*24	**Bus**	**10**	**18**	**23**	**5**	**0**	**33**	**165**
			Rail	**5**	**10**	**9**	**3**	**0**	**2**	**97**
			SOV	**0**	**0**	**1**	**0**	**0**	**0**	**3**
	Other Tour Mode (*tmdo*)	9*24	**Bus**	**6577**	**9110**	**9200**	**10685**	**6940**	**12879**	**18249**
			Rail	**4307**	**5964**	**5975**	**7020**	**4680**	**13770**	**6235**
			SOV	**1**	**2**	**2**	**2**	**2**	**2**	**2**

Table 5.2: Initial aggregate statistics $S_{\pi i}(\beta^0)$, actual aggregate statistics $A_{\pi i}$, and final aggregate statistics $S_{\pi i}(\beta)$. Values entered into objective function are bold.

5.2 Application

In this case study, we used SimMobility [19] for microsimulation to evaluate objective function and simulate activity schedules. The number of coefficients $\beta_{\pi i}$ we calibrated were 28 and they were alternative specific constants and scale coefficients of logsum terms from *dpb*, *dpt*, *ntw*, *nte*, *nts*, *nto*, *tmw*, *tme*, *tmdw*, *tmds*, and *tmdo* choice-models. We divided the population into 5 mini-batches and limited the total number of master iterations to six, which were the same across all the algorithms.

In this experiment, we implemented the proposed solution procedure by five different mini-batch based algorithms which are the following: mini-batch gradient descent, mini-batch gradient with momentum, mini-bath BFGS, mini-batch Adam, and mini-batch SPSA. We implemented mini-batch SPSA as it a typical black-boxed based algorithm for simulation-based optimization to compare with proposed solution procedure with different algorithms. In the following sections, we first explains numerical gradient check to ensure our estimated gradients from analytical expressions are correct. Then we briefly introduce SPSA algorithm. After that, we explain the details of the hyperparameters for each of the five algorithms. Finally, we present comparison among these five algorithms.

5.2.1 Numerical gradient check

As the complexity of the expression, implementations are prone to errors. In order to ensure the correct implementation of the analytical gradient of objective function L with respect to ith coefficient in $\boldsymbol{\beta}_i$, we can compare it against the numerical derivate obtained from finite-differences. A straightforward approach is to use finite difference (FD) two-point estimation, which calculates the slope of a nearby secant line through the points $(\boldsymbol{\beta} - e_i c, L(\boldsymbol{\beta} - e_i c))$ and $(\boldsymbol{\beta} + e_i c, L(\boldsymbol{\beta} + e_i c))$, where e_i is an unit vector with ith element equal to 1 and c is a small number —1×10^{-7} in this experiment— representing a small change in $\boldsymbol{\beta}_i$. The slope of this line, which is also numerical

gradient with respect to $\boldsymbol{\beta}_i$, can be written as

$$\frac{L(\boldsymbol{\beta} + e_i c) - L(\boldsymbol{\beta} - e_i c)}{2c} \tag{5.7}$$

Consequently, we can calculate numerical gradients of all elements in $\boldsymbol{\beta}$ and denote the vector of numerical gradients as $\nabla_{\boldsymbol{\beta}} L'(\boldsymbol{\beta})$. The equation we use to measure the difference between analytical gradients and numerical gradients is shown below

$$\frac{\|\nabla_{\boldsymbol{\beta}} L'(\boldsymbol{\beta}) - \nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta})\|}{\|\nabla_{\boldsymbol{\beta}} L'(\boldsymbol{\beta})\| + \|\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta})\|} \tag{5.8}$$

The value obtained was 1×10^{-5}, which is a small number and proves that our analytical gradients are calculated correctly. The same procedure was repeated at multiple points to make sure that the implementation of the gradient was correct.

