Technische Universität München

Master's Thesis

Dynamic Traffic Assignment Model Calibration Using Islands Genetic Algorithm

by

Yijiong Zhu

supervised by

Dr. Tao Ma

October 23, 2019

Uni.-Prof. Dr. Constantinos Antoniou
Chair of Transportation Systems Engineering
Technical University of Munich
Arcisstr. 21
80333 Munich

Dr. Martin Streichfuss Dr. Jörg Esser
Think: Act HUB
Roland Berger GmbH
Trichtergasse 2
52064 Aachen

Abstract

Dynamic Traffic Assignment (DTA) models are widely in used in transportation system management. Calibration is a crucial step to improve reliability and accuracy of DTA models. We present a systematic framework to offline calibrate the supply and demand component of a DTA model. The essence of DTA model calibration is an optimization problem, aiming to minimize the discrepancy between field conditions and simulated traffic measurements. Instead of relying on a single traffic measurement, a multi-objective function is formulated with different traffic measurements for the supply and demand component respectively. As the calibration process is a nonlinear and stochastic problem, heuristic algorithms are implemented as solution techniques. To overcome the limitations of standard Genetic Algorithm (GA) such as high running time, we introduce the Islands Genetic Algorithm (IGA) to solve the calibration problem. We conduct case studies with a synthetic network and a network of Munich, Germany, to validate the proposed methodology. The promising results indicate that IGA outperforms standard GA in terms of convergence speed and solution quality. Furthermore, we explore the application of Blockchain technology in DTA model calibration.

Acknowledgement

When writing this part in autumn 2019, I approach the destination of my study at TUM.

First of all, I would like to thank Dr. Tao Ma, my university supervisor who supported me to conduct academic research hand by hand, with his extensive research experience and innovative research ideas. He was the most patient teacher I have ever met in my entire university life. Additional from his research guidance, I was deeply influenced by his value on world and life.

I would like to thank Dr. Martin Streichfuss and Jörg Esser, my company supervisor for their valuable advice from the perspective of strategy consulting. In the meeting together with Prof. Dr. Constantinos Antoniou, my university supervisor, I learned a lot from our discussion to bridge the gap between business and academia.

I would like to thank Moied Qurashi for his technical support in programming and transportation modelling.

Last but not least, I would like to thank Zhao Kang for accompanying me over last six years from our 18 to 24 years old, and to my family for their wholehearted love.

Declaration

I hereby confirm that this thesis was written independently by myself without the use of any sources beyond those cited, and all passages and ideas taken from other sources are cited accordingly.

Munich, 23.10.2019

Place and Date

Signature

Table of Contents

CHAPTER 1 INTRODUCTION........1

1.1 STATIC TRAFFIC ASSIGNMENT MODEL........2
1.2 DYNAMIC TRAFFIC ASSIGNMENT MODEL........3
1.2.1 Early Development........3
1.2.2 Analytical DTA Model........3
1.2.3 Simulation Based DTA Model........4
1.3 MOTIVATION........6
1.4 THESIS OUTLINE........7

CHAPTER 2 LITERATURE REVIEW........8

2.1 CALIBRATION SCOPE........9
2.1.1 Supply-Demand Calibration........9
2.1.2 Supply Calibration........11
2.1.3 Demand Calibration........12
2.2 CALIBRATION TECHNIQUES........14
2.2.1 Kalman Filtering........14
2.2.2 Simultaneous Perturbation Stochastic Approximation........14
2.2.3 Machine Learning Techniques........15
2.2.4 Genetic Algorithm........16
2.2.5 Islands Genetic Algorithm........17
2.2.6 Simulated-Based Algorithm........19

CHAPTER 3 METHODOLOGY........20

3.1 CALIBRATION FRAMEWORK........21
3.2 PROBLEM FORMULATION........22
3.3 OPTIMIZATION ALGORITHM........23
3.3.1 Genetic Algorithm........23
3.3.2 Island Genetic Algorithm........25
3.4 SIMULATOR........28
3.4.1 Introduction of SUMO........28
3.4.2 Features of SUMO........29
3.4.3 Workflow of SUMO........29

CHAPTER 4 CASE STUDY........31

4.1 SYNTHETIC CASE STUDY........32
4.1.1 Network Description........32
4.1.2 Calibration Parameter........33
4.1.3 Calibration Process........33
4.1.4 Calibration Result........35

4.2 MUNICH NETWORK CASE STUDY 42
4.2.1 Network Description 42
4.2.2 Calibration Parameter..43
4.2.3 Calibration Process 43
4.2.4 Calibration Result 45

CHAPTER 5 BLOCKCHAIN 52

5.1 NEW MOBILITY MODE 53
5.1.1 Urban Air Mobility 53
5.1.2 Autonomous Vehicle 55
5.1.3 Shared Mobility..56
5.2 BLOCKCHAIN 57

CHAPTER 6 CONCLUSION 59

LIST OF REFERENCES..60

APPENDIX..70

List of Figures

1.1	DTA Structure
3.1	DTA Model Calibration Framework
3.2	Genetic Algorithm Procedure
3.3	Islands Genetic Algorithm Flowchart
3.4	Traditional Island Model
3.5	Three Types of Island Model
3.6	SUMO Workflow
4.1	The Synthetic Network
4.2	The Synthetic Case Study: Performance of IGA
4.3	The Synthetic Case Study: Performance of Island 1
4.4	The Synthetic Case Study: Performance of Island 2
4.5	The Synthetic Case Study: Performance of Island 3
4.6	The Synthetic Case Study: Performance of Standard GA
4.7	The Munich Network
4.8	The Munich Case Study: Performance of IGA
4.9	The Munich Case Study: Performance of Island 1
4.10	The Munich Case Study: Performance of Island 2
4.11	The Munich Case Study: Performance of Island 3
4.12	The Munich Case Study: Performance of Island 4
4.13	The Munich Case Study: Performance of Standard GA
5.1	Two Scenarios of UAM Simulation
5.2	UAM Structure of Munich Case Study
5.3	A Blockchain Approach for Multi-Modality in Belgium
5.4	Application of Blockchain in DTA Model Calibration

List of Tables

3.1 SUMO Components

4.1 IGA Configuration for Synthetic Network

4.2 GA Configuration for Synthetic Network

4.3 IGA Configuration for Munich Network

4.4 GA Configuration for Munich Network

List of Abbreviations

AFWS	Auxiliary Feedwater System
AV	Autonomous Vehicle
CBD	Central Business District
CEKF	Constrained Extended Kalman Filter
c-SPSA	Cluster-wise SPSA
DLR	German Aerospace Center
DSA	Dynamic Spectrum Access
DTA	Dynamic Traffic Assignment
DUE	Dynamic User Equilibrium
DUO	Dynamic User Optimal
EGM	Extended Gradient Method
EKF	Extended Kalman Filter
GA	Genetic Algorithm
GASA	Genetic Algorithm with Simulating Annealing
GUI	Graphical User Interface
IGA	Islands Genetic Algorithm
LimEKF	Limiting EKF
MAP	Maximum a Posteriori
MSE	Mean Squared Error
NPP	Nuclear Power Plant
NSGA-II	Nondominated Sorting Genetic Algorithm II
OD	Origin-Destination
OSM	Open Street Map
PCA	Principal Components Analysis
PC-SPSA	SPSA with Principal Components Analysis
RMSE	Root Mean Squared Error

ROW	Right of Way
SA	Simulated Annealing
SPEA2	Strength Pareto Evolutional Algorithm II
SPEA2+	Strength Pareto Evolutional Algorithm II+
SPSA	Simultaneous Perturbation Stochastic Approximation
SO	System Optimum
SUE	Stochastic User Equilibrium
SUMO	Simulation of Urban Mobility
TAZ	Traffic Analysis Zones
TraCI	Traffic Control Interface
UAM	Urban Air Mobility
UE	User Equilibrium
UKF	Unscented Kalman Filter
W-SPSA	Weighted SPSA (W-SPSA)

Chapter 1 Introduction

Transportation system is a complex system in which goods, individuals, and environment interact with each other. To manage and operate transportation system effectively, Dynamic Traffic Assignment (DTA) models are widely used to estimate and predict traffic state dynamically. Calibration is an essential step to improve reliability of DTA models. The calibration process can be performed in offline or online context. In offline context, calibrated DTA models represent historical traffic patterns. Online calibration uses real-time data to capture dynamic traffic state. In this chapter, we give the overview of the static traffic assignment model and discuss its drawbacks. Then, we summarize the development of DTA models from the earlier stage to the analytical and simulation-based model. In the end of the chapter, we explain the motivation of this research and demonstrate the outline of this thesis.

1.1 Static Traffic Assignment Model

Static traffic assignment models are based on traditional four-step procedure: trip generation, trip distribution, mode choice, and route assignment. Trip generation predicts trip frequency of origin or destination for traffic analysis zones (TAZ), considering socio-economic data such demographic features and travel activities. The corresponding Origin-Destination (OD) pairs are matched in the trip distribution that displays travelers from each origin to each destination. Mode choice determines trips with specific transportation modes. The final step is route assignment that refers to selecting routes in the network. The underlying assumption for static traffic assignment models is the steady OD flows. Popular assignment techniques include All-or-Nothing assignment, User Equilibrium (UE), and System Optimum (SO). All-or-Nothing traffic assignment approach allocates each trip between OD pairs with minimum travel cost. This method assumes unchanged travel time, without considering link capacity. UE assumes that travelers in the network have perfect knowledge about the travel cost. Under this assumption, travelers cannot reduce travel cost through route change. Under SO assignment, total system cost is minimized by cooperation and communication between drivers. In transportation planning and management, static traffic assignment models can be useful in long-term projects such as transportation infrastructure construction and land use planning. To capture time-varying transportation network, static traffic assignment models are unbale to capture driver behaviors. The modeling result of static assignment models is unreliable in the real time traffic management.

1.2 Dynamic Traffic Assignment Model

1.2.1 Early Development

The earlier attempts of DTA model are referred as "quasi-dynamic" assignment. Peeta and Mahmassani (1995) presented two assignment models with the assumption that the OD demand under SO and UE were known. In their research, the planning horizons were divided into several time intervals in which static assignment models were deployed. The solution algorithms for the two assignment models were based on an iterative search approach to improve the overall system performance. However, the division of interval was arbitrary: short interval was computational inefficient and long interval failed to capture the change of the network effectively. The use of static model for divided time intervals ignored the interaction between time horizons.

The earlier efforts of DTA model moves from static traffic assignment models one step forward. But these models are still unable to reflect real-time change in traffic network and cannot precisely represent traffic pattern in a dynamic context.

1.2.2 Analytical DTA Model

Analytical DTA models reply on mathematical formulations to reach equilibrium conditions such as UE and SO. The analytical DTA model is solved by conventional algorithms that involve gradient calculation.

Merchant and Nemhauser (1978), the pioneers of analytical DTA model, developed its formal theory in initial stage. A piecewise linear model with a single destination was solved by decomposition techniques. The global optimum of the single OD model was solved by one-pass simplex algorithm. However, such analytical model failed to deal with multiple destinations and solution techniques were not computational efficient.

With increasing complexity of network scale, analytical DTA models fail to replicate real-world situation. Analytical models may suit for modelling small traffic network, but with the increasing scale of the traffic network, the mathematical formulation is hard to perform. The demand to fully capture the individual behaviors under dynamic conditions in real-world called the development of simulation based DTA model.

1.2.3 Simulation Based DTA Model

The state-of-art DTA models are simulation based. Based on resolution of simulation scale, simulation-based models are divided into three categories: microscopic models, mesoscopic models, and macroscopic models. Microscopic models simulate at the highest level of detail, modelling individual driving behaviors and interactions between drivers. Individual driving behaviors include car-flowing, lane-changing, and acceleration and deceleration. Car-flowing models represent vehicle positions in a continuous traffic stream and describe actions of drivers when follow other vehicles. Macroscopic models treat traffic stream as a whole, formulating homogeneous or heterogeneous traffic flow. Mesoscopic simulation models are combinations of microscopic and macroscopic models, describing individual vehicles in a continuum traffic flow. There are various simulation tools available. PTV AG developed VISSIM for microscopic simulation and VISUM for macroscopic simulation. Other popular simulation tools include SUMO, AIMSUN, and Dyno-MIT.

The basic structure of a simulation based DTA model consists of a supply component and a demand component, as shown in Figure 1.1 (Balakrishna et al., 2007b). The supply component represents network characteristics and simulates individual behaviors including lane changing and car following. The demand component includes Origin-Destination (OD) flows.

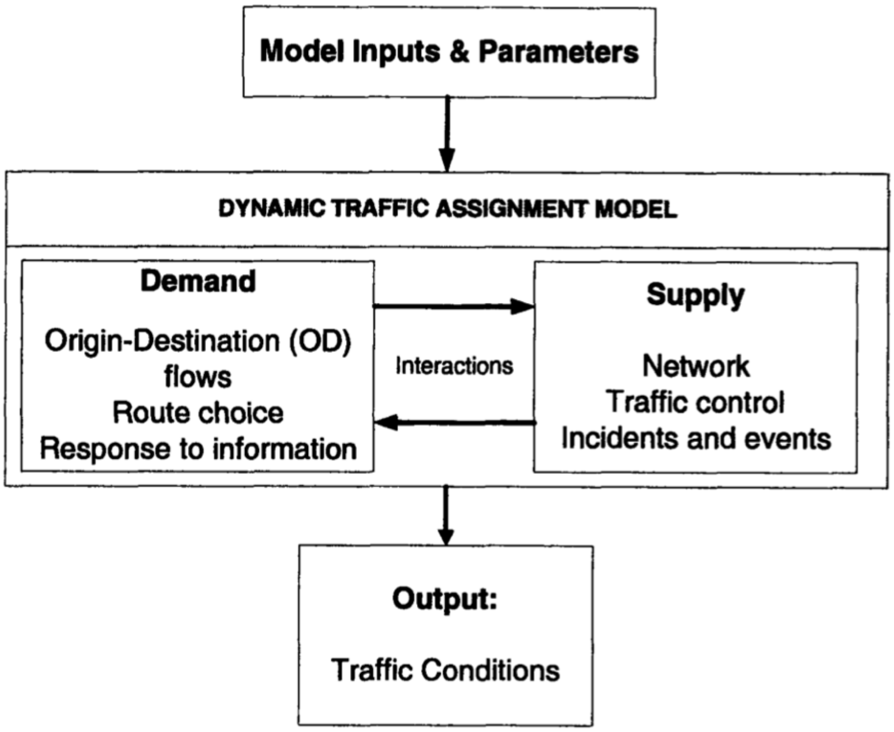

Figure 1.1 DTA Structure

1.3 Motivation

The motivation of this research is to provide a systematic framework for offline DTA model calibration and to give corresponding solution techniques. The purpose of DTA model calibration is to minimize the discrepancy between observed and simulated traffic measurements. Instead of relying on a single traffic measurement such as the speed density relationship (Chiappone et al., 2016), we intend to use different traffic measurements for the supply and demand component, respectively. We follow an iterative procedure to jointly calibrate supply and demand parameters. Considering the complex nature of DTA model calibration, a multi-objective function is formulated with supply parameters as driver behaviors and link speeds and demand parameters as OD flows. This research contributes to the literature by implementing the Islands Genetic Algorithm (IGA) as the solution algorithm for the supply calibration. The application of IGA in DTA model calibration is very sparse despite of advantages such as avoiding premature convergence and improving computational efficiency. In this research, we use the same configuration for both IGA and GA and evaluate the robustness of IGA regarding standard GA as a benchmark in terms of convergence speed and solution quality. Two case studies were used to demonstrate the proposed methodology. Furthermore, we propose a blockchain approach in DTA model calibration, involving emerging transportation modes.

1.4 Thesis Outline

The thesis is organized as follows. The chapter 2 summarizes literatures in DTA model calibration by discussing the calibration scope and solution techniques. The chapter 3 provides the methodology of the calibration approach. The chapter 4 introduces two case studies using a synthetic network and a real-world network respectively. The chapter 5 explores the application of the blockchain technology in the DTA model calibration. The chapter 6 concludes the research findings and gives future research ideas.

Chapter 2 Literature Review

DTA model calibration involves determining the calibration framework, formulating the optimization problem, and the selecting optimization technique. Previous work focused on calibrating either supply parameters or demand parameters, while recent work deals with calibrating supply and demand parameters jointly. In jointly calibrating DTA model, either an iterative approach or a simultaneous approach can be followed. In this chapter, we first give the overview of DTA model calibration framework in three sections: supply-demand calibration, supply calibration, and demand calibration.

The nature of DTA model calibration is a complex stochastic problem. Traditional optimization algorithms with gradient calculation are not appropriate for such application. Various solution techniques without calculating derivatives are used for solving calibration problem, including the Kalman filtering algorithm, heuristic algorithms, and their variants. Among various DTA solution algorithms, GA is widely used in solving the complex optimization problem. We give the overview of different calibration techniques, in which the applicability of GA and its drawbacks of the standard GA are discussed, and then move to the IGA.

2.1 Calibration Scope

2.1.1 Supply-Demand Calibration

To jointly calibrate supply and demand parameters, the calibration framework of a DTA model can follow either an iterative approach or a simultaneous approach.

An iterative calibration approach proposed by Toledo et al. (2004) consisted of two steps, starting with using aggregate data to estimate individual parameters such as driving behavior and route choice parameters, and following by using aggregate data to calibrate the simulation model as a whole. In calibrating the whole simulation model, a sequential solution approach was used: calibration of OD flows with fixed individual parameters and recalibration of individual parameters with fixed OD matrix.

Due to complex interactions among various parameters of DTA models, Balakrishna et al. (2007b) proposed a feasible methodology for offline calibration of DTA models, estimating supply and demand parameters simultaneously. The author calibrated the supply parameters including link capacities and speed–density relationships and the demand parameters including OD flows, based on a case study on a network of Los Angeles, USA.

In the same year, Balakrishna et al. (2007a) presented a methodology to simultaneously calibrate supply (i.e., car-following and lane changing behaviors) and demand (i.e., OD flows) parameters of a microscopic simulation model. The systematic calibration process in a microscopic simulator worked like a black-box, not relying on the traditional four-step transportation model. A case study using a network in Lower Westchester Country, New York, in this research indicated that even with limited sensor data, the calibration framework estimated all parameters efficiently.

