Title
Developing Calibration Tools for Microscopic Traffic Simulation Final Report Part III: Global Calibration - O-D Estimation, Traffic Signal Enhancements and a Case Study

Authors
Zhang, Michael
Ma, Jingtao
Singh, Shailendra P.
et al.

Publication Date
2008-06-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Developing Calibration Tools for Microscopic Traffic Simulation Final Report Part III: Global Calibration - O-D Estimation, Traffic Signal Enhancements and a Case Study

Michael Zhang, Jingtao Ma,
Shailendra P. Singh, Lianyu Chu

California PATH Research Report
UCB-ITS-PRR-2008-8

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation, and the United States Department of Transportation, Federal Highway Administration.

Final Report for Task Order 5308

June 2008

ISSN 1055-1425

T.O. 5308

DEVELOPING CALIBRATION TOOLS FOR MICROSCOPIC TRAFFIC SIMULATION

FINAL REPORT

PART III: GLOBAL CALIBRATION: O-D ESTIMATION, TRAFFIC SIGNAL ENHANCEMENTS, AND A CASE STUDY

April 19, 2007

Michael Zhang, Jingtao Ma, Shailendra P. Singh
Institute of Transportation Studies
University of California
Davis, CA 95616
and
Lianyu Chu
Institute of Transportation Studies
University of California
Irvine, CA 92697

ACKNOWLEDGEMENTS

We wish to thank Nathan Loebs, Division of Research and Innovation, and Steve Hague, Division of Traffic Operations, Caltrans, for their valuable inputs during various stages of this project.

Thanks also go to Professor Will Recker of University of California, Irvine, Henry X. Liu of University of Minnesota, Danny Pishdadian of Caltrans District 12, and Gabriel Gomes and Meng Li of California PATH program.

ABSTRACT

The central goal of this research is to develop a systematic framework and the support tools to ease, streamline and speed up the calibration of micro simulation projects. Part III of the final report documents the accomplishments achieved in the second phase of the research project. They include the following.

First, to overcome the lengthy time it takes for GA to obtain local and global driving behavior modeling parameters, we implemented a faster heuristic optimization technique, the simultaneous perturbation stochastic approximation (SPSA) and compared its performance with other heuristic optimization methods. Results indicate that SPSA can achieve comparable calibration accuracy with much less computational time than the often used Genetic Algorithm (GA) method.

Second, we developed a much faster O-D estimation tool to obtain an initial time-dependent O-D trip table. This O-D trip table can be used as a seed table in Paramics' own O-D estimator for further refinement, or directly used in a micro simulation. In either case, the estimation time of O-D trip tables can be considerably shortened. Since our O-D estimation tool makes use of a macroscopic traffic model (logit path flow estimator, or LPFE), a network conversion tool is therefore developed to convert Paramics's detailed network settings to those of LPFE and vice versa.

Third, we enhanced the vehicle actuated signal control APIs in Paramics, making it more versatile to implement and simulate various types of actuated traffic control strategies found in practice. We also developed a set of guidelines to help micro simulation users to set up and check signal settings in a micro simulation project.

Finally, we developed a summary statistics tools to track, diagnose and report on the calibration as it progresses or after it terminates, and carried out a case study using the SR-41 network in Fresno to demonstrate the use of the developed tools, identify potential problems and summarizing our calibration experiences with large scale networks.

Our case study indicates that the developed calibration tools can indeed ease, streamline and speed up the calibration of micro simulation, particularly when the network concerned is large. It also reveals that the calibration of a micro simulation is a complex task that involves numerous engineering judgments and cannot be fully automated. In a micro simulation, every modeling detail matters and each must be treated properly to ensure a good simulation outcome.

Keywords: SPSA, O-D Estimation, Summary Statistics, Actuated Traffic Signals, Case Study.