5.2.2 SPSA

Simultaneous perturbation stochastic approximation (SPSA) is a type of stochastic approximation (SA) algorithm. Stochastic approximation algorithm attempts to solve stochastic optimization problems iteratively. For our case, the randomness in optimization problem comes from simulated aggregate statistics $S_{\pi i}$ simulated from activity simulator. The general iterative form of SA is:

$$\boldsymbol{\beta}^{\kappa+1} = \boldsymbol{\beta}^{\kappa} - \alpha_{SA}^{\kappa} \nabla_{\boldsymbol{\beta}^{\kappa}} L'(\boldsymbol{\beta}^{\kappa}) \tag{5.9}$$

where $\boldsymbol{\beta}^{\kappa}$ is coefficient vector we want to calibrate at iteration κ and $\nabla_{\boldsymbol{\beta}^{\kappa}} L'(\boldsymbol{\beta}^{\kappa})$ represents the estimated gradient at iteration κ. α_{SA}^{κ} represents the step size at iteration κ, which becomes smaller as κ increases. It can be written as:

$$\alpha_{SA}^{\kappa} = \frac{\alpha_{SA}}{(A+\kappa+1)^{a}} \tag{5.10}$$

where α_{SA}, A and α are hyperparameters need to tune and will be discussed later.

There are many different ways to estimate $\nabla_{\boldsymbol{\beta}^{\kappa}} L'(\boldsymbol{\beta}^{\kappa})$. One way to approximate gradient is to perform two-point estimation for each element of $\boldsymbol{\beta}$ as we did in numerical gradient check. However, there is one modification required to have perturbation parameter c decreases as κ increases, such that perturbation at iteration κ is expressed as $c^{\kappa} = \frac{c}{(\kappa+1)^{\gamma}}$, where c and γ are hyperparameters.

Another way to approximate gradient is to apply SPSA [17], which estimates the gradient by perturbing all parameters in $\boldsymbol{\beta}$ simultaneously. Thus the approximation of gradient only needs two function evaluations instead of $2|\boldsymbol{\beta}|$ function evaluations in finite difference (FD) scheme. The pth element in approximated gradient $\nabla_{\boldsymbol{\beta}^{\kappa}} L'(\boldsymbol{\beta}^{\kappa})$ vector can be written as:

$$(\nabla_{\boldsymbol{\beta}^{\kappa}} L'(\boldsymbol{\beta}^{\kappa}))_p = \frac{(L(\boldsymbol{\beta} + c^{\kappa}\Delta^{\kappa})) - L(\boldsymbol{\beta} - c^{\kappa}\Delta^{\kappa})}{2c^{\kappa}(\Delta^{\kappa})_p} \tag{5.11}$$

where c^{κ} is perturbation parameter same as in the FD approach. Δ^{κ} represents a random perturbation vector at iteration κ, which is generated by independently drawing from a Bernoulli ± 1 distribution with probability $\frac{1}{2}$ for each ± 1 outcome as suggested by [17]. More details about SPSA can also be found in [17]. In terms of convergence performance, [17] argues that SPSA performs as good as FDSA while has $|\boldsymbol{\beta}|$-fold simulation saving. Since FDSA is proved to surely converge [52] provided that objective function satisfies certain regularity conditions, SPSA is popular and applied in calibration problems. However, in real world applications, if the domain is quite large, SPSA does not have satisfactory performance [27].

5.2.3 Stepsize tuning

Further, the stepsizes for each of the algorithms was chosen empirically. First we choose an initial step size by observing initial magnitudes of search direction and making sure that the scale of the update is reasonable. Then, we tuned the step size by randomly sampling several different values around initial step size and selected the one that resulted in the least final objective value.

Mini-batch Gradient Descent (GD)

For mini-batch gradient descent, the step size α_{GD} was chosen as 3×10^{-10}.

Mini-batch BFGS

For mini-batch BFGS algorithm, we also needed to decide on the step size, α_{BFGS}. After tuning, the best step size we adopted was:

$$\alpha_{BFGS} = 3 \times 10^{-10} \tag{5.12}$$

Mini-batch Gradient with momentum

The mini-batch gradient with momentum needs two hyper parameters, which are the first moment exponential decay rate θ_1 and step size α_{GM}. After tuning, we adopted the following values:

$$\theta_1 = 0.9, \ \alpha_{GM} = 3 \times 10^{-10} \tag{5.13}$$

Mini-batch Adam

For mini-batch Adam, in addition to the first moment exponential decay rate θ_1 and step size α_{adam}, there is one more hyper parameter, which is the second moment exponential decay rate θ_2. After tuning, we adopted the following values:

$$\theta_1 = 0.9, \quad \theta_2 = 0.99, \ \alpha_{Adam} = 0.05 \tag{5.14}$$

Mini-batch SPSA

For mini-batch SPSA algorithm, we need to compute α_{SPSA}^{κ}, which is the step size used in coefficients update at iteration κ, and C^{κ} which is the parameter used in perturbation and gradient calculation at iteration κ.