Vaze et al. (2009) calibrated DTA model using multiple data sources: conventional data from loop detector and disaggregated data from emerging Automatic Vehicle Identification (AVI) technology. The author employed a microscopic simulator and two solution techniques (GA and SPSA) to jointly calibrate supply and demand parameters. The methodology was validated by a small synthetic network and a real network in Lower Westchester Country, New York, demonstrating that using multiple data sources significantly improved model accuracy, compared with using only loop detector data.

Antoniou et al. (2011) presented offline and online DTA model by calibrating supply and demand parameters to replicate travel behavior and traffic dynamics in real world. The offline DTA calibration generally required data in the previous such as archived sensor data and data from emerging technologies including AVI systems. Based on the calibration results of offline DTA model, online calibration used small amount of real time data and a few estimated parameters. The case studies with network in Los Angeles, USA and Southampton, UK identified the proposed methods.

Seyedabrishami et al. (2014) calibrated a macroscopic DTA model in the off-line context with supply and demand parameters under an iterative approach. The demand parameter (time dependent OD matrix) was estimated by a bi-level optimization using an Extended Gradient Method (EGM). Then the fine-tuned supply parameters were adjusted by trial-and-error method to match the congestion pattern following a three-stage approach: single segment, sub-network, and the entire network. The proposed methodology was applied in Ardebil, Iran. Two calibration scenarios were presented: demand calibration only and joint supply and demand calibration. The promising calibration results indicated that the offline calibrated OD matrix can be used as a priori estimators for online calibration and jointly supply-demand calibration outperformed the demand only calibration.

2.1.2 Supply Calibration

Abundant literatures focused on calibrating supply parameters. Qin and Mahmassani (2004) presented an operational framework for DTA model calibration with a transfer function in which lagged response of speed-density relationship were recognized and system noises were autocorrelated. The model input was the differenced equilibrium speed given density level and the model output was the differenced speed. The calibration results of Irvine network showed that adaptive DTA model with a transfer function outperformed static model such as Greenshields model. In the context of real-time estimation and prediction, adaptive DTA model was more preferable and reliable.

Flötteröd et al. (2011) presented a framework for DTA demand model calibration in individual level with Bayesian estimation where traffic measurements (traffic counts) were combined with a dynamic simulator-based of the modeler's prior knowledge. The proposed methodology overcomes limitations in DTA model calibration such as many oversimplified assumptions and is compatible with equilibrium-based model. A case study using Zurich, Switzerland with calibration parameters of individual behaviors validated the effectiveness of the proposed methodology.

Gangi (2011) extended a mesoscopic DTA model in the context of emergency conditions. In the modified DTA model, various indicators to quantify risk exposures of hazardous event were developed and multimode network was introduced. The proposed DTA model was able to handle large capacities of queue spillover and was applicable under various scenarios in the urban area. The case study with a network from a southern city in Italy evaluated the proposed methodology and identified that the extended mesoscopic DTA model was useful in decision making process of urban planning and management in order to mitigate potential risks.

Ben-Akiva et al. (2012) enhanced a mesoscopic simulation based DTA model to replicate highly congested urban networks with short links, complex intersections, and long queues. Within the Case study in Beijing network, the author considered overlapping routes, queue length, and spillbacks and captured pedestrians and bicycles impacts of the DTA model, inputting surveillance traffic data such as traffic counts and travel times. The calibrated results showed that the enhanced DTA model improved efficiency in traffic management and traffic planning for transportation authorities.

2.1.3 Demand Calibration

Abundant literatures dealt with calibrating demand parameters. Mahut et al. (2004) presented a calibration methodology in which DTA model iteratively reassign flow to paths and OD matrix of DTA model was calculated by turning counts from a trip generation and trip distribution model. The case study network using Calgary, Canada with precise traffic signal plans was developed and calibrated. The corresponding results indicated that the proposed methodology enhanced turning movement flows and path flows.

Zhou and Taylor (2014) presented a queue based mesoscopic DTA model to enable quick simulation of a highly congested network. The mesoscopic DTA model incorporated Newell's simplified kinematic wave model and OD demand calibration system to estimate dynamic flow. The OD demand matrix was determined by aggregate flow counts from queue points. The test results of a triangle regional model from Raleigh, North Carolina showed that the queue based DTA model was highly potential to facilitate real-time prediction in traffic management.

Lu et al. (2015) proposed Weighted SPSA (W-SPSA), incorporating a weight matrix with SPSA to add network information in terms of spatial and temporal correlations between model parameters and observed traffic measurements. W-SPSA, with the added correlation weight matrix helps in reducing the noise due to uncorrelated measurements. The W-SPSA can reduce the noise of traditional SPSA by considering spatial-temporal correlation with a weighted matrix. A synthetic network and the expressway network of Singapore were used to compare the performance of traditional SPSA and W-SPSA. The results showed that W-SPSA outperformed traditional SPSA in terms of convergence rate, model accuracy, and robustness due to the presence of a weighted matrix which reduced the computational dimensions and gradient error.

Cluster-wise SPSA (c-SPSA) proposed by Tympakianaki et al. (2015), clusters the OD flows to reduce the gradient biasness by estimating the gradient separately for each cluster. The author proposed clustering based on spatial interactions of OD pairs using free-flow travel times to combine uncorrelated OD flows in a cluster to reduce the non-linearity in estimation problem.

Recently, Qurashi et al. (2019) proposed PC-SPSA by combining SPSA with Principal Components Analysis (PCA) to reduce the estimation problem dimensions and non-linearity significantly. Within PC-SPSA, the estimation variables are transformed into lower dimension PC-scores based on the PC-directions estimated from the historical estimates of OD flows. A case study used a network of Vitoria, Spain, where PC-SPSA was able to improve calibration results significantly in terms of having faster convergence and finding better solutions.

2.2 Calibration Techniques

2.2.1 Kalman Filtering

Antoniou et al. (2007) presented an online calibration approach involving the three extensions of the Kalman filtering algorithm including the extended Kalman filter (EKF), the limiting EKF (LimEKF), and the unscented Kalman filter (UKF). The calibration results of a Southampton network in UK demonstrated that LimEKF significantly improves modeling accuracy and outperforms both EKF and UKF.

Zhang et al. (2017) extended this research by presenting a new Constrained Extended Kalman Filter (CEKF) which computes the maximum a posteriori (MAP) estimates, to overcome problems in large-scale network calibration.

2.2.2 Simultaneous Perturbation Stochastic Approximation

SPSA proposed by Spall (1998), is part of stochastic approximation algorithms which are majorly used for large-scale, non-linear and stochastic problems having expensive objective function evaluations and noisy measurements. Within SPSA, the estimation variables are perturbed simultaneously and only two objective function evaluations are required to estimate the gradient (Tympakianaki et al., 2018). Once the gradient is evaluated, it is used to minimize the estimation variables by predefined step size.

Many variants of SPSA are used in DTA model calibration. For example, PC-SPSA, a Combination of SPSA and Principal Components Analysis (PCA) (Qurashi et al., 2019). First, a set of PC-directions are estimated using a data matrix of historical estimates using singular value decomposition. Then, a set of PC-directions were reduced due to a property of PCA capturing most of the variance within the first few

PC components. The reduced PC-directions were used to transform estimation variables to a set of PC-scores. These PC-scores are then estimated using SPSA instead of directly estimating the estimation variables. Also, to use PC-scores for estimation using SPSA the perturbation and minimization is done as a percentage change in PC-scores instead of addition or subtraction in generic SPSA.

2.2.3 Machine Learning Techniques

Machine Learning techniques are used in DTA model calibration. Antoniou and Koutsopoulos (2006) used machine-learning techniques (k-means, clustering, k-nearest-neighborhood classification, and locally weighted regression) in estimation of speed-density relationship in the DTA model, especially in the simulation-based DTA model. The proposed methodology overcomes the limitations of classic paradigm for estimation speed-density relationship which is based on traffic flow theory. The case study of Irvine, California demonstrated that although machine leaning based approach did not provide clear insight of the traffic flow theory, it was more practical and efficient than classic method and was more flexible in using multiple data sources.

Flötteröd et al. (2011) presented a DTA calibration framework using machine learning techniques for the Melbourne, Australia network in the case study, simulating over 2 million travelers. Machine Learning techniques was applied to calibrate the fundamental diagram of traffic flow with historical observation traffic data. Other factors that influenced simulation results such as pedestrians and cyclists in the Central Business District (CBD) were considered. The calibration results indicated that the proposed methodology largely improved calibration quality, as indicated by 30% improvement in Root Mean Squared Error (RMSE). Therefore, the calibrated DTA model can represent the real-world traffic state and can apply in traffic management and operation in Australia.

2.2.4 Genetic Algorithm

Among various optimization techniques, GA is widely used in DTA model calibration (Kim et al., 2005). For example, Ma and Abdulhai (2002) applied GA as the solution technique for calibrating a microscopic model, aiming at minimizing discrepancy between observed and simulated measurements of traffic counts. The corresponding results in their research showed robustness of GA for solving calibration problem.

Runmei Li and Wei Li (2005) explored the GA in DTA model with variational equality and physical queue in the early stage. The author indicated that GA was in advantage of other local search methods in terms of the simple application and not requiring preliminary knowledge about the optimization problem. The Dynamic User Optimal (DUO) model was used to validate the effectiveness of the GA. By comparing the optimization performance with F-W algorithm, the GA can achieve the equilibrium of the network quicker.

Wismans et al. (2011) formulated the Dynamic Traffic Management problem as a multi-objective optimization problem and applied three evolutional algorithms: nondominated sorting genetic algorithm II (NSGA-II), the strength Pareto evolutional algorithm II (SPEA2), and the strength Pareto evolutional algorithm II+ (SPEA 2+) to obtain optimal solutions. External factors such as noise, climate, and congestion were also optimized. In comparing the optimization results of three proposed algorithms thorough different metrics, SPEA 2+ outperforms SPEA 2 in terms of all applied fitness measurement. However, it was hard to compare the performance of SPEA 2+ and NAGA-II due to the insensitivity to mutation rate in the experiments.

Omrani and Kattan (2013) presented multi-criteria framework for simultaneously calibrating supply and demand parameters using GA which was run in parallel in a computational cluster. A multi-objective optimization problem was formulated, aiming to minimize the difference between estimated and priori OD matrix while involving other factors such as traffic counts, turn counts, and link speeds. The author applied the proposed methodology using a large size network in Toronto, Canada, with 67426 number of supply and demand parameters to be calibrated. The calibration results indicated that GA in high computational cluster was eligible to improve calibration quality in terms of speed and fitness value.

Varia et al. (2013) optimized a congested urban network with signal parameters and DUE traffic assignment. The GA was applied as the optimization technique to find optimal signal settings (signal cycle times, green times, and phase sequence) and distribution factors of traffic flow. In the case study of Mumbai, India, the author explored different GA parameters including crossover and mutation operators to obtain the best solutions. The application of GA successfully obtained DUE condition and optimized signal settings to achieve travel times in the network.

Cobos et al. (2016) proposed a memetic algorithm using NSGA-II for performing global search and Simulated Annealing (SA) for performing local search, called NSGA-II-SA. Based on their case study for calibrating microscopic traffic flow models, the proposed algorithm outperformed Genetic Algorithm with Simulating Annealing (GASA) in terms of runtime and calibration measurements.

2.2.5 Islands Genetic Algorithm

Having realized the limitations of traditional GA including expensive computation (Henderson and Fu, 2004), various extensions of GA have been used to improve computational efficiency. Another limitation of standard GA is the premature convergence because individuals can be trapped in local dilemma.

IGA (Whitley et al., 1999) as a multi-population GA, has potential to overcome local hills and valleys and to provide the global optimum. The existence of semi-isolated islands contributes to genetic diversity due to independent evolution of each island and the migration process between islands to exchange information (Mühlenbein, 1992; Starkweather et al., 1991). The isolated islands avoid inbreeding, a biological behavior of producing off-springs from mating individuals with similar genes (Collins and Jefferson, 1991). Therefore, IGA can maintain genetic diversity in the population and prevent premature convergence compared to standard GA.

However, there is no previous research using IGA for DTA model calibration. The applicability and scalability of IGA have been demonstrated in other fields, such as solving job scheduling problem (Kurdi, 2015) and optimizing the design of satellite separation systems (Hu et al., 2014).

Calégari et al. (1997) applied island genetic algorithm in the field of telecommunication, aiming to optimize the setting cost of the transmitters within a given geographical area. The problem was formulated as an optimization problem to solve setting cover problem (i.e., usage of minimum transmitters to realize maximum area coverage). The IGA compared with standard GA, had advantages of fast execution and higher quality of results, as indicated in their research findings.

Pereira and Lapa (2003a) applied IGA in optimization of nuclear reactor core design. The reactor cell parameters were dimensions, enrichment, and materials. And the restrictions of the optimization problem were thermal flux, criticality, and sub-moderation. The aim of the optimization problem was to minimize the average peak-factor of the nuclear reactor core. The author indicated that the implementation of the IGA can reduce computational effort without relying on high performance computer.

In the same year, Pereira and Lapa (2003b) applied IGA in the field of Nuclear Power Plant (NPP) Auxiliary Feedwater System (AFWS) surveillance tests policy. The aim was to optimize the system availability while considering several realistic factors: the aging effects, revealing failures, and distinct test parameters. The optimization results showed that IGA has excellent performance not only in computational time, but also the solution quality.

Friend et al. (2008) presented an architecture to solve a channel allocation problem in the cognitive network design using IGA. The channel allocation problem was unique to Dynamic Spectrum Access (DSA). The heart of the cognitive problem was the cognitive controller and distributed optimization process. The IGA with the potential to solve computationally challenging problems in the cognitive architecture was used for distributed reasoning. The allocation results showed that the solution provided by IGA was close to the optimal solution in each simulation (25 repeated experiments).

2.2.6 Simulated-Based Algorithm

Many researchers in DTA model calibration applied several new techniques to address the complex problem. Zhang et al. (2017) applied metamodel Simulation-based optimization (SO) algorithms to address a calibration problem for a large-scale network. The metamodel included analytical structural problem-specific information. The proposed algorithm reduced 80% running time on average, validated by a synthetic toy network and a real network from Berlin, Germany. Osorio (2019) applied the SO algorithm to solve high dimensional calibration problem. The experiment results showed a 77% improvement of link counts compared with benchmark methods.

Chapter 3 Methodology

In this chapter, a systematic framework for DTA model calibration with supply and demand parameters is presented. Due to the complexity of DTA model calibration problem, a multi-objective function is formulated to minimize discrepancy between simulation outputs and historical traffic data. Different traffic measurements are used for evaluation of the supply and demand component. The calibration process follows an iterative calibration process: calibration of supply parameters with fixed demand parameters, and recalibration of demand parameters (OD demand) with constant supply parameters. We also give the overview of SUMO, our DTA model simulator.

3.1 Calibration Framework

We propose a systematic framework for offline DTA model calibration, as shown in Figure 3.1. We use archived traffic data to represent historical traffic patterns. A multi-objective function is formulated to evaluate the calibration results (discrepancy between simulated and observed traffic measurements). Different traffic measurements are used for the supply and demand component. We follow an iterative calibration process: calibration of supply parameters with fixed demand parameters and recalibration of demand parameters with constant (calibrated values from previous step) supply parameters.

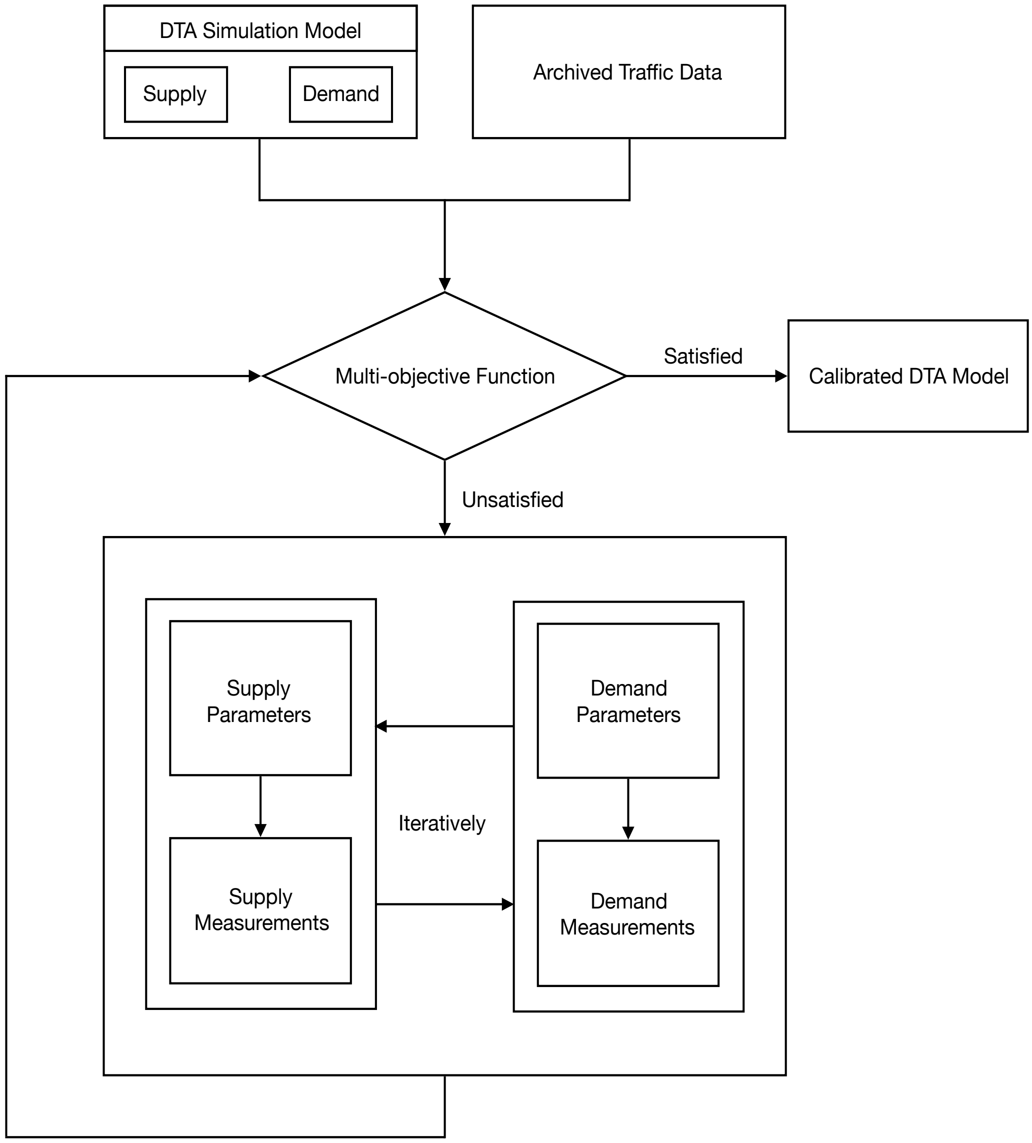

Figure 3.1 DTA Model Calibration Framework

3.2 Problem Formulation

The essence of DTA model calibration boils down to an optimization problem. The overall form of the objective function can be written as:

$$\min_{s,d} z(y_s, y_s', y_d, y_d', s, s^p, d, d^p) \tag{1}$$

s.t.

$$y_s' = f_1(s, d, N)$$
$$y_d' = f_2(s, d, N)$$

where:

y_s, y_s'	= observed and simulated traffic measurements for the supply component
y_d, y_d'	= observed and simulated traffic measurements for the demand component
s, s^p	= current and prior supply parameters
d, d^p	= current and prior demand parameters
$f_1\ (\cdot)$	= simulation model that generates measurements for the supply component
$f_2\ (\cdot)$	= simulation model that generates measurements for the demand component
N	= network

The proposed Eq. (1) is a Non-deterministic Polynomial-time Hardness (NP-hard) problem. In computability and computational complex theory, Non-deterministic Polynomial (NP) is a complexity class including the complexity class P and the complexity class NP-complete (NPC) (Van Leeuwen and Leeuwen, 1990). In the complexity class P, problems can be solved in polynomial time, while in the complexity class NPC, problems cannot be solved in polynomial time. NP-hard is a complexity class in which problems are NPC or harder. Traditional optimization techniques which rely on gradient calculation are not appropriate for solving such problem. Heuristic algorithms which generate approximate optimum solutions, are viable options to solve the problem in an efficient (in polynomial time) manner.