EXECUTIVE SUMMARY

The main objective of this research is to develop a systematic framework and the support tools to ease, streamline and speed up the calibration of micro simulation projects. In Part III of this final report, we present the accomplishments achieved in the second phase of the research project, which include:

- The Implementation of a faster heuristic optimization technique, the simultaneous perturbation stochastic approximation (SPSA) method to calibrate local and global model parameters, and compared its performance with other heuristic optimization methods. Results indicate that SPSA can achieve comparable calibration accuracy with much less computational time than the often used Genetic Algorithm (GA) method.

- The development of an O-D estimation tool based on the logit path flow estimator (LPFE), which uses a macroscopic representation of traffic and networks. A network conversion tool is therefore developed to convert Paramics's detailed network settings to those of LPFE and vice versa. It was shown that with the seed O-D provided by LPFE, Paramics' O-D Estimator can usually find a good O-D demand pattern (one that produces the closest match to observed traffic counts and/or travel times) much more quickly. The time-dependent O-D trips tables provided by LPFE can also be directly used in a micro simulation when there is not time or no means to further refine them using the O-D estimator that comes with a particular simulation.

- The enhancement of vehicle actuated signal control APIs in Paramics and the development of a set of guidelines to help set up and check signal settings in a micro simulation. Based on previous efforts (e.g., Liu, Chu& Recker, 2001), an actuated signal control plugin was extended to include a volume-density control function, detector placement and output diagnoses, and a conversion of SEPAC coordination data to a format acceptable by Caltrans C8's logic. A detailed, step by step procedure to code, diagnose and optimize arterial traffic signals in Paramics is provided. These can all be used together to ease the effort of calibrating traffic signal operations and improve simulation performance.

- The development of a summary statistics tool to monitor, diagnose, and report on the calibration of local and global driving behavior parameters. This tool produces the fundamental diagram from simulated data for selected locations, so that one can judge if the capacity and shape of the fundamental diagram of any selected location is reproduced by the simulation. It also generates a convergence curve from which one can judge if the calibration is progressing well and when to terminate it. Besides these visual aids, the summary statistics tool also allows users to save intermediate and final calibration results in a Microsoft Excel file for later analysis. And finally,

- The completion of a case study using the SR-41 network in Fresno to demonstrate the use of the developed tools, evaluate their effectiveness, and summarize our calibration

experiences with large scale networks. Following our five-step calibration procedure, detailed error-checking of the network settings was first performed, and certain coding problems were corrected. Then local and global parameter calibration on selected sites was performed using the developed tools, and the time-dependent O-D trip tables for the 2-hour PM peak period was obtained using our own O-D estimation tool, LPFE. The calibrated simulation showed significant improvement in matching observed traffic conditions over the original network settings, and this was accomplished in much less time than it would have taken without using the developed tools. Nevertheless, our case study also revealed some critical issues of micro simulation and areas for further improvement. These include improved lane utilization models, proper treatment of blocking traffic at intersections, and elimination of software conflicts between Paramics' O-D Estimator and the traffic signal control API.

While the developed tools prove to be quite helpful in reducing calibration time and improving calibration results, several improvements to the developed tools can be made in future work. One would be the further development of the network conversion tool between LPFE and Paramics' networks, since at this stage the conversion still needs quite an amount of human intervention. Another improvement would be expanding the control logic of the developed signal control APIs, so that more controller types can be simulated in Paramics. Last but not least, the Paramics simulation itself also needs further enhancement. For example, it was observed in our case study that lane utilization and intersection blocking could create quite serious problems for calibration, but it is not straightforward to fix such problems in Paramics. Besides these improvements, emerging ATMIS features should also be incorporated into the microscopic simulation packages and their calibration tools.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION AND OVERVIEW

This report describes the second phase of the project “DEVELOPMENT OF A CALIBRATION PROCEDURE AND ASSISTANCE TOOLS FOR MICO SIMULATION”. This project aims to develop an effective calibration procedure for microscopic traffic simulation, and a set of tools to streamline and automate as much as possible the calibration process, which is usually time-consuming and labor intensive.