$$\alpha_{SPSA}^{\kappa} = \frac{\alpha_{SPSA}}{(A+\kappa+1)^{a}}, \; C^{\kappa} = \frac{c}{(\kappa+1)^{\gamma}} \tag{5.15}$$

By observing initial magnitudes of SPSA estimated gradients, we adopted the following values:

$$\alpha_{SPSA} = 3 \times 10^{-10}, A = 10, c = 0.1, a = 0.602, \gamma = 0.101 \tag{5.16}$$

As we can see, the step size of mini-batch GD is greater than the step size of mini-batch SPSA. This is because the magnitude of the search direction in mini-batch SPSA is greater than the magnitude of the search direction from mini-batch GD. When we let mini-batch SPSA have the same step size as α_{GD}, mini-batch SPSA seems to diverge to an extremely high objective function value in the beginning.

5.2.4 Performance comparisons

In this section, we examine the performances of all algorithms using the estimated objective function values of full population. Figure 5-2 plots the log of estimated objective function values of all algorithms we implement across every iteration. Then, Figure 5-3 depicts each objective function component's estimated value by mini-batch GD, mini-batch GM, mini-btach BFGS, and mini-batch Adam algorithms. Finally, Figure 5-4 details each choice-model's estimated objective value in different choice-model levels for the best algorithm, mini-batch GM.

As we can see from Figure 5-2, mini-batch SPSA algorithm (purple line) has the worst performance, which converges slower than other four algorithms and has the highest log of objective function value as 19.8 in the end. Also, its log of objective function value does not change very much between iteration 5 and iteration 13, which

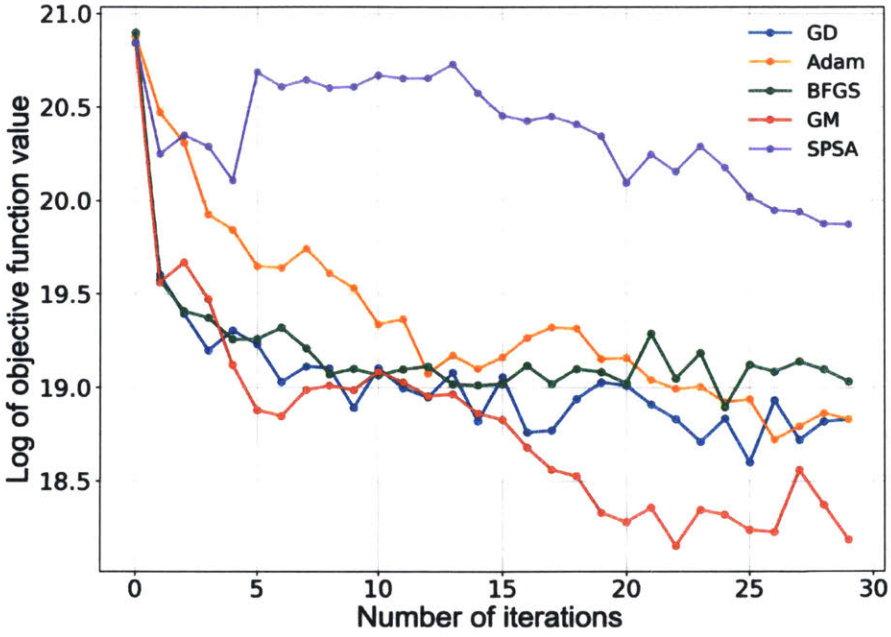
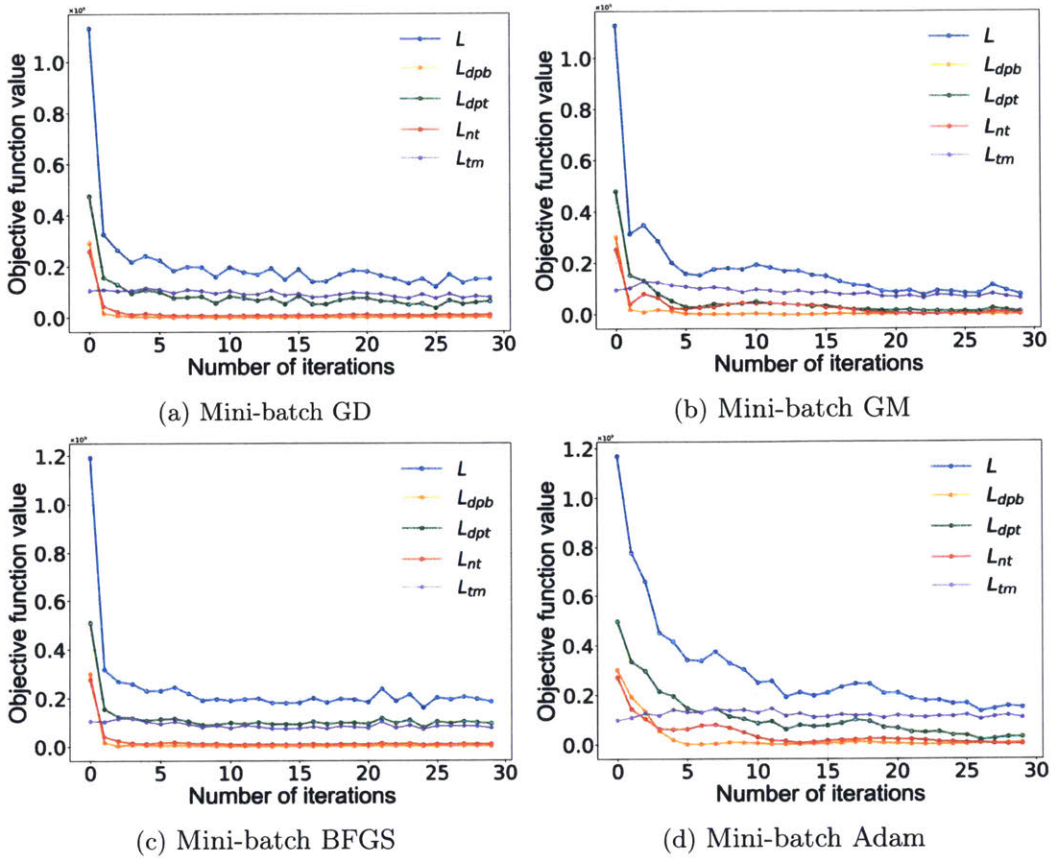