3.3 Optimization Algorithm

3.3.1 Genetic Algorithm

A heuristic algorithm, GA, is employed as our solution technique with its applicability and scalability as discussed earlier.

GA (Holland, 1975) is inspired by Darwin's theory of natural selection. A population of individuals (solutions) is evolved in an iterative process through three genetic operators: selection, mutation, and crossover. Population size (number of individuals in each generation) depends on the complexity of the optimization problem. The Figure 3.2 (Razali and Geraghty, 2011) presents how GA works. GA starts with a random initialization, a process that generates solutions in a predefined search space. An objective function is used to evaluate the fitness of each candidate solution. A selection operator enables selecting candidate solutions for the next generation, based on the fitness value of every solution in a population. Solutions with large fitness value have greater potential to be selected for a new generation.

As a bio-inspired algorithm, a crossover operator and a mutation operator are used for a pair of parent solutions to generate off-springs (new solutions). At the crossover stage, off-springs inherit the genes from their parents. Crossover methods include single-point crossover, multi-point crossover, and uniform crossover. Single-point crossover choices the crossover point of both parents' chromosomes randomly. Two off springs carry part of genetic information from parents' chromosomes. Under multi-point crossover, there are several crossover points during genetic information exchange. Uniform crossover does not segment the chromosomes of the parent. Each gene of parents' chromosomes treats separately to determine whether to include in the off springs.

At the mutation stage, genes are altered in a solution from their initial values, resulting in a different solution to improve diversity of genes. Mutation method include bit flip mutation, swap mutation, and inversion mutation. Bit flip mutation is a traditional method over bit strings. Each bit in a chromosome is acted independently with probability, a parameter of this operator. The common value for this parameter is $\frac{1}{l}$, where l is the length of the bit string. Swap mutation selects randomly two positions of the parents' chromosomes and interchanges the values. Inversion mutation select a section of the chromosome and inverts the section string. GA terminates when the best solution is found, or the maximum generations are reached.

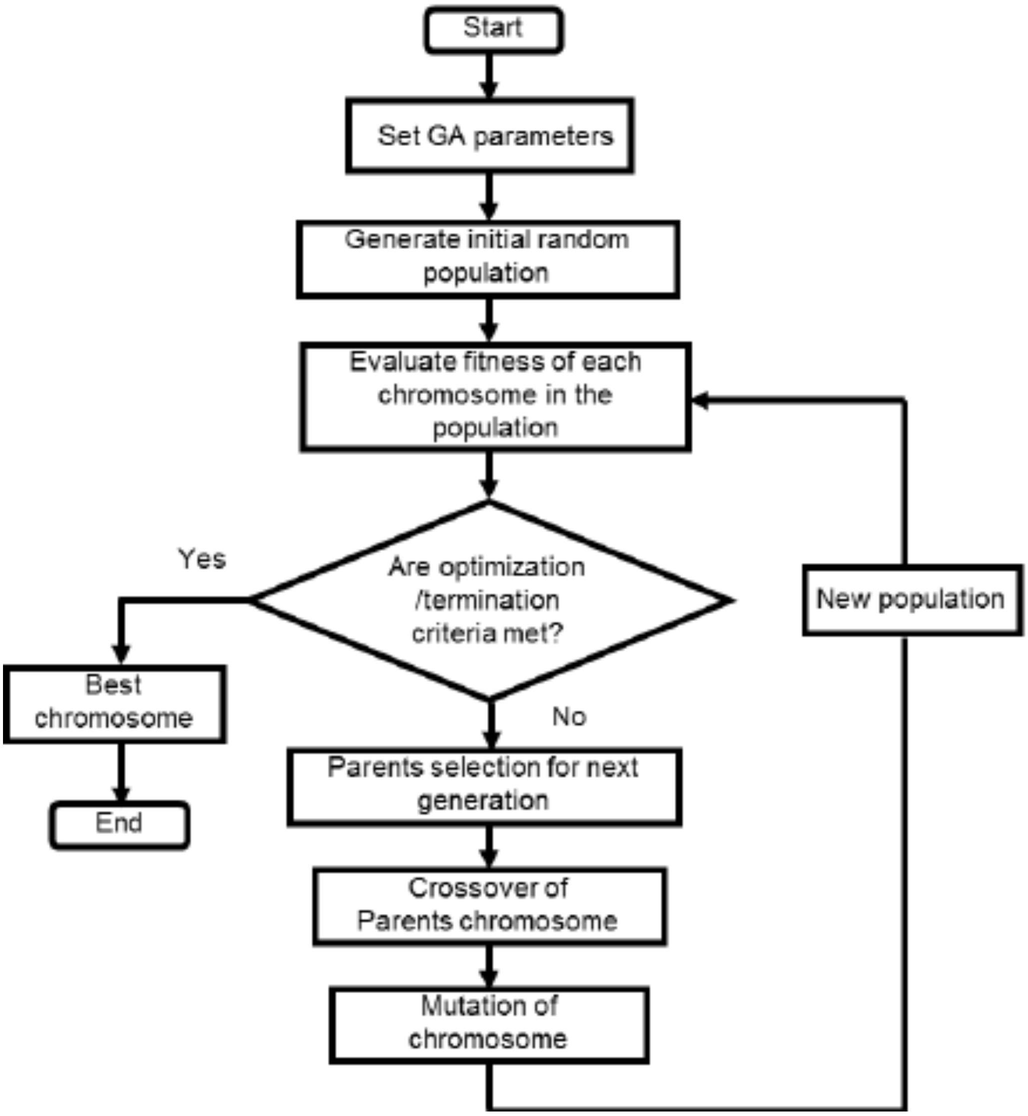

Figure 3.2 Genetic Algorithm Procedure

3.3.2 Island Genetic Algorithm

IGA is a distributed multi-population GA where individuals can migrate between islands (Whitley et al., 1999). The population in IGA is partitioned into subpopulations called islands, as presented in Figure 3.3 (Protopapadakis et al., 2012). In each island, a standard GA is executed in sequence or in parallel. In addition to three basic operators in the standard GA, a migration operator exchanges a proportion of population between islands during the evolution: migration interval determines the number of generations at which the migration occurs; migration size is the proportion of population that migrate between islands. There are various topological migration models, including fully connected models, ring-shaped models, and star-shaped models (Meng et al., 2017). Figure 3.4 shows the traditional island model. Islands in the search space connect with each other. Individuals can move between islands without constraints. Figure 3.5 presents three types of island models. A fully connected model allows individuals to migrate without trajectory constraints, while individuals can only migrate to neighbor islands in a ring-shaped model. A star-shaped model consists of a main island and subordinate islands. Individuals in a star-shaped model pass the main island in the migration.

Keeping the demand component constant, the supply calibration problem C solved by IGA is formulated as: $C(z(y_s, y_s', s, s^p | y_d, y_d', d, d^p),\ S)$, where z is the objective function to evaluate performance of each solution s and S is a search space ($s \subset S$). A Population P in IGA is divide into N subpopulations (islands), $P_1\ P_2 \dots P_n$. In generation i, each subpopulation evolves independently using three basic operators (selection, crossover, and mutation) and exchanges $\frac{P}{N} \times \gamma$ individuals using a migration operator, where γ is migration size. The termination criteria used in this research is $if\ i = i_{max}$, where i_{max} is the maximum iteration number.

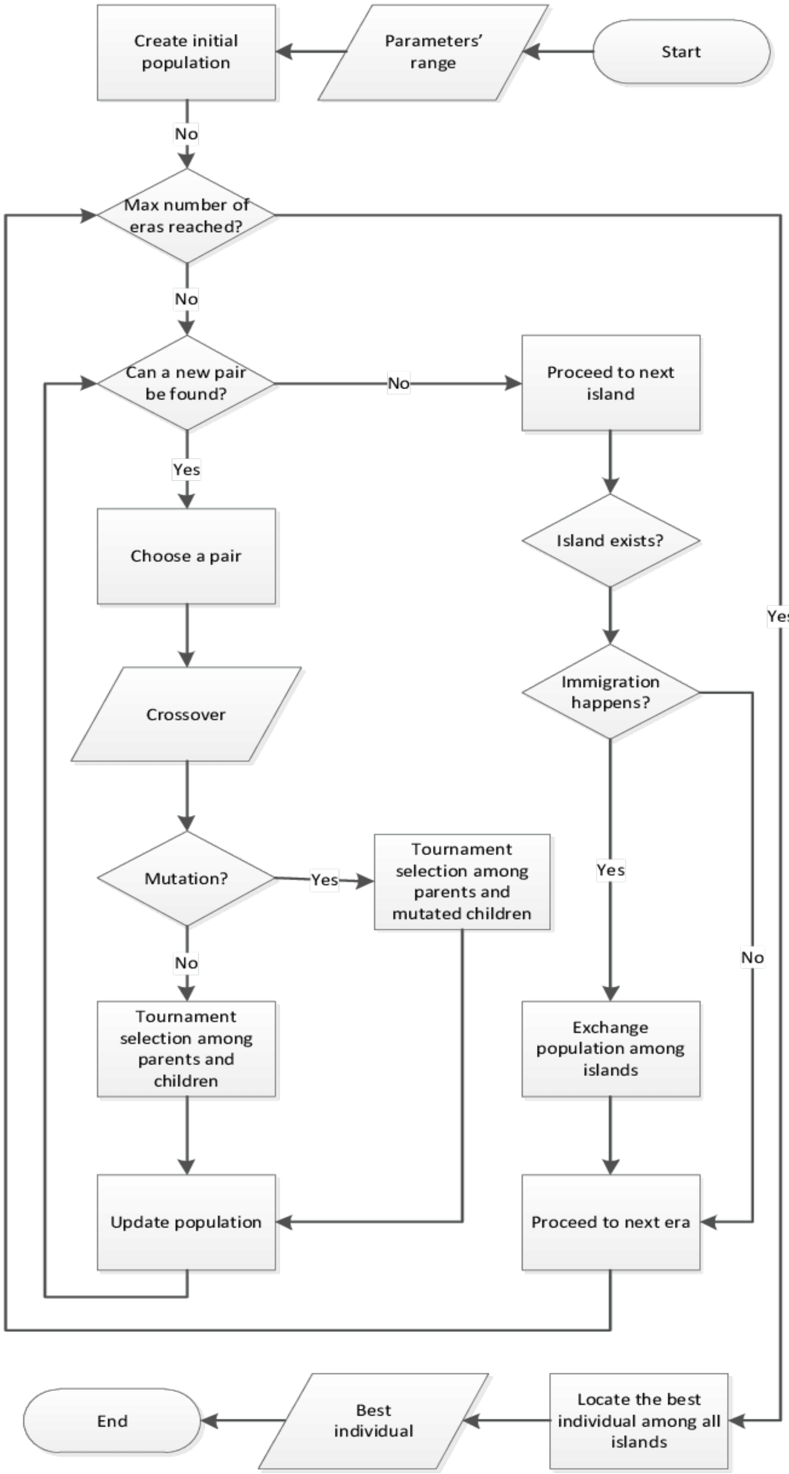

Figure 3.3 Islands Genetic Algorithm Flowchart

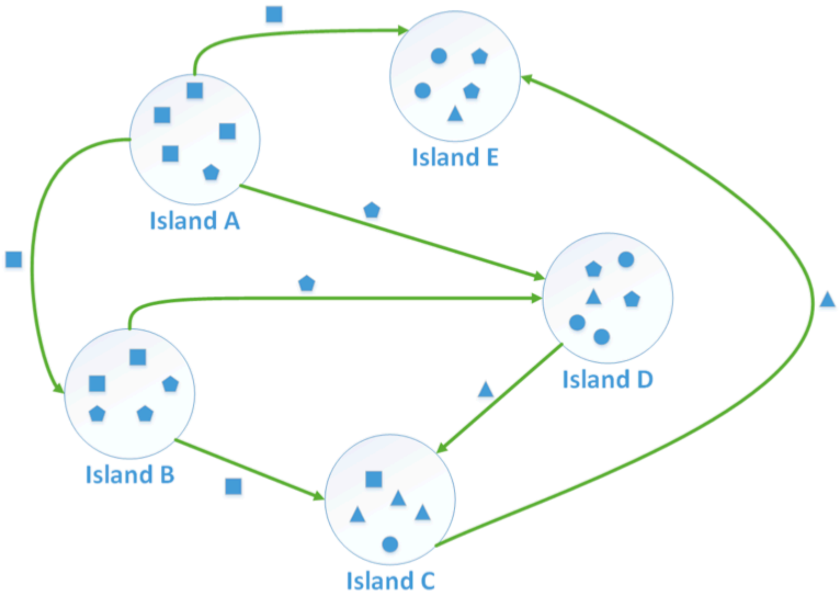

Figure 3.4 Traditional Island Model

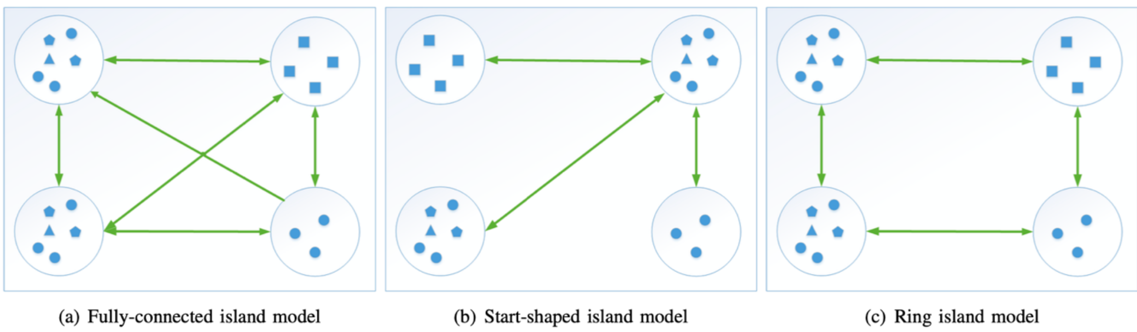

Figure 3.5 Three Types of Island Model

3.4 Simulator

The state-of-art DTA models are simulation based. We deploy Simulation of Urban Mobility (SUMO) (Lopez et al., 2018) as our DTA model simulator.

3.4.1 Introduction of SUMO

SUMO is a free, portable, and open-source software developed by German Aerospace Center (DLR) to simulate traffic network of any size. The purpose of SUMO is to simulate how individual vehicle travels through a network with a given traffic demand. Users can simulate multiple transportation modes including vehicles, public transport, and pedestrians. SUMO as a traffic simulation tool with the support of programming language Python, can be used in route assignment and emission calculation. With integrated Graphical User Interface (GUI) tool, users can visualize the simulation. SUMO provides various APIs which allow users to control the simulation remotely. Table 3.1 (Lopez et al., 2018) presents the basic components in SUMO.

Table 3.1 SUMO Components

SUMO	Simulation Command
GUISIM	Simulation with GUI
NETCONVERT	Import and convert network
NETGEN	Network Generator
OD2TRIPS	Convert OD matrices to trip file
JTRROUTER	Route generator based on turning ratios at intersections
DUAROUTER	routes generator based on a dynamic user assignment
DFROUTER	route generator with use of detector data
MAROUTER	macroscopic user assignment based on capacity functions
NETEDIT	Visualization of the network

3.4.2 Features of SUMO

The primary feature of SUMO is explicit microscopic simulation: each vehicle is modelled explicitly and travels through traffic network individually with its own route. Vehicle movement in SUMO is space-continuous and time-discrete. SUMO can simulate different vehicle types, multiple lanes, and different ROW rules. The simulation of SUMO is very fast and can work with other software during run-time, without limitations of the network size and vehicle numbers. SUMO can import network from other software such as PTV VISSIM, MATsim, and OSM. The module Traffic Control Interface (TraCI) can facilitate the online interaction for simulation control.