Developing a micro simulation application usually takes a few steps that include data collection, network coding, baseline model calibration and model application. Among these steps, model calibration is a process of adjusting the model parameters to fit local conditions. It is often a critical step towards a successful application of micro simulation, because without this step one cannot be sure that the developed application reflects the local traffic conditions, hence its results are reliable. Because its high fidelity and stochastic nature, micro simulation often involves a great number of parameters that need to be calibrated application by application, therefore a systematic calibration procedure with support tools to automate as much as possible the often tedious and time-consuming manual calibration work is highly desirable.

In the first phase of this project (Year 1), the study performed an extensive review of literature and calibration experiences for a variety of micro simulation applications. That review identified several deficiencies in past calibration studies, which include: lack of a systematic procedure, ad hoc calibration methods, not enough effort to disentangle the convoluted parameter space, and repetitions of tasks due to poor project planning. These findings, together with a set of guidelines for project planning, data collection and checking coding errors, are documented in the Phase I (Year-1) report and will not be repeated in this report.

Base on the above findings, our subsequent study proposed a comprehensive, systematic calibration framework (Figure 1.1 in Zhang, Ma & Dong 2006) that consists of the following five steps: 1) error-checking, 2) global parameter calibration, 3) local parameter calibration, 4) departure time and route choice calibration and 5) demand estimation. In Year-1, support tools have been developed to ease the effort of the first four steps, and micro simulation networks developed in California have been used to evaluate them. In the evaluation tests, the tools were shown to have greatly reduced the amount of time spent on calibration while achieving more reliable calibration results. Readers are referred to (Zhang & Ma 2006; Zhang, Ma & Dong 2006) for detailed descriptions of the developed procedure and calibration support tools.

The tasks completed in Phase II, which are the subject of this report, included the following.

First, various heuristic optimization techniques are tested and evaluated. When developing the support tools to conduct global/local and D-R choice parameter calibration, we have found that different optimization algorithms have varying performances in terms of reliability and computational efficiency. For example, some algorithms are quite fast to converge but produced rougher parameter estimates while others are slow to converge but were able to produce better parameter estimates. It is therefore desirable to implement a suite of optimization algorithms for differing calibration needs. In Phase II we implemented another promising search algorithm-simultaneous perturbation stochastic approximation, and compared it with others. These results are reported in Chapter 2 of this report.

Second, a traffic demand estimation tool is developed to aid the calibration of O-D demand. Demand estimation is a vital step in calibration and various micro simulation packages, such as Paramics, offer their own O-D estimation tools. In the case of Paramics, it was found that the estimation of O-D demand was often slow if one started with a poor initial seed O-D matrix. The implemented O-D estimation tool makes use of the results from a previous PATH project (TO 5502, "DEVELOPMENT OF A PATH FLOW ESTIMATOR FOR DERIVING STEADY-STATE AND TIME-DEPENDENT ORIGIN-DESTINATION TRIP TABLES"), the time-dependent path flow estimator (TD-PFE), and developed an interface between TD-PFE and Paramics. It was shown that with the seed O-D provided by TD-PFE, Paramics' O-D Estimator can usually find a good O-D demand pattern (one that produces the closest match to observed traffic counts and/or travel times) much more quickly. These results are reported in Chapter 3.

Third, this project considerably enhanced the traffic control functions of Paramics through API development. Based on previous efforts (e.g., Liu, Chu& Recker, 2001), actuated signal control API could simulate the signal control functions prevalent in California. Meanwhile, a step-by-step guideline is also documented in Chapter 4 to facilitate the analyst calibrate the traffic signal functioning in Paramics.

Fourth, a summary statistics tool is developed to review the goodness of fit of the calibrated parameters. The calibration process represented by the convergence curve and the quality of the calibrated simulation model represented by the fundamental diagram, are both compiled automatically into a presentation format (tables and figures in EXCEL). This tool would greatly ease the analyst to monitor and report the calibration work.