Figure 5-2: Comparison of convergence across different algorithms. As we can see, in the end, mini-batch GM performs the best and mini-batch SPSA performs the worst. Other three algorithms have similar performances.

implies it is stuck in a local minimal. Mini-batch GM algorithm (red line) has the best performance, which minimizes the objective function value with respect to the actual aggregate statistics by 83% in first 5 iterations and has the lowest log of objective function value as 18.2 in the end. It minimizes the 2-norm difference between the simulated aggregate statistics vector and the actual aggregate statistics vector by 62.4% in first 5 iterations and 68.3% in the end. Among the rest three algorithms, mini-batch BFGS (green line) and mini-batch GD (blue line) converge in similar paths, but mini-batch GD has slightly smaller log of objective function value as 18.8 in the end. Finally, mini-batch Adam (yellow line) converges slower than mini-batch GD and mini-batch BFGS at first but converges to the same value as mini-batch GD in the end.

We can have a closer look at each objective function component's estimated value by different algorithms in Figure 5-3. We plot the estimated objective function value of full population (blue line), the *dpb* choice-model level's estimated objective function value (yellow line), the *dpt* choice-model level's estimated objective function value



(a) Mini-batch GD

(b) Mini-batch GM

(c) Mini-batch BFGS

(d) Mini-batch Adam

Figure 5-3: Detailed objective function component values plots by four algorithms. As we can see, *tm* choice-model's objective function value does not converge significantly as other choice-models across all four algorithms, which implies it may be stuck in local minimum.

(green line), the *nt* choice-model level's estimated objective function value (red line), and the *tm* choice-model level's estimated objective function value (purple line). As shown in Figure 5-3, the estimated objective function values of ***dpb*** choice-model and *nt* choice-model converge close to zero in the end across all four algorithms. For the estimated objective function value of *dpt* choice-model with respect to the actual aggregate statistics of ***dpt*** choice-model, unlike mini-batch GD and mini-batch BFGS that do not decrease it much after first few iterations, mini-batch Adam minimizes it by 93.87% in the end and mini-batch GM minimizes it by 97.6% in the end. For the 2-norm difference between the simulated aggregate statistics of *dpt* choice-model level and the actual aggregate statistics of ***dpt*** choice-model level, mini-batch Adam

minimizes it by 72.6% in the end and mini-batch GM minimizes it by 76.8% in the end. Finally, for the estimated objective function value of *tm* choice-model, none of algorithm has it converged significantly, which may because it is stuck in local minimum.