3.4.3 Workflow of SUMO

The workflow of SUMO in DTA model calibration is presented in Figure 3.6. There are three essential components in simulation: historical traffic data, OD demand, and urban network. The basic files required in DTA model calibration including a network file and a trip file are described below.

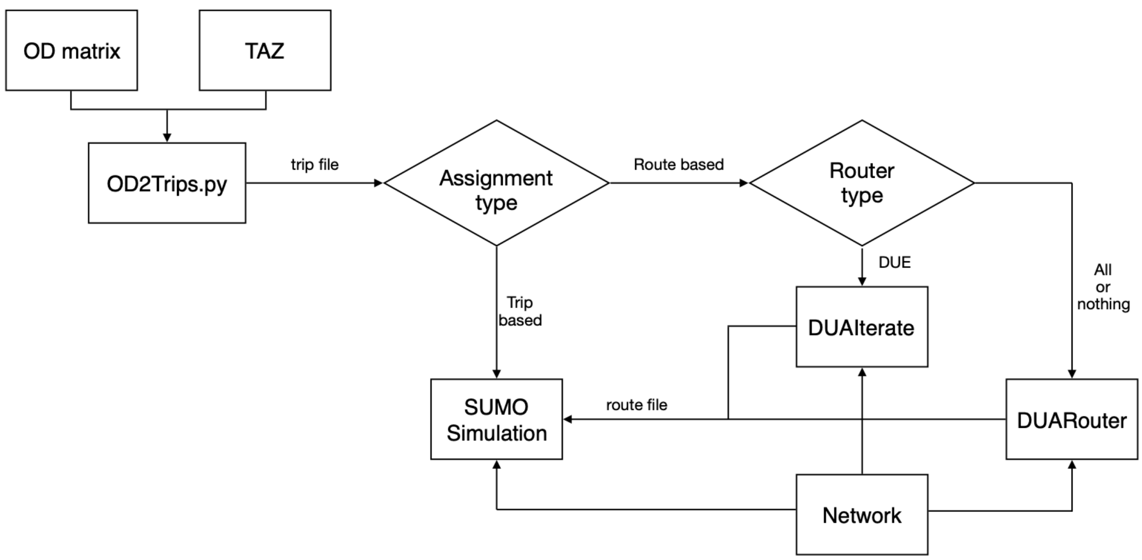

Figure 3.6 SUMO Workflow

Network files in SUMO consist of nodes and edges to represent intersections and streets. The attributes of edges include number of lanes, width, and maximum speed. The right of way (ROW) rules are included in the SUMO network to determine the driver behavior. As described in Table 3.1, we use NETCONVERT to import networks from various sources such as Open Street Map (OSM) and other simulation software (i.e., PTV VISSIM and MATsim). In case of missing traffic data in the network, NETCONVERT can refine the network in the simulation. We use NETEDIT to create and edit road network. The basic functions of NETEDIT include editing junctions and links. However, even with the help of NETCONVERT and NETEDIT, a desirable network involves intensive manual editions. The result of this step is the network file in the format of xml.

With a given network, traffic demand is essential in simulation. In SUMO, traffic demand contains information such as arrival and departure time, origin and destination, and transportation mode. We use OD matrix to describe traffic demand between traffic analysis zones, which is a required file in SUMO simulation. OD matrix can generate manually or import from other applications, but it cannot be directly used in SUMO and should be converted to trip files with the help of OD2TRIPS, a tool assigning individual trips in random or uniform distribution. Other methods in generating trips include agent-based demand model and randomTrips.py. With the given OD demand, we use DUAROUTER to facilitate the traffic assignment in the network. Traffic assignments are based on different principles, such as UE and Stochastic User Equilibrium (SUE), Dynamic User Equilibrium (DUE), and All-or-Nothing. The output of traffic assignment in SUMO is the route file. In summary, the results of this step are the trip file and the route file in format of xml.

Chapter 4 Case Study

This chapter presents two case studies to demonstrate the proposed methodology. Specifically, our case studies mainly focus on supply calibration, with assumption of given demand. The first case study uses a synthetic network and the second case study uses a real-world network from Munich, Germany. The overview of the network properties is given, and the calibration process is described. We discuss the corresponding calibration results to compare the performance of IGA and the standard GA.

4.1 Synthetic Case Study

The objective of the synthetic network is to validate the proposed methodology and to compare the calibration results of IGA and standard GA in terms of convergence speed and solution quality. After validating the robustness of IGA in DTA model calibration using this synthetic network, we extend network scale to the real-world case.

4.1.1 Network Description

The synthetic network was created by using NETEDIT in SUMO, is shown in Figure 4.1. The network has a uniform pattern and a simply graph, consisting of 16 nodes and 48 links. To simplify the traffic analysis, we regard each individual node as one traffic analysis zone (TAZ), with 256 TAZ in total.

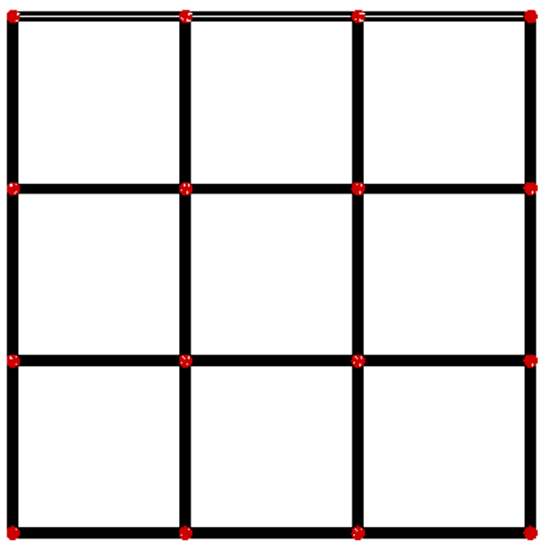

Figure 4.1 Synthetic Network

4.1.2 Calibration Parameter

In our synthetic case study, we calibrate the car-following model with parameters of acceleration, deceleration, and driver imperfection. Krauß (1998) described various existing car-following models and developed Krauss car-following model. In traffic flow theory, there are three traffic states: free flow, synchronized flow, and jammed flow. The transition phase of different traffic states can be described as first order transition phase. The car following models capture the properties of phase transition first. The author classified different car-following models based on acceleration and deceleration characteristics into three categories. The macroscopic Kerner-Konhaeuser-model and microscopic Bando-model were in category I with features including realistic acceleration and deceleration, correctly modeling of jams, and jamming transition is the first order phase transition. The Nagel-Schreckenberg model with features of unbounded deceleration and no phase transition in jamming phase was in category II. The Fukui-Ishibashi model with features of unbounded accelerations and no jams was in category III. The Krauss model considers the phenomenon of synchronized flow with assumptions that drivers can give up speed in case of highly jammed traffic state. Therefore, the Krauss model considers effects of limited acceleration and deceleration of vehicles. We use the Krauss model in SUMO to calibrate the synthetic network, aiming to reproduce travel times between OD pairs.

4.1.3 Calibration Process

We use a genetic algorithm function and an islands genetic algorithm function from the Package ‘GA’ in R (Scrucca, 2013). The calibration procedure for the supply component consists of four steps:

Step 1: Set configuration for IGA and GA

Table 4.1 and 4.2 present configuration for IGA and GA in the supply calibration procedure respectively. For comparing the performance of IGA and GA in the case study, basic parameters such as population size, are same for both algorithms. IGA requires additional settings for a migration operator with parameters of migration rate and interval. In setting configuration for IGA and GA, there are no clear guidelines for an optimal combination. The rule of thumb is that the best and mean fitness value of IGA and GA should converge before termination.

Table 4.1 IGA Configuration for Synthetic Network

Number of islands	3
Chromosome	Acceleration, deceleration, and driver imperfection
Population size	60
Selection	Fitness proportional selection with fitness linear scaling
Crossover	Local arithmetic crossover
Mutation	Uniform random mutation
Crossover probability	0.8
Mutation probability	0.1
Migration rate	0.1
Migration interval	5
Maximum iteration	60

Table 4.2 GA Configuration for Synthetic Network

Chromosome	Acceleration, deceleration, and driver mmperfection
Population size	60
Selection	Fitness proportional selection with fitness linear scaling
Crossover	Local arithmetic crossover
Mutation	Uniform random mutation
Crossover probability	0.8
Mutation probability	0.1
Maximum iteration	60

Step 2: Execute the microscopic simulation model in SUMO

In the synthetic case study, we require a network file, a trip file, and an additional file for running microscopic simulations. The additional file encoded in XML format contains the calibrated parameters including acceleration, deceleration, and driver imperfection that are generated by IGA and GA. The trip file is converted from OD matrix that are calibrated in the demand component. The network file that describes network properties keeps constant in this case study.

Step 3: Evaluate model output

After running simulations, SUMO generates output files with OD travel time. In this case study, we use Mean Squared Error (MSE) to assess the fitness value of each solution generated by IGA and GA. The formulation of MSE is:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_s' - y_s)^2 \tag{2}$$

where y_s' and y_s are simulated and observed travel time, respectively.

Step 4: Terminate calibration process

The IGA and GA terminate when reaching the maximum iteration number (e.g., 60 generations).

4.1.4 Calibration Result

Figure 4.2-4.6 present the calibration results of IGA and the standard GA as indicated by MSE, the fitness value. Figure 4.2 shows the fitness value of best solution (during 60 generations) using IGA with three islands. In the process of evolution, three subpopulations converge in the generation of 42, with the fitness value of 13.93. When comparing the evolution speed of individual island, three islands find the best solution approximately at the same time with the island 1 slightly faster. Figure 4.3-4.5 show the convergence of the best and mean fitness value of each island. All three islands converge their average and best fitness value before termination.

Figure 4.6 shows the performance of standard GA with a single population using same configurations of IGA. The standard GA finds its best solution after 46 iterations which is 4 generation slower than the IGA. Overall, IGA outperforms the standard GA in terms of convergence speed.

In our synthetic case study, however, the solution quality and the convergence speed of IGA and of the standard GA do not show large difference. The reason is probably due to the simplicity of the synthetic network. We explore the performance of IGA and the standard GA in the real-world network (the second case study) further.

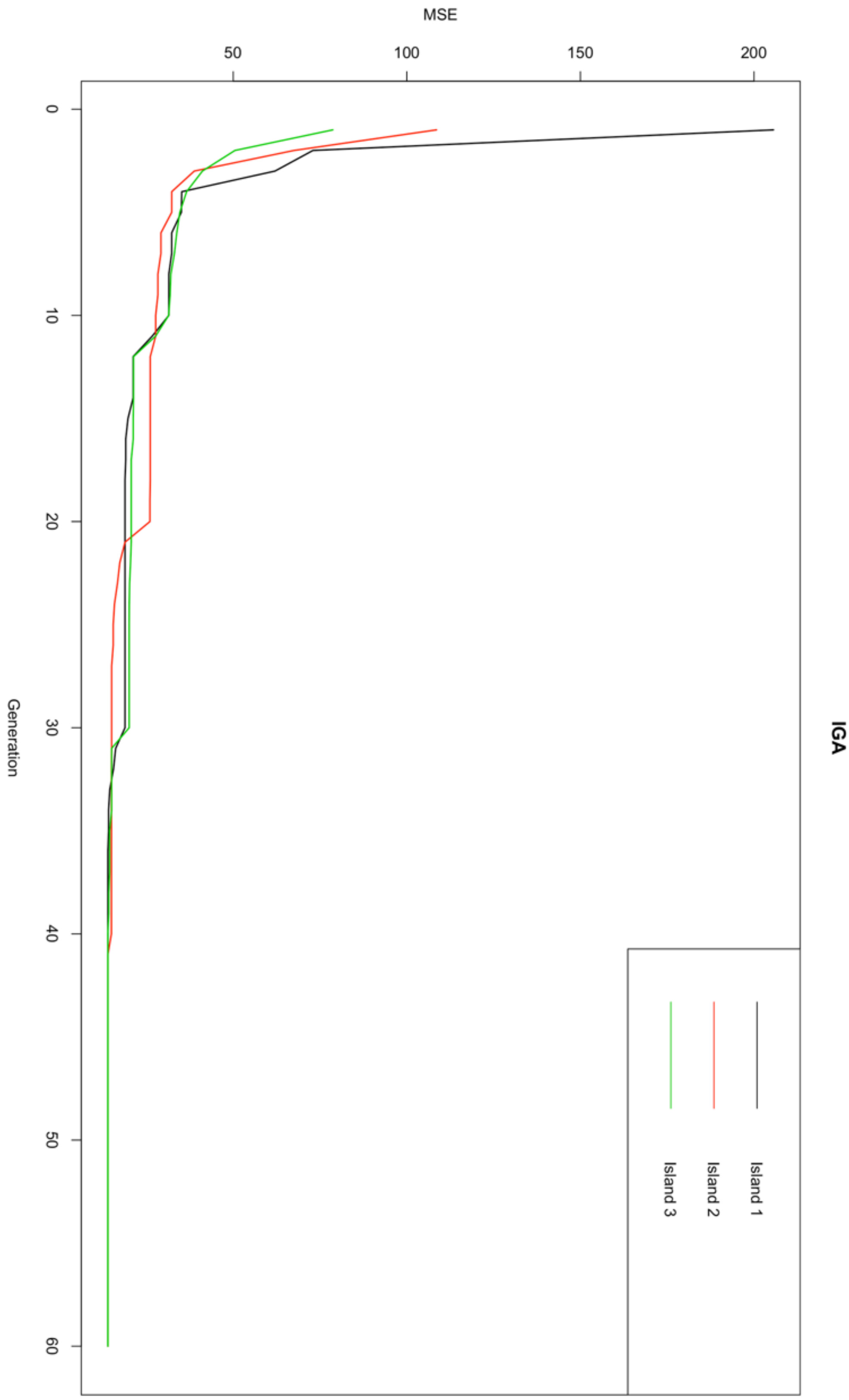

Figure 4.2 The Synthetic Case Study: Performance of IGA

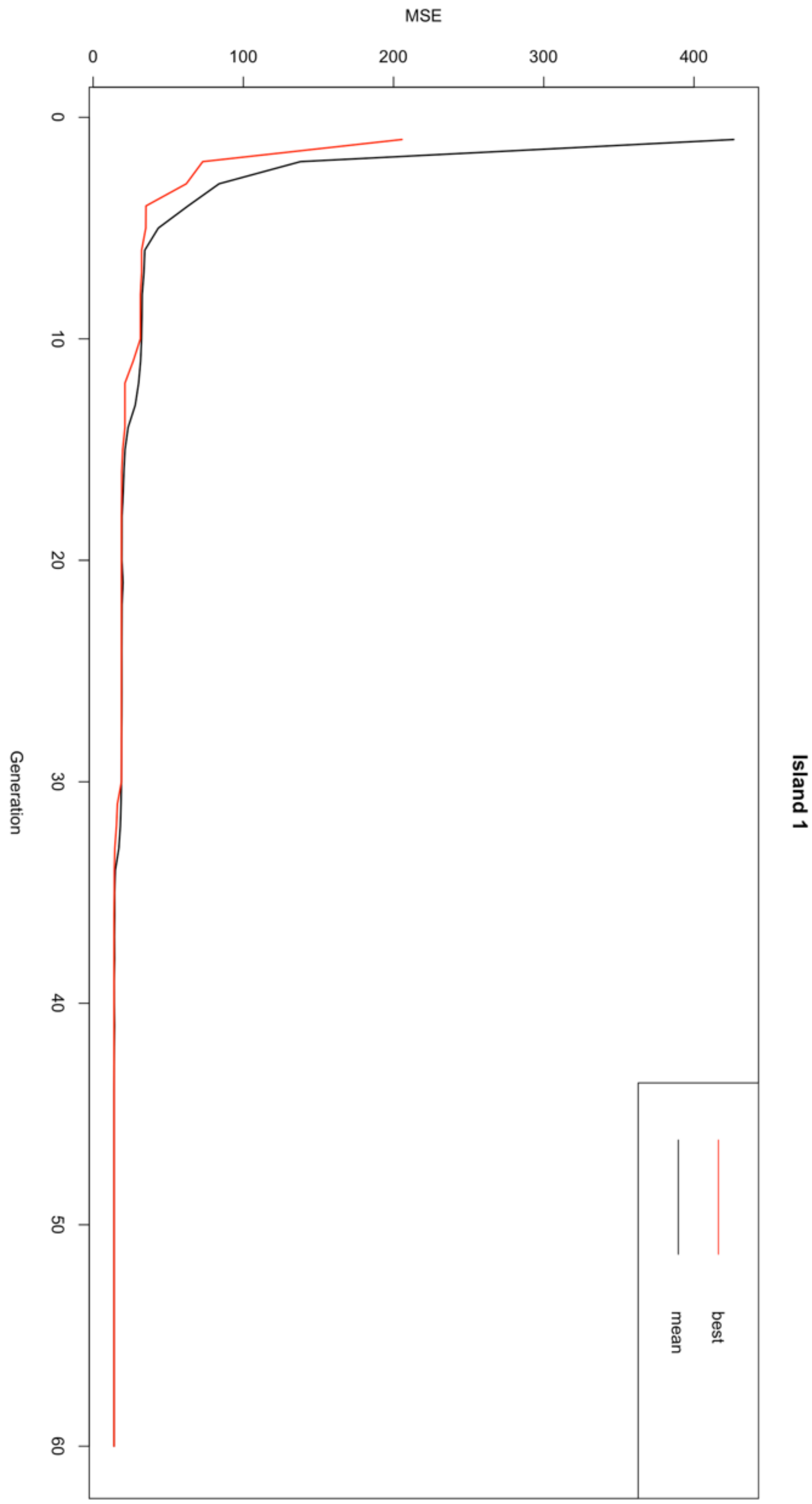

Figure 4.3 The Synthetic Case Study: Performance of Island 1

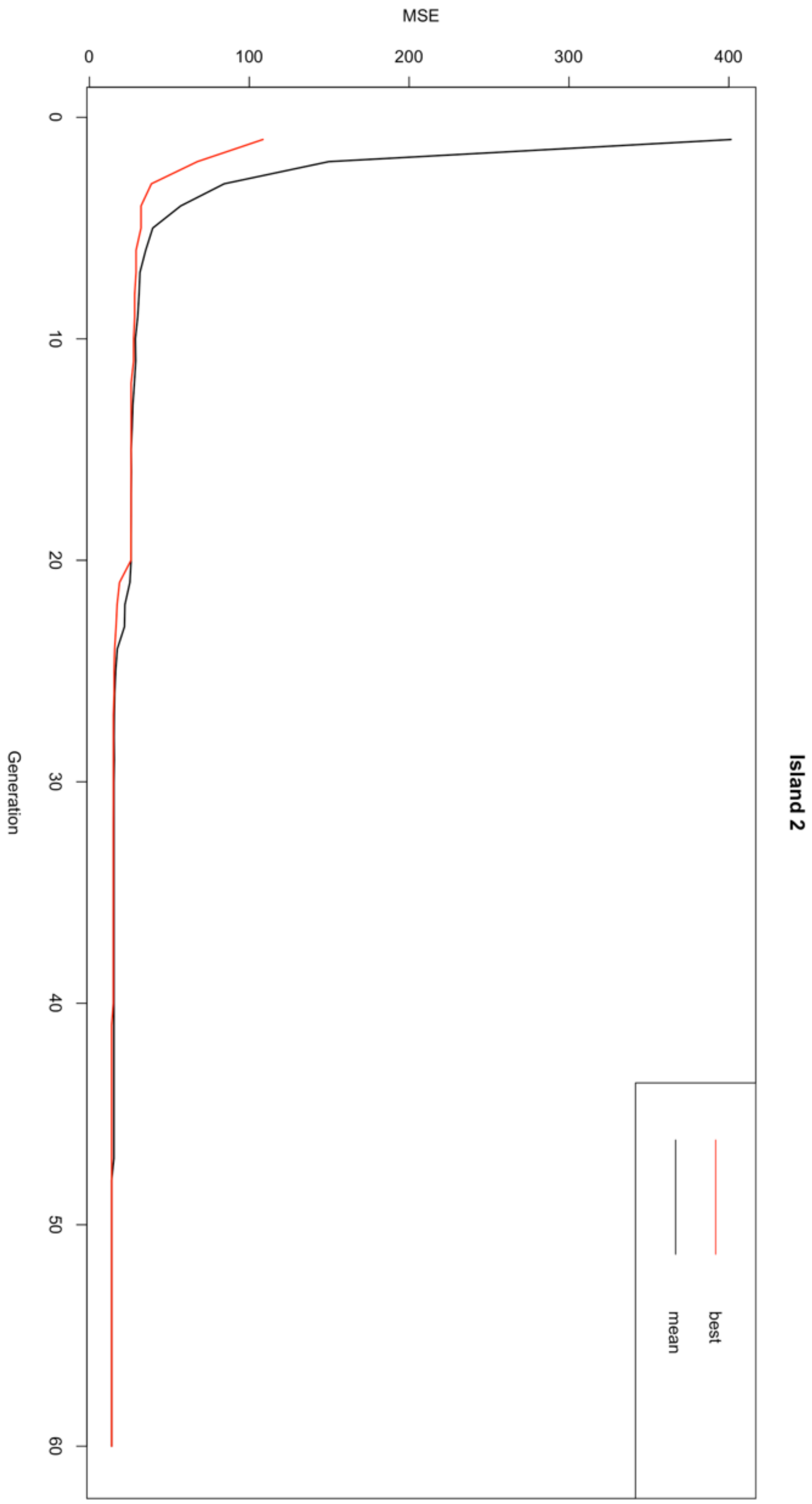

Figure 4.4 The Synthetic Case Study: Performance of Island 2

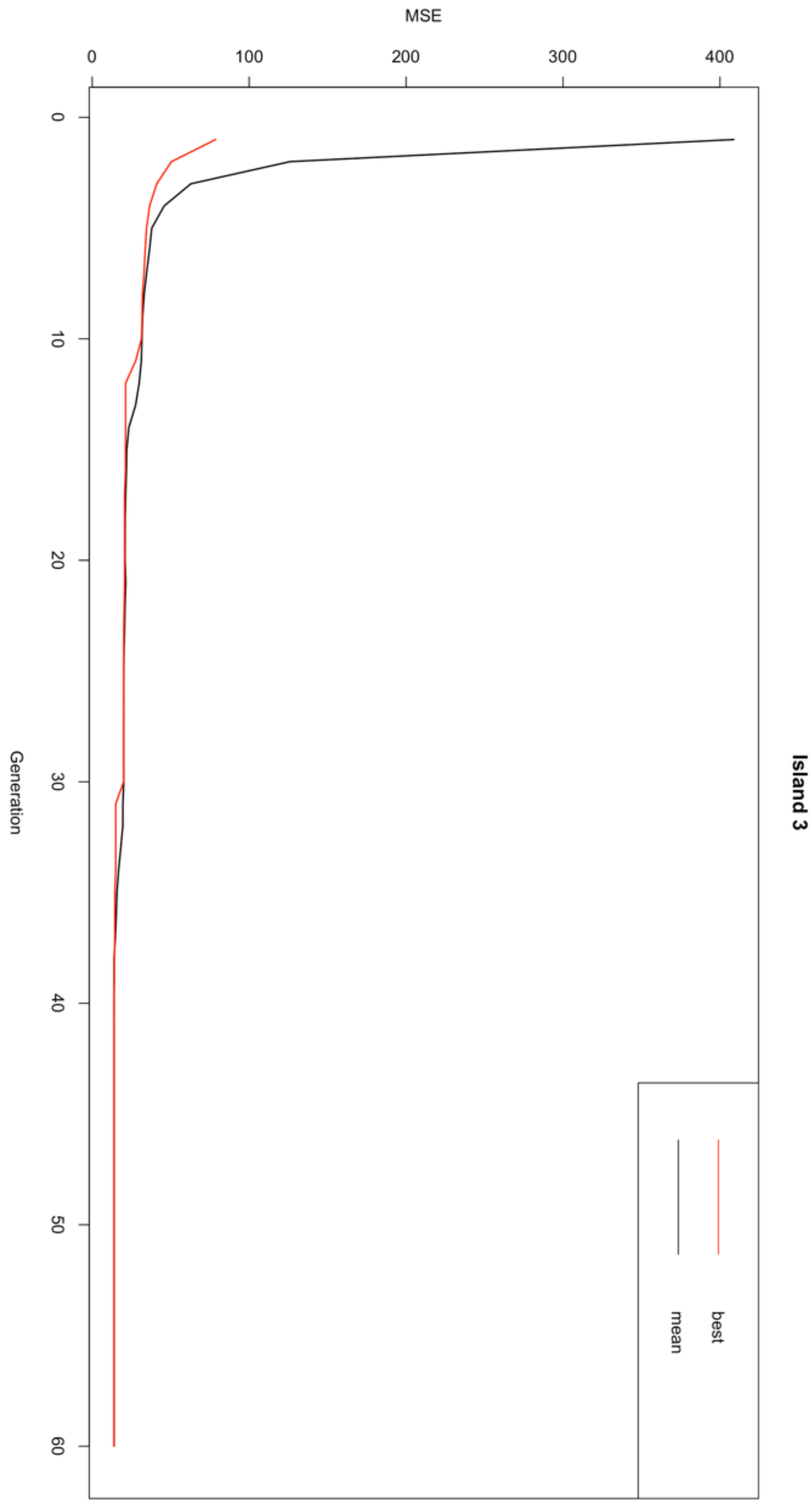

Figure 4.5 The Synthetic Case Study: Performance of Island 3

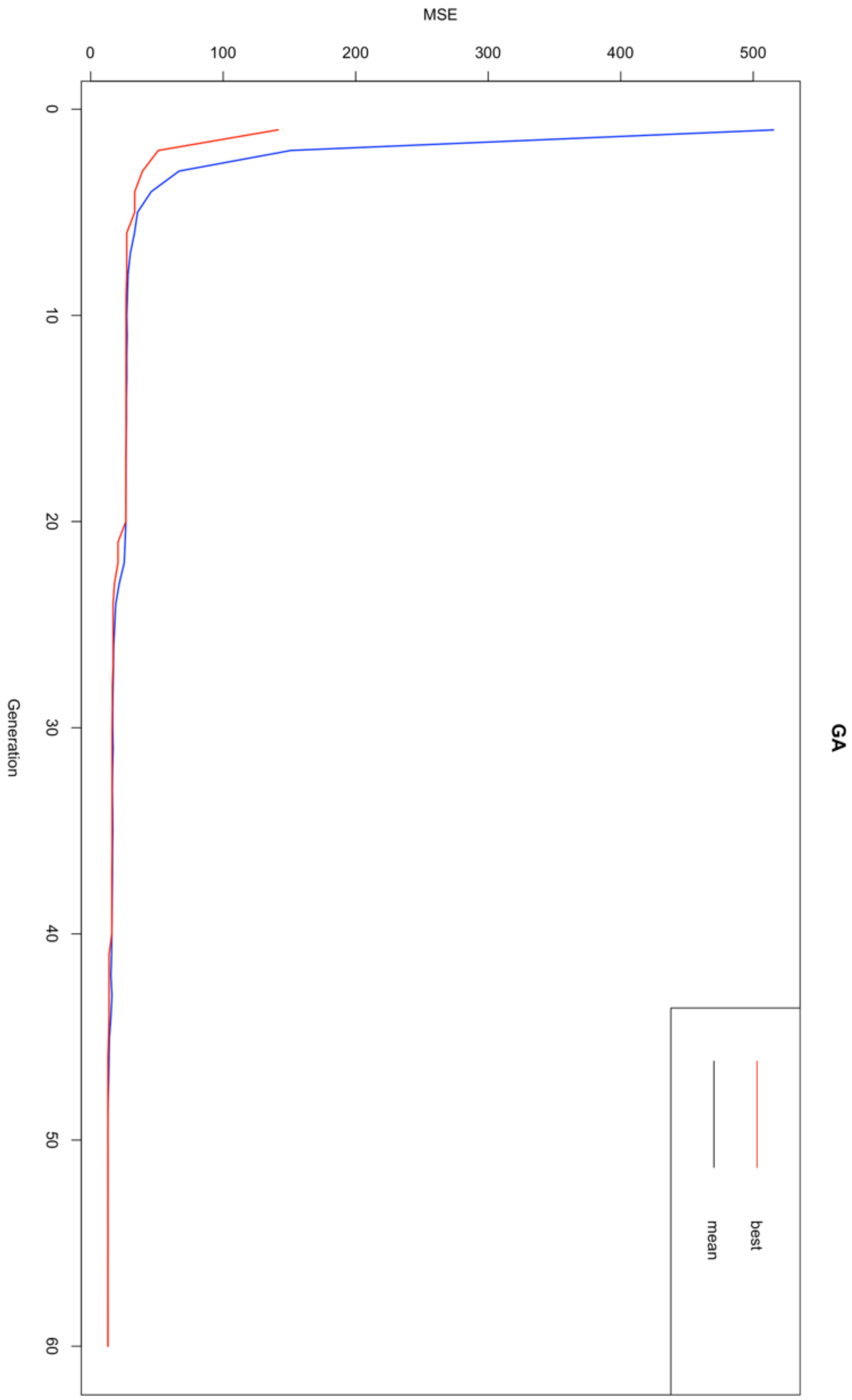

Figure 4.6 The Synthetic Case Study: Performance of Standard GA

4.2 Munich Network Case Study

After calibrating the synthetic network, we conduct our second case study on a network of Munich, Germany. We give the description of the network feature and illustrate the calibration parameters and procedure. Afterwards, we discuss about the calibration results.

4.2.1 Network Description

A metropolitan urban network of Munich consists of a dense city center, surrounding arterials, and a highway bypass, as shown in Figure 4.7. There are 2408 links about 946 km in length and 1475 nodes in the network model. For the traffic analysis purpose, the network is divided into 61 traffic analysis zones, generating an OD matrix with 3721 OD pairs.

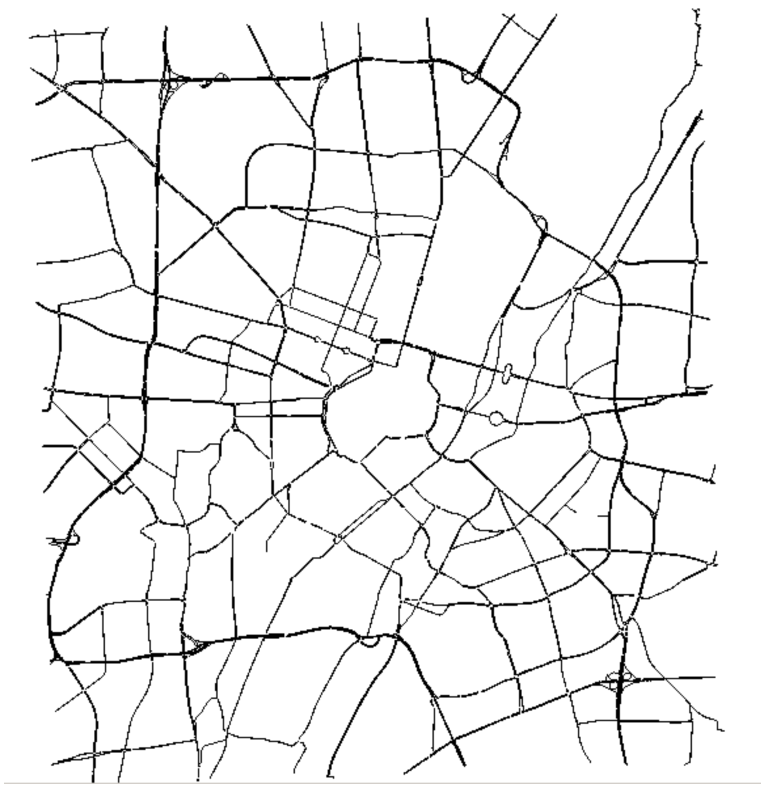

Figure 4.7 Munich Network

4.2.2 Calibration Parameter

In this case study, we calibrate link speeds as the supply parameters and keep the demand parameters (OD flows) as fixed. The proposed methodology is to iteratively calibrate the supply and demand component with a multi-objective function, i.e., travel time for measuring the supply component and traffic counts for measuring the demand component. However, we focus mainly on the supply calibration with traffic measurement of travel time. Additionally, due to unavailability of real-world data, we build the experimental data synthetically, as suggested by Antoniou et al. (2016)

4.2.3 Calibration Process

Step 1: Set configuration for IGA and GA

Table 4.3 and 4.4 present configuration for IGA and GA in the supply calibration procedure respectively. For comparing the performance of IGA and GA in the case study, basic parameters (e.g., population size) and basic operators (e.g., selection, mutation, and crossover), are same for both algorithms. IGA requires additional settings for a migration operator with parameters of migration rate and interval. Considering the increasing network size compared with the synthetic network, we increase the population size to 200 for both IGA and GA. We introduce four islands in this case study which means that each island contains 40 chromosomes in each generation.

In setting configuration for IGA and GA, there are no clear guidelines for an optimal combination. The purpose of our case study is to compare the performance of IGA and the standard GA in DTA model calibration. Therefore, we use the same settings for both algorithms. The rule of thumb for GA is that the best and mean value of GA should converge before termination of the evolution process.

Table 4.3 IGA Configuration for Munich Network

Number of islands	4
Chromosome	Link speeds
Population size	200
Selection	Fitness proportional selection with fitness linear scaling
Crossover	Local arithmetic crossover
Mutation	Uniform random mutation
Crossover probability	0.8
Mutation probability	0.1
Migration rate	0.1
Migration interval	5
Maximum iteration	100

Table 4.4 GA Configuration

Chromosome	Link speeds
Population size	200
Selection	Fitness proportional selection with fitness linear scaling
Crossover	Local arithmetic crossover
Mutation	Uniform random mutation
Crossover probability	0.8
Mutation probability	0.1
Maximum iteration	100

Step 2: Execute the mesoscopic simulation model in SUMO

In this case study, SUMO requires a network file and a trip file for running mesoscopic simulations. A Munich network file encoded in XML format contains the calibrated link speeds that are generated by IGA and GA. A trip file is converted from OD matrix that are calibrated in the demand component. The supply calibration is to find a set of speeds by IGA and GA for the network, while the trip OD matrix remains as constant.

Step 3: Evaluate model output

After running simulations, SUMO generates output files with travel time. RMSE is used to assess the fitness value of each solution generated by IGA and GA. The formulation of RMSE is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_s' - y_s)^2} \qquad (3)$$

where y_s' and y_s are simulated and observed travel time, respectively.

Step 4: Terminate calibration process

We use the maximum iteration number (e.g., 100 generations) as the termination criterion for both IGA and GA

4.2.4 Calibration Result

The results of the supply component calibration with Munich network are presented in Figure 4.8-4.13. Figure 4.8 illustrates the calibration results using IGA with 4 islands after 100 generation. In the process of evolution, four subpopulations converge in the generation of 62, with the fitness value of 13.26. The subpopulation 2 evolves the fastest, converging in the generation of 41, while subpopulation 1 converges the slowest, converging after 62 iterations. Figure 4.9-4.12 show the convergence of the best and mean fitness value of each island. All four islands converge their average and best fitness value before termination.

The performance of standard GA with a single population is shown in Figure 4.13. The standard GA gives its best solution after 71 iterations. The mean and best fitness value converge in the generation of 79. When comparing the performance of IGA and standard GA in terms of convergence speed, IGA outperforms the standard GA which is 12.6% slower than IGA. In addition, the solution quality of IGA is slightly better than standard GA, with fitness values of 13.26 and 14.67, respectively. Therefore, IGA improves efficiency and accuracy in DTA supply model calibration.