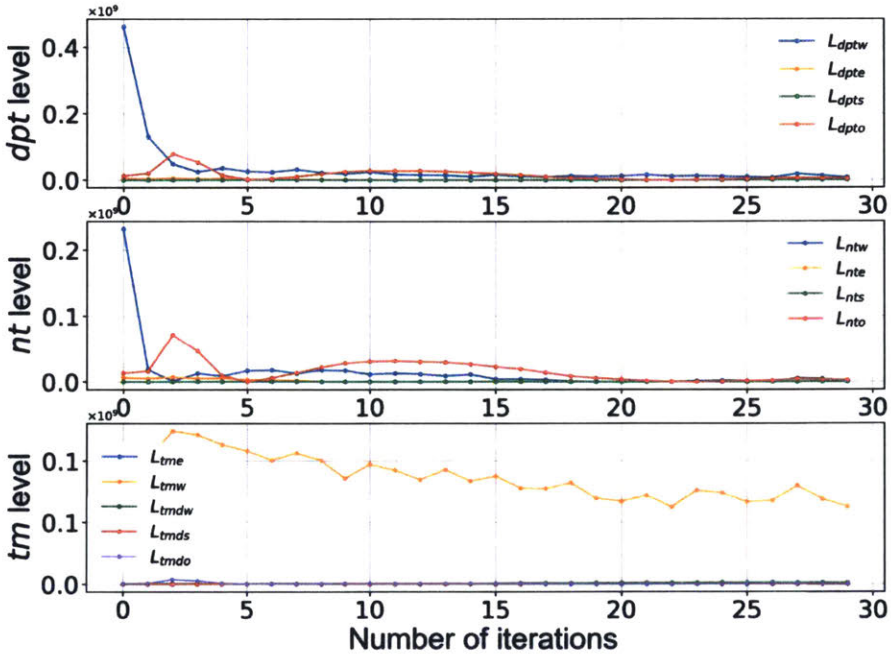


Figure 5-4: Mini-batch gradient with momentum objective function value by tour purpose. In each choice-model level, we show the change of estimated objective function value of each tour purpose. For example, L_{dptw} represents the sum of estimated objective function value of ***dpt*** choice-model's all alternatives containing $Work$ tour purpose, which is expressed as $w_{dptwork}(S_{dptwork}(\boldsymbol{\beta}) - A_{dptwork})^2$.

Then, as we can see in Figure 5-4, in ***dpt*** level, $Work$ tour purpose has the highest initial objective function value with respect to the actual observed aggregate statistics, but mini-batch GM minimizes it by 92.35% in the first 5 iterations and minimizes it by 98.35% in the end. In terms of the 2-norm difference between $S_{dptwork}$ and $A_{dptwork}$, mini-batch GM minimizes it by 76.9% in the first 5 iterations and minimizes it by 88.5% in the end. In *nt* level, $Work$ tour purpose still has the highest initial objective function value with respect to the actual aggregate statistics; Mini-batch GM minimizes it by 96.22% in the first 5 iterations and minimizes it by 99.63% in the end. In terms of the 2-norm difference between S_{ntw1} and A_{ntw1}, mini-batch

minimizes it by 73.5% in the first 5 iterations and minimizes it by 95.4% in the end. Although the estimated objective function value of *Other* tour purpose fluctuates a bit in the middle, mini-batch GM minimizes it by 82.25% in the end. In *tm* level, *tmw* choice-model has the highest initial estimated objective function value with respect to the actual aggregate statistics and mini-batch GM only minimizes it by 33.27% in the end. In terms of the 2-norm difference between the simulated aggregate statistics of *tmw* choice-model and the actual aggregate statistics of *tmw* choice-model, mini-batch GM minimizes it by 21.5% in the end. As we mentioned earlier, this may be because it is stuck in local minimum.

Finally, the 2-norm of difference between updated coefficients $\boldsymbol{\beta}$ by mini-batch GM in the end of 30 iterations and original coefficients $\boldsymbol{\beta^0}$ is 1.11, which is a small number considering large initial differences between $S_{\pi i}(\boldsymbol{\beta}^0)$ and $A_{\pi i}(\boldsymbol{\beta})$. This means we do not improve objective function by changing coefficients a lot which implies that we avoid over-fitting.