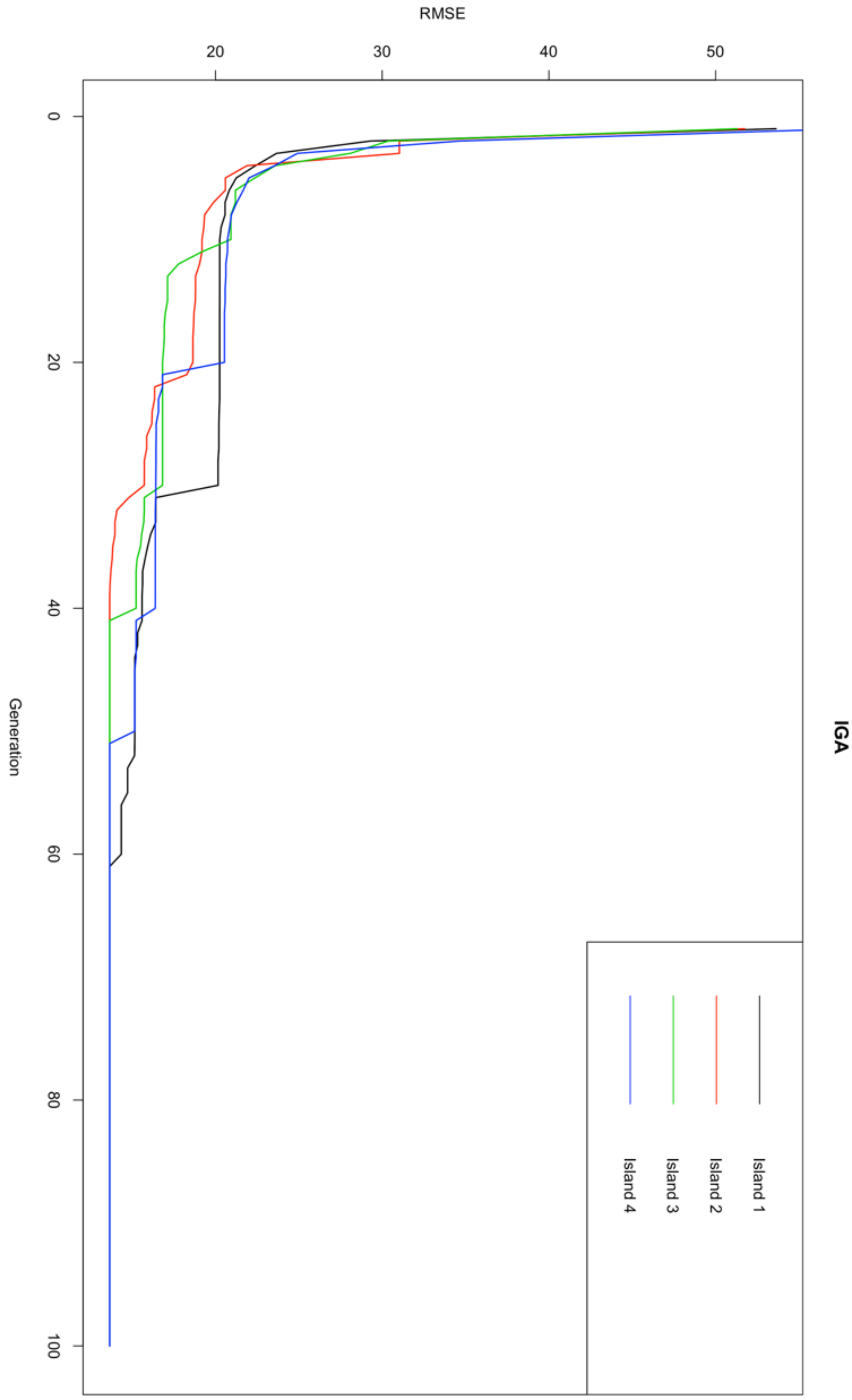

Figure 4.8 The Munich Case Study: Performance of IGA

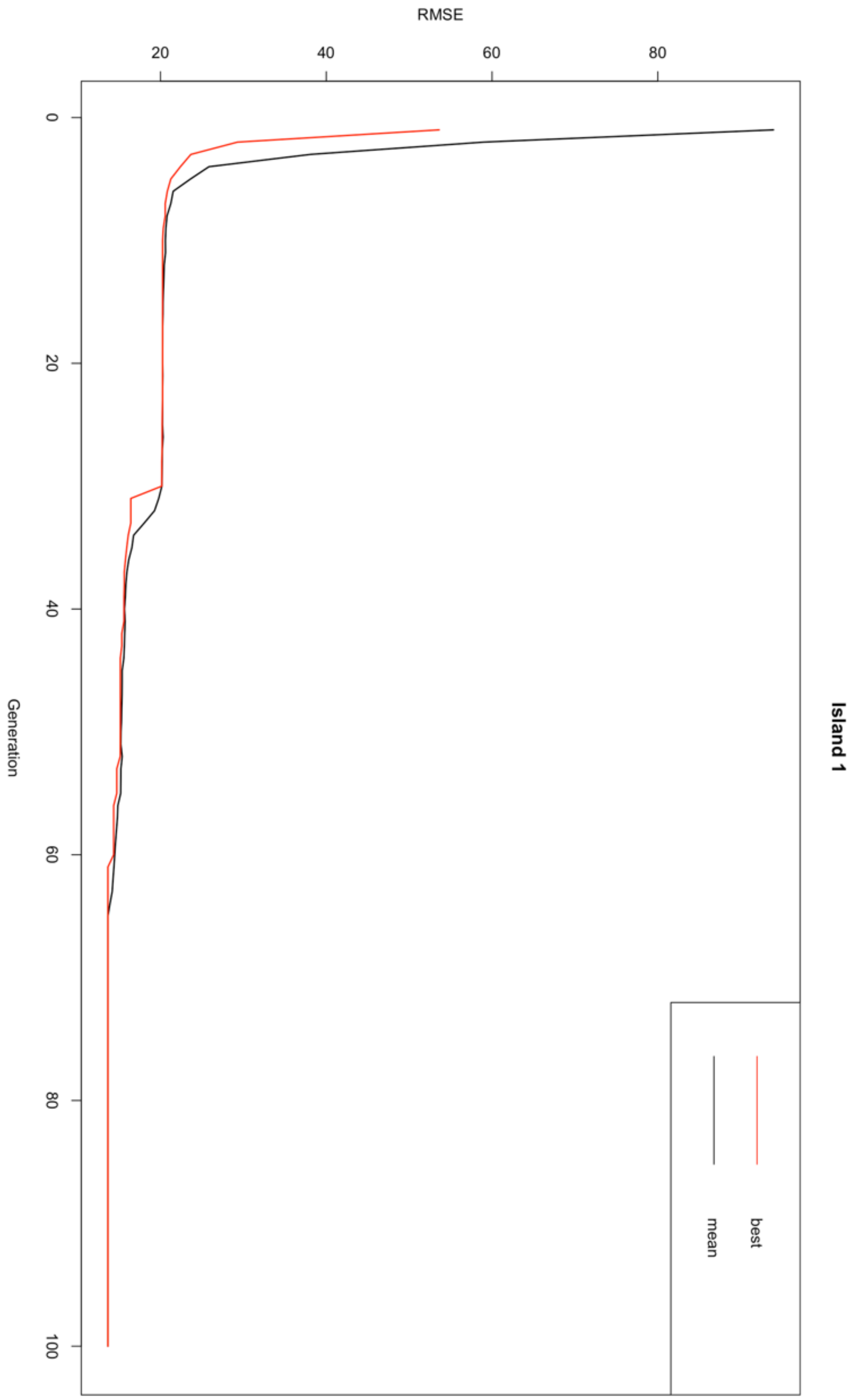

Figure 4.9 The Munich Case Study: Performance of Island 1

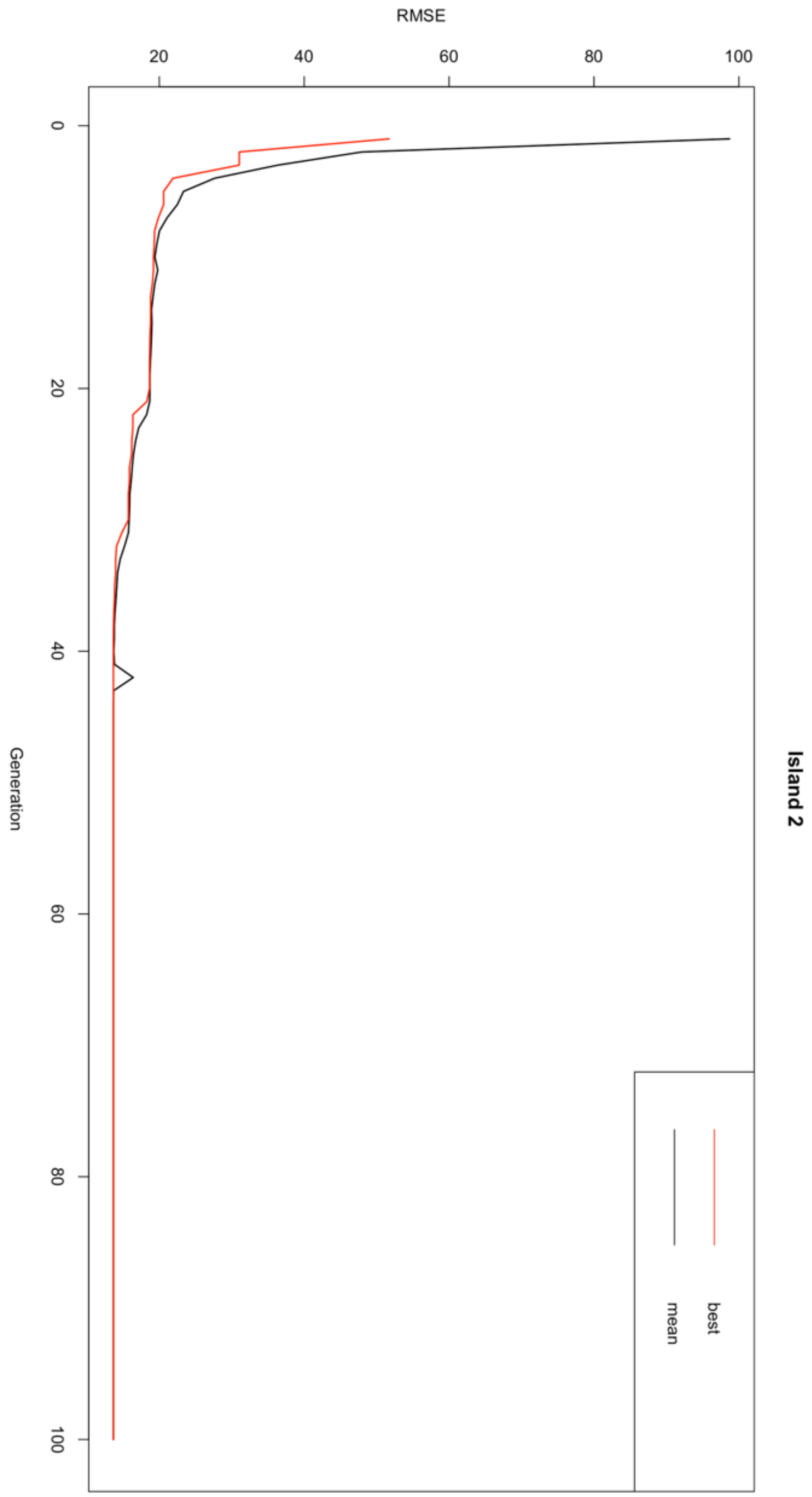

Figure 4.10 The Munich Case Study: Performance of Island 2

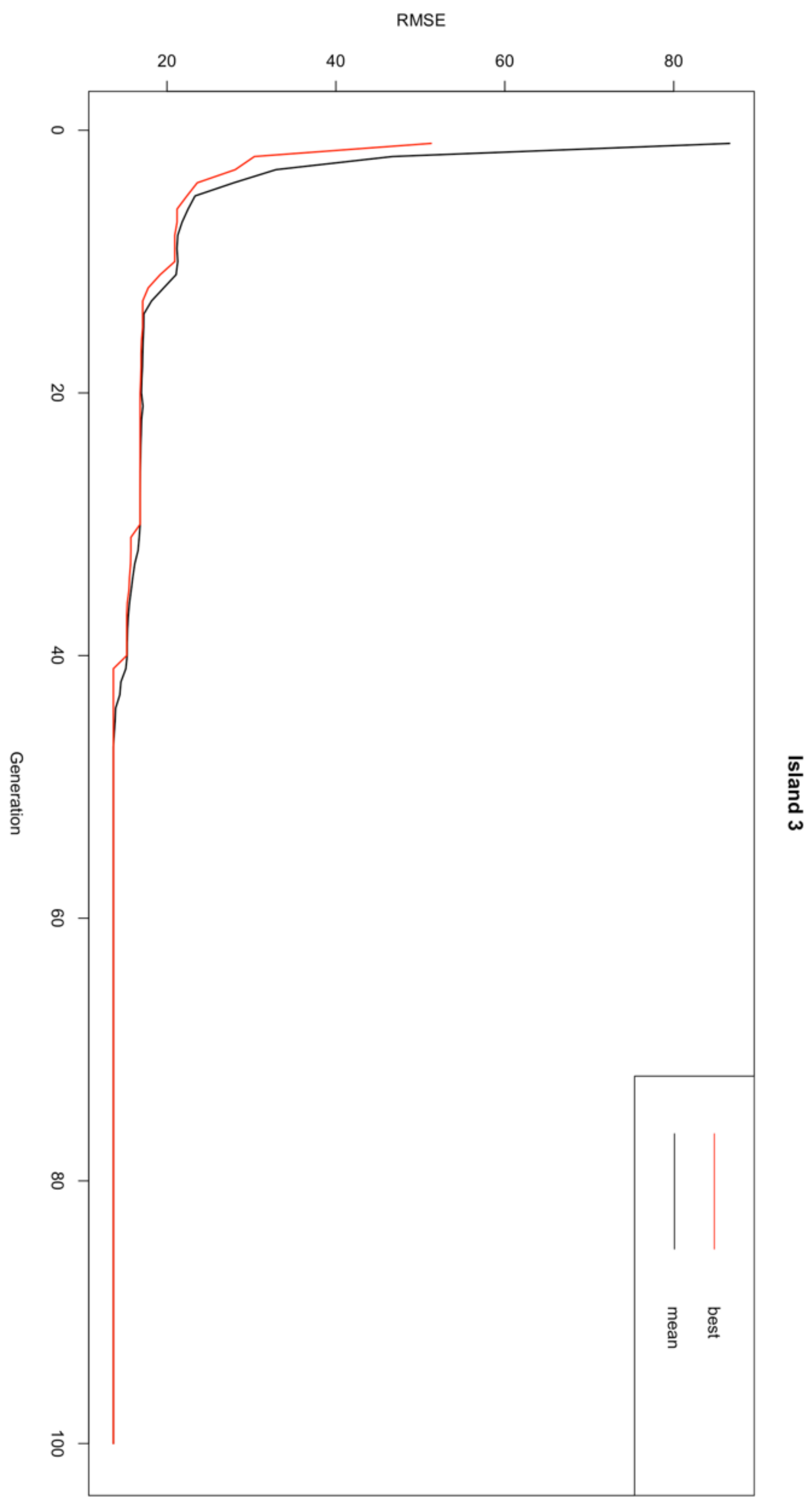

Figure 4.11 The Munich Case Study: Performance of Island 3

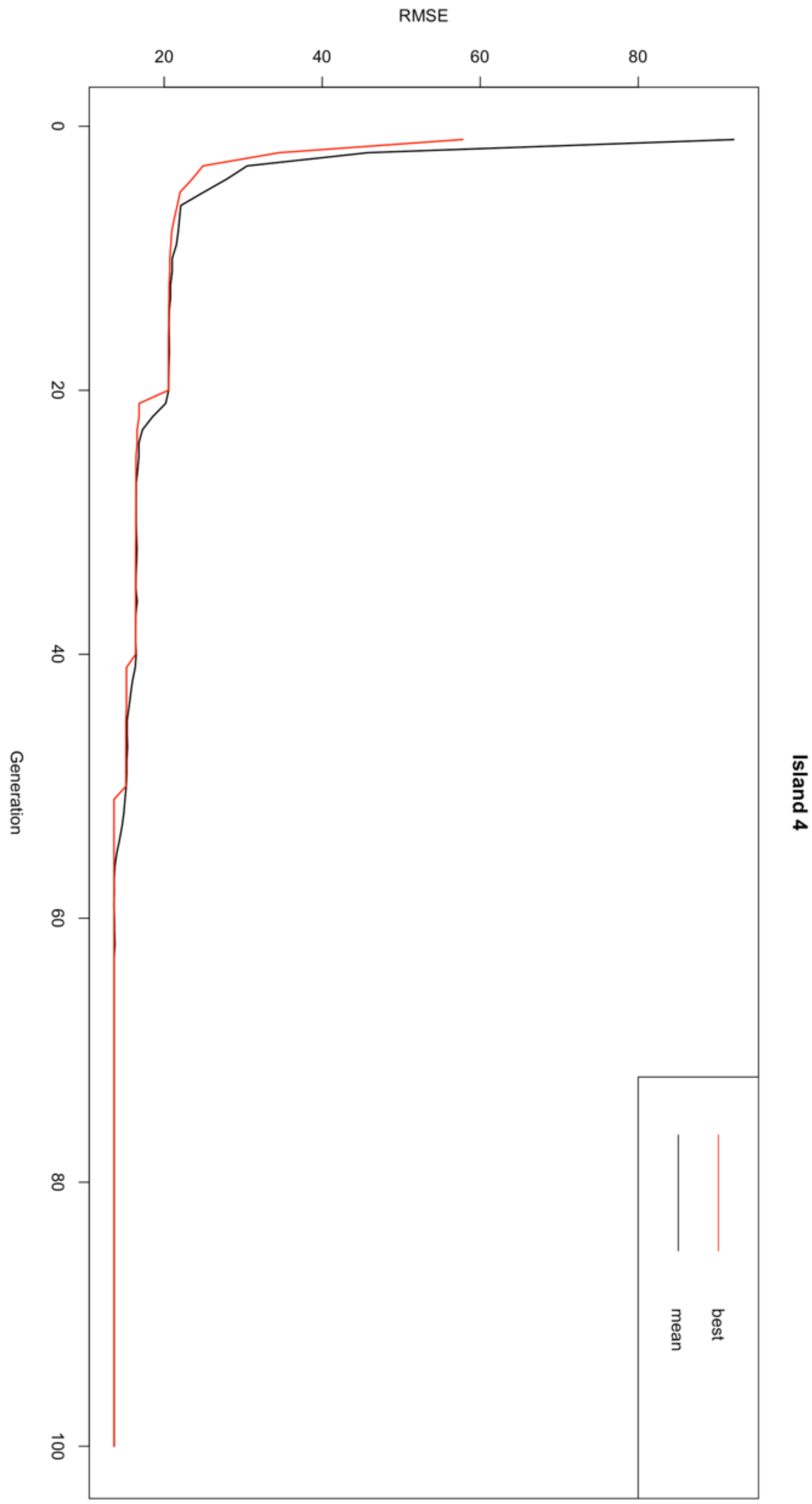

Figure 4.12 The Munich Case Study: Performance of Island 4

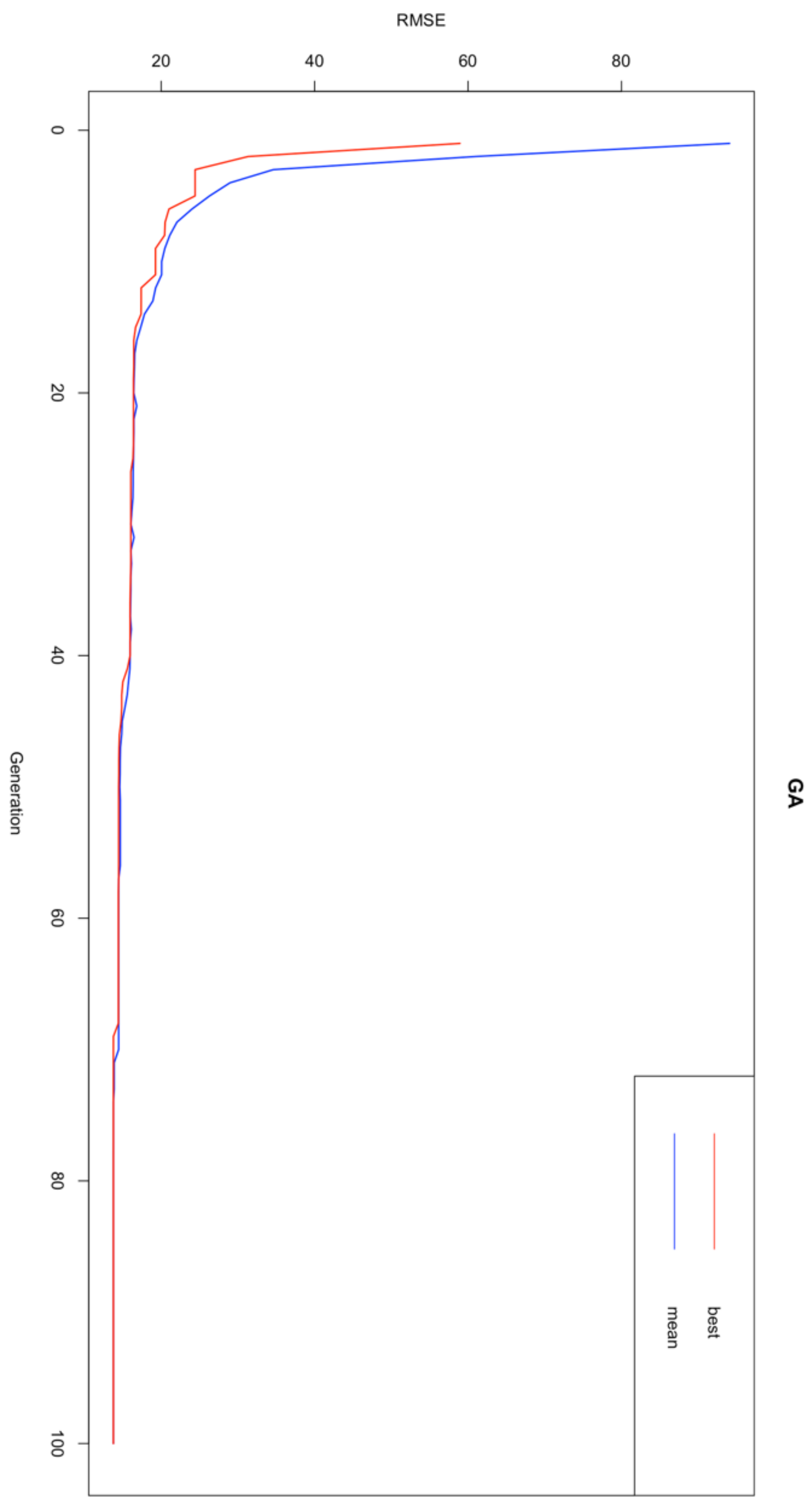

Figure 4.13 The Munich Case Study: Performance of Standard GA

Chapter 5 Blockchain

In real time calibration of DTA model in the future, we will require traffic data from various mobility modes. The Blockchain technology provides a new way in traffic data management. First, we introduce three new transportation modes in this chapter: Urban Air Mobility (UAM), Autonomous Vehicle (AV), and Shared Mobility. Afterwards, we give the overview of Blockchain technology and its application to facilitate online calibration of DTA model.

5.1 New Mobility Mode

New transportation modes will reshape the way of travelling in the future. We introduce three mobility modes: Urban Air Mobility, Autonomous Vehicle, and Shared Mobility. We give the overview of these new mobility modes regarding their features and market potentials. Afterwards, we focus on the modelling efforts on these new mobility concepts.

5.1.1 Urban Air Mobility

The increasing urbanization and congestion problems bring the emergence of Urban Air Mobility (UAM), a new solution for urban travel. The UAM serves as a flying taxi in the urban area, easing the congested urban network by adding a third dimension. The service provided by UAM is appealing because it saves a lot of time for commuters, compared with conventional transportation mode such as bus and train. UAM has potential to become the fastest transport option among others by getting rid of congestion. The on-demand service in UAM reduces waiting time and ensure a pleasant travel experience. UAM has large market potential, as predicted by Roland Berger (2018) that the estimated number of passenger drones worldwide would be 98,000 in 2050 due to the implementation of electric propulsion, autonomous flight technology, and 5G technology.

Modelling of UAM has received much attention. Rothfeld et al. (2018) presented a methodology for the multi-agent transport simulation, with two examples (as shown in Figure 5.1) of UAM network type used in their research. The two scenarios were built on Sinoux Falls under MATsim. The first example follows decentralized network structure in which UAM service is point to point. The network structure of the second scenario has a central station and UAM flights through the central station every time.

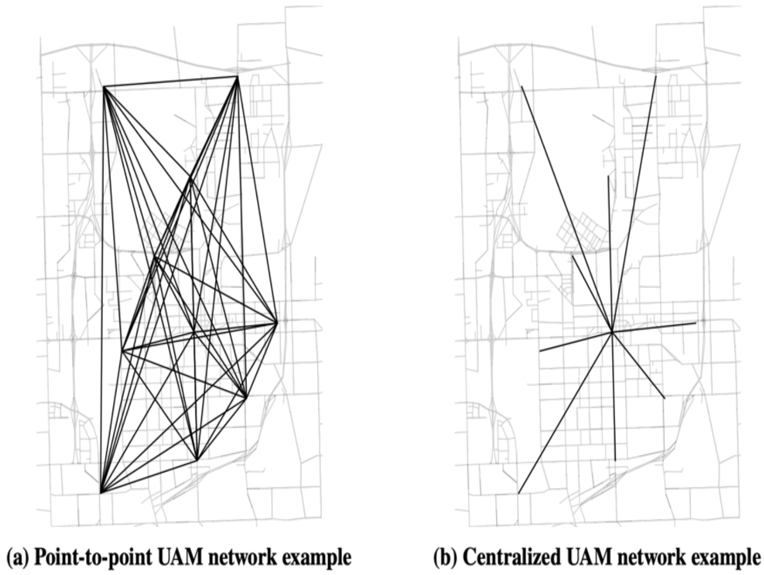

Figure 5.1 Two Scenarios of UAM Simulation

Roland Berger (2018) provided a case study of UAM in Munich, Germany, as shown in Figure 5.2. The point of interests (POI) of their research include key traffic nodes such as airports and city centers in the Bavarian metropolitan area. Potential landing sites for UAM are identified depending on the population density and available infrastructure in four cities (Munich, Ingolstadt, Augsburg, and Rosenheim). The research findings indicated that there would be 100 drones in 2030 and 800 drones in 2050 and the switching rate would be 5% an average in 2050.

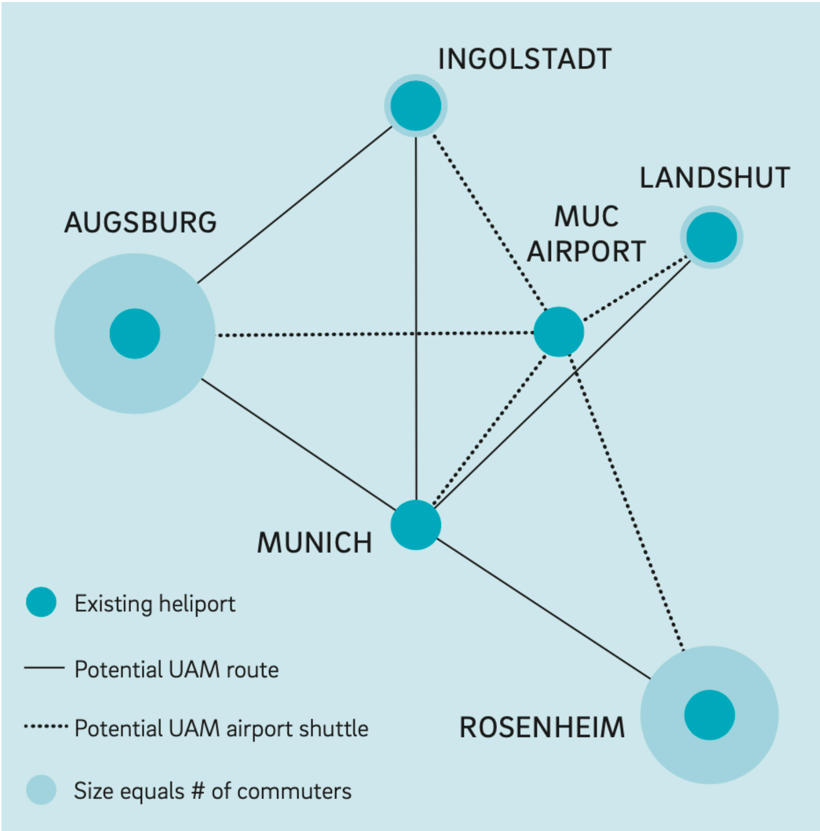

Figure 5.2 UAM Structure of the Munich Case Study

5.1.2 Autonomous Vehicle

The progress of Autonomous Vehicle (AV) is promising worldwide. Many investors such as venture capitals regard AV as a highly potential market. For example, the American startup Nuro received nearly one billion US dollars from Japan's Softbank (Roland Berger, 2019). In addition to investment activities, transport modelling of AV is an attractive area in academia. Hörl et al. (2018) conducted a case study in Zurich, Switzerland, simulating automated fleet vehicles in MATsim.

5.1.3 Shared Mobility

Sharing is a new trend in the transportation industry. New business models involving bike sharing, ride sharing, and scooter sharing are transforming the traveling habits. The sharing mobility grows rapidly, with 20%-30% increase annually (Roland Berger, 2014).