Chapter 6

Conclusion

In this study, we presented the formulation and solution procedure for calibrating activity-based travel demand model systems. The calibration of the ABM was formulated as a simulation-based optimization problem and stochastic gradient-based solution procedure was proposed to solve it. The proposed solution procedure uses analytical expressions for the objective function and its gradient —whose derivation was presented in detail— which are evaluated through microsimulation. The approach avoids evaluating complex analytical expressions of the *complete* model system or calculating gradient based purely on simulation. In other words, the proposed approach utilizes the existing structure of the model system to calculate the gradient of the objective function efficiently and accurately. This also allows the procedure to explicitly consider the interdependence of various choice-models in ABM —as opposed to myopic calibration of individual models— thus making calibration more effective. Finally, we presented an illustrative example that demonstrates the workings of the solution procedure.

We also conducted a case study where the proposed solution procedure was tested on an existing Day Activity-Schedule model system for a prototype city. The test led to following insights:

1. The proposed approach efficiently estimates the gradient using closed-form expressions through microsimulation. It is able to calculate each individual's

contribution to the gradient along with generating her activity-travel schedule and finally estimates the complete gradient by summing over all individuals at the end of each iteration.

2. The approach was able to minimize all components of the objective function by adopting appropriate weights. In addition, the proposed approach does not require redundant runs like in the case of numerical gradient estimation and the mini-batch approach further improves its computational efficiency.

3. As the best algorithm of the proposed solution procedure, mini-batch GM is able to effectively minimize the objective function value by 93% in only 6 master iterations. Specifically, it decreases objective function value by 86% in the first master iteration.

4. The proposed solution procedure performs much better than the SPSA algorithm in the matter of computational efficiency, stable performance, and fast convergence. The SPSA algorithm decreased the objective function value by only 64% compared to the proposed procedure in 6 master iterations.

There are also a few limitations of solution procedure worth to keep in mind. First, we assume the actual observed aggregate statistics we calibrate against are fixed, which might have day-to-day distributions in reality. Second, we do not consider calibrating supply simulator —a chanllenging problem—in this research. Third, taking advantage of simple logit choice-model probability expressions, microsimulation and Hansen-Hurwitz estimator, simple analytical expressions we can acquire are limited to count based simulated aggregate statistics, like the number of individuals traveling, the number of individuals having 1 work tour, the number of individuals using car and so on. Fourth, the proposed solution procedure can only apply to 1). utility-maximizing and 2). nested logit activity-based model systems because simple analytical expressions we derive relying on nice properties of such ABMs.

Finally, trying other second-order algorithms, testing on a large model systems, including unobserved heterogeneity, and including wider categories of aggregate statistics into the objective function offer some interesting avenues for further investigation.

Bibliography

[1] Soora Rasouli and Harry Timmermans. Activity-based models of travel demand: promises, progress and prospects. *International Journal of Urban Sciences*, 18(1):31–60, 2014.

[2] Harvey J Miller. Modelling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information System*, 5(3):287–301, 1991.

[3] Soora Rasouli and HJP Timmermans. New impetus to theoretical and empirical perspectives on space-time behavior: stochastic representation, dynamic choice sets and activity spaces. *Environment and Planning B: Planning and Design*, 41(6):954–959, 2014.

[4] Moshe E Ben-Akiva and John L Bowman. Activity based travel demand model systems. In *Equilibrium and advanced transportation modelling*, pages 27–46. Springer, 1998.

[5] Moshe Ben-Akivai, John L Bowman, and Dinesh Gopinath. Travel demand model system for the information era. *Transportation*, 23(3):241–266, 1996.

[6] John Lawrence Bowman. *Activity based travel demand model system with daily activity schedules.* PhD thesis, Massachusetts Institute of Technology, 1995.

[7] John Lawrence Bowman. *The day activity schedule approach to travel demand analysis.* PhD thesis, Massachusetts Institute of Technology, 1998.

[8] John L Bowman and Moshe E Ben-Akiva. Activity-based disaggregate travel demand model system with activity schedules. *Transportation research part a: policy and practice*, 35(1):1–28, 2001.