Martinez et al. (2015) presented an agent-based simulation model to evaluate the shared mobility in a central dispatching system. The simulation model replicated a typical working day in real-time, by setting rules for spatial and temporal matching between passengers and shared vehicles. A case study in Lisbon, Portugal, was implemented to evaluate the simulation results of the shared mobility, with the measurements of waiting time and demand elasticity. The results recognized the effectiveness of the proposed model which was able to incorporate characteristics of involved agents.

5.2 Blockchain

A Blockchain is a decentralized and distributed ledge that links by a list of blocks (Zyskind et al., 2015). The blockchain technology has been widely applied in the finance industry. Many companies are exploring the potential of Blockchain technology in the transportation industry. Roland Berger (2017) analyzed a case study in Belgium to facilitate multi-modality using Blockchain technology, as shown in Figure 5.3 (Roland Berger, 2017). Belgium has the objective to achieve multi-modality in the entire country due to the increasing problem of traffic congestion. The difficulty to achieve the objective is that there are various companies providing transportation services such as public transportation companies (i.e., SNCB, STIB, Tec, and De Lijn) and numerous private companies (i.e., Drive Now and Cambio Car Sharing). Sharing data bases of these transportation service providers can be costly under other data management system. Blockchain as a decentralized system can be implemented to reduce operation and maintenance cost.

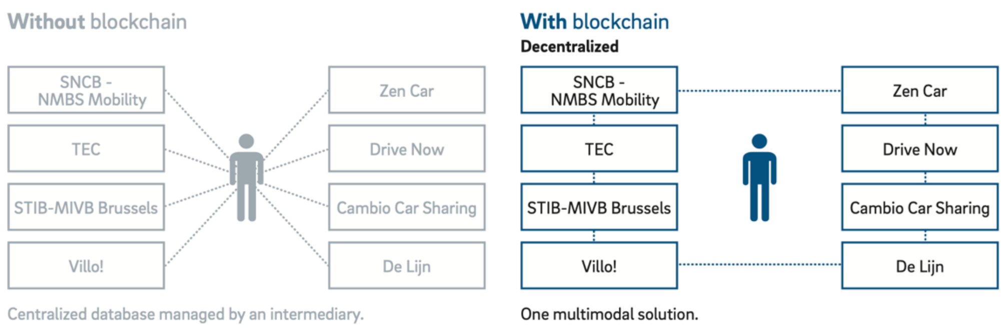

Figure 5.3 A Blockchain Approach for Multi-Modality in Belgium

When considering DTA model for traffic management and operation, Blockchain technology provides a shared data bases involving different transportation mode provided by numerous service providers. The shared data base can be used for online calibration of the DTA model, as presented in Figure 5.4.

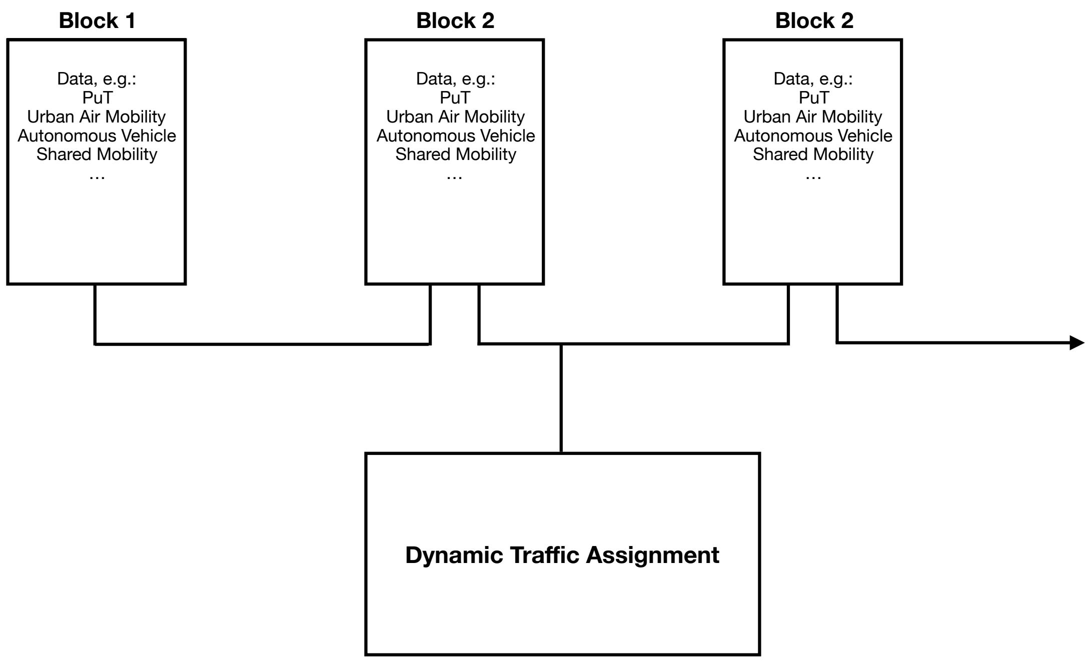

Figure 5.4 Application of Blockchain in DTA Model Calibration

Chapter 6 Conclusion

A systematic framework for offline DTA model calibration is presented. The calibration problem is formulated as a multi-objective function with different traffic measurements for the supply and demand component respectively. Considering that the nature of model calibration is a stochastic problem, we introduce IGA, a heuristic optimization algorithm, to facilitate the calibration process for the supply component.

A synthetic case study and a case study on a network of Munich are used to validate the proposed methodology. We calibrate the car-following model for the synthetic network and the link speeds for the Munich network. The calibration results indicate that IGA significantly improve model accuracy. In addition, IGA outperforms standard GA with the same configuration. Thanks to the existence of the isolated islands, IGA maintains genetic diversity and therefore evolves much faster than the standard GA. With the enlarged search space, IGA overcomes the problem of trapping in local dilemma that often occurs to the standard GA, thus providing nearly global optimum solution.

In this research, our case study is based on a synthetic scenario due to absence of observed traffic data. Further research is required to collect real traffic data and to extend the offline calibration framework to online context. Simultaneous calibration of DTA model using IGA is of our interest. In the future, various transportation modes are available. Blockchain technology provides a way for traffic data management. The Blockchain based traffic data system can be used for online DTA model calibration.

List of References

Antoniou, C., Balakrishna, R., Koutsopoulos, H.N., Ben-Akiva, M., 2011. Calibration Methods for Simulation-Based Dynamic Traffic Assignment Systems. Int. J. Model. Simul. 31, 227–233.

Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., Montero, L., Nigro, M., Perarnau, J., Punzo, V., Toledo, T., van Lint, H., 2016. Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation. Transp. Res. Part C Emerg. Technol., Advanced Network Traffic Management: From dynamic state estimation to traffic control 66, 79–98.

Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2007. Nonlinear Kalman Filtering Algorithms for On-Line Calibration of Dynamic Traffic Assignment Models. IEEE Trans. Intell. Transp. Syst. 8, 661–670.

Antoniou, C., Koutsopoulos, H.N., 2006. Estimation of Traffic Dynamics Models with Machine-Learning Methods. Transp. Res. Rec. 1965, 103–111.

Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., Wen, Y., 2007a. Calibration of Microscopic Traffic Simulation Models: Methods and Application. Transp. Res. Rec. 1999, 198–207.

Balakrishna, R., Ben-Akiva, M., Koutsopoulos, H.N., 2007b. Offline Calibration of Dynamic Traffic Assignment: Simultaneous Demand-and-Supply Estimation. Transp. Res. Rec. 2003, 50–58.

Ben-Akiva, M.E., Gao, S., Wei, Z., Wen, Y., 2012. A dynamic traffic assignment model for highly congested urban networks. Transp. Res. Part C Emerg. Technol. 24, 62–82.

Calégari, P., Guidec, F., Kuonen, P., Kobler, D., 1997. Parallel Island-Based Genetic Algorithm for Radio Network Design. J. Parallel Distrib. Comput. 47, 86–90.

Chiappone, S., Giuffrè, O., Granà, A., Mauro, R., Sferlazza, A., 2016. Traffic simulation models calibration using speed–density relationship: An automated procedure based on genetic algorithm. Expert Syst. Appl. 44, 147–155.

Chow, J.Y.J., Regan, A.C., Arkhipov, D.I., 2010. Faster Converging Global Heuristic for Continuous Network Design Using Radial Basis Functions: Transp. Res. Rec.

Cobos, C., Erazo, C., Luna, J., Mendoza, M., Gaviria, C., Arteaga, C., Paz, A., 2016. Multi-objective Memetic Algorithm Based on NSGA-II and Simulated Annealing for Calibrating CORSIM Micro-Simulation Models of Vehicular Traffic Flow, in: Luaces, O., Gámez, J.A., Barrenechea, E., Troncoso, A., Galar, M., Quintián, H., Corchado, E. (Eds.), Advances in Artificial Intelligence, Lecture Notes in Computer Science. Springer International Publishing, pp. 468–476.

Collins, R.J., Jefferson, D.R., 1991. Selection in Massively Parallel Genetic Algorithms, in: ICGA.

Eby, D., Averill, R.C., Punch, W.F., Goodman, E.D., 1999. Optimal design of flywheels using an injection island genetic algorithm. AI EDAM 13, 327–340.

Flötteröd, G., Bierlaire, M., Nagel, K., 2011. Bayesian Demand Calibration for Dynamic Traffic Simulations. Transp. Sci. 45, 541–561.

Friend, D.H., Nainay, M.Y.E., Shi, Y., MacKenzie, A.B., 2008. Architecture and Performance of an Island Genetic Algorithm-Based Cognitive Network, in: 2008 5th IEEE Consumer Communications and Networking Conference. Presented at the 2008 5th IEEE Consumer Communications and Networking Conference, pp. 993–997.

Gangi, M.D., 2011. Modeling Evacuation of a Transport System: Application of a Multimodal Mesoscopic Dynamic Traffic Assignment Model. IEEE Trans. Intell. Transp. Syst. 12, 1157–1166.

Henderson, J., Fu, L., 2004. Applications of genetic algorithms in transportation engineering, in: 83rd Annual Meeting of the Transportation Research Board.

Holland, J.H., 1992. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K.W., 2018. Fleet control algorithms for automated mobility: A simulation assessment for zurich. Transp. Res. Part C Emerg. Technol.

Hu, X., Chen, X., Zhao, Y., Yao, W., 2014. Optimization design of satellite separation systems based on Multi-Island Genetic Algorithm. Adv. Space Res. 53, 870–876.

Kim, S.-J., Kim, W., Rilett, L.R., 2005. Calibration of Microsimulation Models Using Nonparametric Statistical Techniques. Transp. Res. Rec. 1935, 111–119.

Krauß, S., 1998. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics (PhD Thesis). Dt. Zentrum für Luft-und Raumfahrt eV, Abt. Unternehmensorganisation.

Kurdi, M., 2015. A new hybrid island model genetic algorithm for job shop scheduling problem. Comput. Ind. Eng. 88, 273–283.

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., WieBner, E., 2018. Microscopic Traffic Simulation using SUMO, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). Presented at the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2575–2582.

Lu, L., Xu, Y., Antoniou, C., Ben-Akiva, M., 2015. An enhanced SPSA algorithm for the calibration of Dynamic Traffic Assignment models. Transp. Res. Part C Emerg. Technol. 51, 149–166.

Ma, T., Abdulhai, B., 2002. Genetic Algorithm-Based Optimization Approach and Generic Tool for Calibrating Traffic Microscopic Simulation Parameters. Transp. Res. Rec. 1800, 6–15.

Mahut, M., Florian, M., Tremblay, N., Campbell, M., Patman, D., McDaniel, Z.K., 2004. Calibration and Application of a Simulation-Based Dynamic Traffic Assignment Model. Transp. Res. Rec. 1876, 101–111.

Martinez, L.M., Correia, G.H.A., Viegas, J.M., 2015. An agent-based simulation model to assess the impacts of introducing a shared-taxi system: an application to Lisbon (Portugal). J. Adv. Transp. 49, 475–495.

Meng, Q., Wu, J., Ellis, J., Kennedy, P.J., 2017. Dynamic island model based on spectral clustering in genetic algorithm, in: 2017 International Joint Conference on Neural Networks (IJCNN). Presented at the 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1724–1731.

Merchant, D.K., Nemhauser, G.L., 1978. A Model and an Algorithm for the Dynamic Traffic Assignment Problems. Transp. Sci. 12, 183–199.

Mühlenbein, H., 1992. PARALLEL GENETIC ALGORITHMS IN COMBINATORIAL OPTIMIZATION, in: Balci, O., Sharda, R., Zenios, S.A. (Eds.), Computer Science and Operations Research. Pergamon, Amsterdam, pp. 441–453.

Omrani, R., Kattan, L., 2013. Simultaneous calibration of microscopic traffic simulation model and estimation of origin/destination (OD) flows based on genetic algorithms in a high-performance computer, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). Presented at the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 2316–2321.

Osorio, C., 2019. Dynamic origin-destination matrix calibration for large-scale network simulators. Transp. Res. Part C Emerg. Technol. 98, 186–206.

Osorio, C., Punzo, V., 2019. Efficient calibration of microscopic car-following models for large-scale stochastic network simulators. Transp. Res. Part B Methodol. 119, 156–173.

Peeta, S., Mahmassani, H.S., 1995. System optimal and user equilibrium time-dependent traffic assignment in congested networks. Ann. Oper. Res. 60, 81–113.

Pereira, C.M.N.A., Lapa, C.M.F., 2003a. Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem. Ann. Nucl. Energy 30, 555–565.

Pereira, C.M.N.A., Lapa, C.M.F., 2003b. Parallel island genetic algorithm applied to a nuclear power plant auxiliary feedwater system surveillance tests policy optimization. Ann. Nucl. Energy 30, 1665–1675.

Protopapadakis, E., Doulamis, A., Makantasis, K., Voulodimos, A., 2012. A semi-supervised approach for industrial workflow recognition, in: Proceedings of the Second International Conference on Advanced Communications and Computation (INFOCOMP 2012), Venice, Italy. pp. 21–26.

Qin, X., Mahmassani, H.S., 2004. Adaptive Calibration of Dynamic Speed-Density Relations for Online Network Traffic Estimation and Prediction Applications. Transp. Res. Rec. 1876, 82–89.

Qurashi, M., Ma, T., Chaniotakis, E., Antoniou, C., 2019. PC-SPSA: Employing Dimensionality Reduction to Limit SPSA Search Noise in DTA Model Calibration. IEEE Trans. Intell. Transp. Syst. 1–11.

Razali, N.M., Geraghty, J., 2011. Genetic Algorithm Performance with Different Selection Strategies in Solving TSP.

Rothfeld, R., Balac, M., Ploetner, K.O., Antoniou, C., 2018. Agent-based Simulation of Urban Air Mobility, in: 2018 Modeling and Simulation Technologies Conference. p. 3891.

Roland Berger, 2017. Blockchain: A promising technology for the Belgian public administration. Think: Act Magazine, 2017, 11. https://www.rolandberger.com/publications/publication_pdf/roland_berger_blockchain_a_promising_technology_for_the_belgian_public_administr.pdf

Roland Berger, 2018. Urban air mobility: The rise of a new mode of transportation. Think: Act Magazine, 2018, 11. https://www.rolandberger.com/publications/publication_pdf/Roland_Berger_Urban_Air_Mobility.pdf

Roland Berger, 2019. Mobility's great leap forward: a world on the verge of autonomy. Think: Act Magazine, 2019, Issue #5. https://www.rolandberger.com/publications/publication_pdf/roland_berger_automotive_disruption_radar_mobilititys_leap_forward.pdf

Roland Berger, 2014. Shared Mobility: How new businesses are rewriting the rules of the private transportation game. Think: Act Magazine, 2014. 07. https://www.rolandberger.com/publications/publication_pdf/roland_berger_tab_shared_mobility_1.pdf

Roland Berger, 2018. The blockchain bandwagon: Is it time for automotive companies to start investing seriously in blockchain? Think: Act Magazine, 2017, 09. https://www.rolandberger.com/publications/publication_pdf/roland_berger_blockchain_automotive.pdf

Runmei Li, Wei Li, 2005. The application of genetic algorithm to dynamic traffic assignment, in: IEEE Proceedings. Intelligent Vehicles Symposium, 2005. Presented at the IEEE Proceedings. Intelligent Vehicles Symposium, 2005., pp. 827–832.

Scrucca, L. (2013). GA: a package for genetic algorithms in R. Journal of Statistical Software, 53(4), 1-37.

Seyedabrishami, S., Nazemi, M., Shafiei, M., 2014. Off-line calibration of a macroscopic Dynamic Traffic Assignment model: Iterative demand-supply parameters estimation, in: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). Presented at the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 2035–2040.

Shafiei, S., Gu, Z., Saberi, M., 2018. Calibration and validation of a simulation-based dynamic traffic assignment model for a large-scale congested network. Simul. Model. Pract. Theory 86, 169–186.

Spall, J.C., 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans. Aerosp. Electron. Syst. 34, 817–823.

Starkweather, T., Whitley, D., Mathias, K., 1991. Optimization using distributed genetic algorithms, in: Schwefel, H.-P., Männer, R. (Eds.), Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 176–185.

Toledo, T., Ben-Akiva, M.E., Darda, D., Jha, M., Koutsopoulos, H.N., 2004. Calibration of Microscopic Traffic Simulation Models with Aggregate Data. Transp. Res. Rec. 1876, 10–19.

Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2018. Robust SPSA algorithms for dynamic OD matrix estimation. Procedia Comput. Sci., The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops 130, 57–64.

Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2015. c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation. Transp. Res. Part C Emerg. Technol., Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue 55, 231–245.

Van Leeuwen, J., Leeuwen, J., 1990. Handbook of theoretical computer science. Elsevier.

Varia, H.R., Gundaliya, P.J., Dhingra, S.L., 2013. Application of genetic algorithms for joint optimization of signal setting parameters and dynamic traffic assignment for the real network data. Res. Transp. Econ., Economics of Sustainable Transport in India 38, 35–44.

Vaze, V., Antoniou, C., Wen, Y., Ben-Akiva, M., 2009. Calibration of Dynamic Traffic Assignment Models with Point-to-Point Traffic Surveillance. Transp. Res. Rec. 2090, 1–9.

Whitley, D., Rana, S., Heckendorn, R.B., 1999. The island model genetic algorithm: On separability, population size and convergence. J. Comput. Inf. Technol. 7, 33–47.
Wismans, L.J.J., Van Berkum, Eric.C., Bliemer, M.C.J., 2011. Comparison of Multiobjective Evolutionary Algorithms for Optimization of Externalities by Using Dynamic Traffic Management Measures. Transp. Res. Rec. 2263, 163–173.

Zhang, C., Osorio, C., Flötteröd, G., 2017. Efficient calibration techniques for large-scale traffic simulators. Transp. Res. Part B Methodol. 97, 214–239.

Zhang, H., Seshadri, R., Prakash, A.A., Pereira, F.C., Antoniou, C., Ben-Akiva, M.E., 2017. Improved Calibration Method for Dynamic Traffic Assignment Models: Constrained Extended Kalman Filter. Transp. Res. Rec. 2667, 142–153.

Zhou, X., Taylor, J., 2014. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration. Cogent Eng. 1, 961345.

Zyskind, G., Nathan, O., Pentland, A. “Sandy,” 2015. Decentralizing Privacy: Using Blockchain to Protect Personal Data, in: 2015 IEEE Security and Privacy Workshops. Presented at the 2015 IEEE Security and Privacy Workshops, pp. 180–184.

Appendix

This appendix presents some codes used in DTA model calibration. We connect two programming languages R and Python in setting simulation.

Python Script for Generation of the Synthetic Network

```
#generate trips
def van_trips(Paths,Network_char,Network_var,Network_no):

    Network_path      = Paths["pathtoCaseOutput"] + str(Network_no) + '/'
    Net_file_path     = Network_path + str(Network_no) + '.net.xml'
    Van_Trip_file     = Network_path + str(Network_no) + 'van.trip.xml'
    csv_file          = Network_path + str(Network_no) + 'van_trips.csv'

    Van_Trips = 'python ' '"' + Paths["pathtoSUMO"] + 'tools/randomtrips.py" '\
            ' -n ' '"' + Net_file_path + '"\
            ' -b ' + Network_var["beginSimtime"] + ' -e ' +
Network_var["endSimtime"] +\
            ' --persontrips ' ' -p '+ Network_var["Random_van_trips"] + \
            ' -o ' '"' + Van_Trip_file + '"'

    subprocess.run(Van_Trips, shell=True)
    bs.xml2csv(Paths, Network_var, Van_Trip_file, csv_file)
    van_trips = pd.read_csv(csv_file, sep = ';', skiprows =0, header=0)
    return van_trips

#generate routes
def van_routes(Paths,Network_char,Network_var,Network_no):

    Network_path      = Paths["pathtoCaseOutput"] + str(Network_no) + '/'
    Net_file_path     = Network_path + str(Network_no) + '.net.xml'
    Van_Trip_file     = Network_path + str(Network_no) + 'van.trip.xml'
    Van_route_file    = Network_path + str(Network_no) + 'van.route.xml'
    csv_file          = Network_path + str(Network_no) + 'van_routes.csv'

    Van_route = '"' + Paths["pathtoSUMOBIN"] + 'duarouter" --no-step-log'\
            ' -n ' '"' + Net_file_path + '"' ' --route-files ' '"' + Van_Trip_file + '"\
            ' -o ' '"' + Van_route_file + '"'
```

```
    subprocess.run(Van_route, shell=True)
    bs.xml2csv(Paths, Network_var, Van_route_file, csv_file)
    van_routes = pd.read_csv(csv_file, sep = ';', skiprows =0, header=0)
    return van_routes
```

Python Script for Execution of SUMO Simulation

```
#convert net.csv to net.xml
    xml_output = Paths['ScenarioDataFolder'] + str(i) + '.net.xml'
    csv_input = Paths['ScenarioDataFolder']   + str(i) + '_net.csv'
    xml_schema = 'net_file.xsd'
    bs.csv2xml(Paths, Network_var, xml_output, xml_schema, csv_input)

#run sumo and generate vehroutes.xml
    Net_file_path= Paths['ScenarioDataFolder'] + str(i) + '.net.xml'
    Trip_file_path= Paths['ScenarioDataFolder'] + 'Munich.trips.xml'
    VehRoutes_file_path= Paths['ScenarioDataFolder'] + str(i) + '.vehroutes.xml'

    Routes = 'sumo' + ' -n ' '"' + Net_file_path + '"' \
             ' -r ' '"' + Trip_file_path + '"'\
             ' --vehroutes ' '"' + VehRoutes_file_path + '"'\
             ' --mesosim '

    subprocess.run(Routes, shell=True)

#convert vehroutes.xml to vehroutes.csv
    xml_file = Paths['ScenarioDataFolder'] + str(i) + '.vehroutes.xml'
    csv_file = Paths['ScenarioDataFolder'] + str(i) + '_vehroutes.csv'
    bs.xml2csv(Paths, Network_var, xml_file, csv_file)

if __name__ == "__main__":
   main(sys.argv[1])
```

Rscript for DTA Model Calibration

```
f <-   function(x){
  i<- rnorm(1)
  net2c$lane_speed <- as.character(ceiling(x))
```

```
  net<-rbind(net2c,netX)
  file_net<-paste(i,"_net.csv",sep = "")
  write.csv2(net,file= file_net ,row.names = F)
  sumo <- paste("python Munich.py", i)
  system(sumo)
  file_vehroutes<-paste(i,"_vehroutes.csv",sep = "")
  simulated_vehroutes<-read.csv2(file = file_vehroutes, colClasses = 'character')
  simulated_vehroutes$travel_time <-
as.numeric(simulated_vehroutes$vehicle_arrival) -
as.numeric(simulated_vehroutes$vehicle_depart)
  sim <- simulated_vehroutes %>%
    group_by(vehicle_fromTaz, vehicle_toTaz) %>%
    summarize(t = mean (as.numeric(travel_time)))
  RMSE<-rmse(sim$t, obs$t)
  file_vehroutesxml<-paste(i,".vehroutes.xml",sep = "")
  file_netxml<-paste(i,".net.xml",sep = "")
  file.remove(file_net)
  file.remove(file_vehroutes)
  file.remove(file_vehroutesxml)
  file.remove(file_netxml)
  return(-RMSE)
}
```

Rscript for Execution of GA and IGA

```
GA <- ga(type = "real-valued", fitness = f,
          lower = low, upper = up,
          popSize = 200, maxiter = 100,
          parallel = T)

IGA <- gaisl(type = "real-valued", fitness = f,
          lower = low, upper = up,
          popSize = 50, maxiter = 50,
          numIslands = 4)
```