[9] John L Bowman, Mark Bradley, Yoram Shiftan, T Keith Lawton, and Moshe E Ben-Akiva. Demonstration of an activity based model system for portland. In *8th World Conference on Transport Research, Antwerp, Belgium*, 1998.

[10] Ryuichi Kitamura, Satoshi Fujii, et al. Two computational process models of activity-travel behavior. *Theoretical foundations of travel choice modeling*, pages 251–279, 1998.

[11] Chandra R Bhat, Jessica Y Guo, Sivaramakrishnan Srinivasan, and Aruna Sivakumar. Comprehensive econometric microsimulator for daily activity-travel patterns. *Transportation Research Record*, 1894(1):57–66, 2004.

[12] Khandker Nurul Habib. A comprehensive utility-based system of activity-travel scheduling options modelling (custom) for worker's daily activity scheduling processes. *Transportmetrica A: Transport Science*, 14(4):292–315, 2018.

[13] Theo Arentze and Harry Timmermans. *Albatross: a learning based transportation oriented simulation system.* Citeseer, 2000.

[14] Tommy Gärling, Mei-po Kwan, and Reginald G Golledge. Computational-process modelling of household activity scheduling. *Transportation Research Part B: Methodological*, 28(5):355–364, 1994.

[15] Ram M Pendyala, Ryuichi Kitamura, Cynthia Chen, and Eric I Pas. An activity-based microsimulation analysis of transportation control measures. *Transport Policy*, 4(3):183–192, 1997.

[16] Joshua Auld and Abolfazl Mohammadian. Framework for the development of the agent-based dynamic activity planning and travel scheduling (adapts) model. *Transportation Letters*, 1(3):245–255, 2009.

[17] James C Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest*, 19(4):482–492, 1998.

[18] Andreas Horni, Kai Nagel, and Kay W Axhausen. *The multi-agent transport simulation MATSim.* Ubiquity Press London:, 2016.

[19] Rounaq Basu, Andrea Araldo, Arun Prakash Akkinepally, Bat Hen Nahmias Biran, Kalaki Basak, Ravi Seshadri, Neeraj Deshmukh, Nishant Kumar, Carlos Lima Azevedo, and Moshe Ben-Akiva. Automated mobility-on-demand vs. mass transit: A multi-modal activity-driven agent-based simulation approach. *Transportation Research Record: Journal of the Transportation Research Board (Online)*, 2018.

[20] Nicholas Fournier, Siyu Chen, Isabel Hemerly Viegas de Lima, Zach Needell, Aikaterini Deliali, Andrea Araldo, A Arun Prakash, Carlos Lima Azevedo, Eleni Christofa, Jessica Trancik, et al. Integrated simulation of activity-based demand and multi-modal dynamic supply for energy assessment. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2277–2282. IEEE, 2018.

[21] Parsons Brinckerhoff. 2010 base year update and validation of the nymtc new york best practice model (nybpm). Technical report, New York Metropolitan Transportation Council, 2014.

[22] Parsons Brinckerhoff Consult. The morpc travel demand model: Validation and final report. *Prepared for the Mid-Ohio Region Planning Commission*, 2005.

[23] Parsons Brinckerhoff. Activity-based model calibration report: Coordinated travel âĂŞ regional activity based modeling platform (ct-ramp). Technical report, Atlanta Regional Commission, 2017.

[24] Torgil Abrahamsson. Estimation of origin-destination matrices using traffic counts-a literature survey. 1998.

[25] Ennio Cascetta and Maria Nadia Postorino. Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks. *Transportation science*, 35(2):134–147, 2001.

[26] Ramachandran Balakrishna. *Off-line calibration of dynamic traffic assignment models*. PhD thesis, Massachusetts Institute of Technology, 2006.

[27] Lu Lu, Yan Xu, Constantinos Antoniou, and Moshe Ben-Akiva. An enhanced spsa algorithm for the calibration of dynamic traffic assignment models. *Transportation Research Part C: Emerging Technologies*, 51:149–166, 2015.

[28] Constantinos Antoniou, Carlos Lima Azevedo, Lu Lu, Francisco Pereira, and Moshe Ben-Akiva. W–spsa in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Procedia*, 7:233–253, 2015.

[29] Carolina Osorio. High-dimensional offline od calibration for stochastic traffic simulators of large-scale urban networks. Technical report, Technical report, Massachusetts Institute of Technology. Under review. Available at: http://web. mit. edu/osorioc/www/papers/osoODCalib. pdf, 2017.

[30] Chao Zhang, Carolina Osorio, and Gunnar Flötteröd. Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B: Methodological*, 97:214–239, 2017.

[31] Constantinos Antoniou, Moshe Ben-Akiva, and Haris N Koutsopoulos. Nonlinear kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):661–670, 2007.

[32] A Arun Prakash, Ravi Seshadri, Constantinos Antoniou, Francisco C Pereira, and Moshe Ben-Akiva. Improving scalability of generic online calibration for real-time dynamic traffic assignment systems. *Transportation Research Record: Journal of the Transportation Research Board (Online)*, 2018.

[33] A Arun Prakash, Ravi Seshadri, Constantinos Antoniou, Francisco C Pereira, and Moshe E Ben-Akiva. Reducing the dimension of online calibration in dynamic traffic assignment systems. *Transportation Research Record: Journal of the Transportation Research Board*, (2667):96–107, 2017.

[34] John L Bowman and Moshe E Ben-Akiva. Activity-based disaggregate travel demand model system with activity schedules. *Transportation research part a: policy and practice*, 35(1):1–28, 2001.

[35] Chandra Bhat, Jessica Guo, Sivaramakrishnan Srinivasan, and Aruna Sivakumar. Comprehensive econometric microsimulator for daily activity-travel patterns. *Transportation Research Record: Journal of the Transportation Research Board*, (1894):57–66, 2004.

[36] Isabel Viegas de Lima, Mazen Danaf, Arun Akkinepally, Carlos Lima De Azevedo, and Moshe Ben-Akiva. Modeling framework and implementation of activity-and agent-based simulation: An application to the Greater Boston Area. *Transportation Research Record: Journal of the Transportation Research Board (Online)*, 2018.

[37] William Davidson, Robert Donnelly, Peter Vovsha, Joel Freedman, Steve Ruegg, Jim Hicks, Joe Castiglione, and Rosella Picado. Synthesis of first practices and operational research approaches in activity-based travel demand modeling. *Transportation Research Part A: Policy and Practice*, 41(5):464–488, 2007.

[38] John L Bowman, Mark A Bradley, and J Gibb. The sacramento activity-based travel demand model: estimation and validation results. In *European Transport Conference, Strasbourg, France, September 18-20, 2006*, 2006.

[39] Cambridge Systematics Inc. Fsutms-cube framework phase ii: Model calibration and validation standards. Technical report, Florida Department of Transportation Systems Planning Office, 2008.

[40] John L Bowman, Mark Bradley, Joe Castiglione, and Supin L Yoder. Making advanced travel forecasting models affordable through model transferability. In *93rd Transportation Research Board Annual Meeting, Washington DC, USA, January 12-16, 2014*, 2014.

[41] The Corradino Group. Southeast regional planning model 6.5: Model data, calibration and validation. Technical report, Florida Department of Transportation, 2008.

[42] Daniel Goldfarb. Nchrp 08-36: Evaluating and communicating model results: Guidebook for planners. Technical report, AASHTO Standing Committee on Planning, 2010.

[43] Mario Cools, Elke Moons, and Geert Wets. Calibrating activity-based models with external origin-destination information: overview of possibilities. *Transportation Research Record: Journal of the Transportation Research Board*, 2175(1):98–110, 2010.

[44] Ali Najmi, Taha H Rashidi, and Eric J Miller. A novel approach for systematically calibrating transport planning model systems. *Transportation*, pages 1–36, 2018.

[45] Laura Schultz, Vadim Sokolov, Josh Auld, Dominik Karbowski, and Aymeric Rousseau. Optimization to fuse large scale transportation data sets into simulation models. In *INFORMS. Houston, Texas, USA, October 22-25, 2017*, 2017.

[46] Gunnar Flötteröd, Michel Bierlaire, and Kai Nagel. Bayesian demand calibration for dynamic traffic simulations. *Transportation Science*, 45(4):541–561, 2011.

[47] Gunnar Flötteröd, Yu Chen, and Kai Nagel. Behavioral calibration and analysis of a large-scale travel microsimulation. *Networks and Spatial Economics*, 12(4):481–502, 2012.

[48] Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[49] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[52] Julius R Blum et al. Approximation methods which converge with probability one. *The Annals of Mathematical Statistics*, 25(2):382–386, 1954.