Finally, a case study with a real network (State Route 41 in Fresno) was performed to demonstrate the developed calibration procedure and support tools. This network has been coded by Caltrans District 6 engineers; the case study indicates that the tools can save a great amount of modeling time while achieving better performance. Some challenges have also been identified in the case study.

CHAPTER 2 PARAMETER OPTIMIZATION ALGORITHMS

2.1 Introduction

In the developed toolkit, the components of calibrating the global, local, departure time and route choice (D-R) parameters all apply the same heuristic optimization algorithm, namely the genetic algorithm (GA), a widely used global search method in micro simulation studies (Cheu et al. 1998, Lee, Yang & Chandrasekar 2001, Ma & Abdulhai 2002, Kim & Rillet 2004, Ma, Zhang & Dong 2006). Because of the stochastic nature of micro simulation and the complex relations between model parameters and simulation outcomes, gradient information is generally difficult to obtain and mathematical programming methods are hardly applicable. However, many heuristic optimization methods besides GA are available to find "optimal" parameter values, i.e., those values that minimize the objective function – some form of "distance" metric between simulation results and real world traffic measurements. Examples of such algorithms include simulated annealing (Hoyer and Fellendorf 1997), the complex algorithm (Ben-Akiva et al 2004) and the most rudimentary of them all--enumeration (Gomez, May & Horowitz. 2004). It was generally reported that they improve simulation performance over the default model parameter values. At the same time, all methods automate the calibration process to a certain degree.

Yet several important questions remain unanswered pertaining to these heuristic optimization algorithms. Even though all heuristics methods, including the genetic algorithm used in our study, were reported to produce improved calibration results, did they obtain the global ("true" optimal set of parameter values, and if not, how far are their results away from the globally optimal parameter values? Which method can obtain similar or better calibration results with less computational effort? The latter is important to know as micro simulations are applied to larger and larger networks. Answers to these questions will help the analysts carry out model calibration more efficiently and reliably.

In this chapter we attempt to find an answer to the above questions through calibrating the driving behavior model parameters in Paramics. We first introduce our parameter optimization technique, the simultaneous perturbation stochastic approximation (SPSA) method (Spall 1992), a stochastic optimization technique that has been used successfully in other fields (Spall 1998). This heuristic method differs from others in that it does not rely on evaluating a large pool of feasible solutions when updating the search direction in each iteration. Meanwhile, the search direction is along the approximated gradient in every iteration. After obtaining a set of parameter values with SPSA, we then use GA and/or trial-and-error iterative adjustments to calibrate this same set of parameters and compare their calibration results with the SPSA results. Next, we apply these three methods to the calibration of SR-99 (Zhang, Ma & Dong 2006), which sheds lights on some of the questions raised above. Finally, we provide some guidelines of applying these heuristic calibration methods.

2.2 Heuristic Calibration Methods

Most heuristic methods start with a feasible set(s) of parameter values, apply them to obtain model results and then compare them with field measurements. Based on rules unique to each method, parameter value sets with poor modeling results (unfit) will be discarded and replaced by ones that could produce better results. This process is carried out iteratively until the gap between measured and modeled outputs is narrowed to an acceptable level. Three heuristics methods are compared in this chapter: Simultaneous Perturbation Stochastic Approximation (SPSA), Genetic Algorithm (GA), and trial-and-error Iterative Adjustment (IA). Since the last two are widely applied and have been documented in the Year-1 report, they are only briefly presented here. Our major coverage is on the SPSA method.

2.3 The Simultaneous Perturbation Stochastic Approximation (SPSA) Algorithm

This introduction of the SPSA method draws mainly on the theoretical work from (Spall 1992, Spall 1998, Sadegh 1997).

The simultaneous perturbation stochastic approximation (SPSA) method works in the following way. For a modeling system, the general objective function $L(\theta)$ is a scalar-valued performance measure, and θ is a continuous-valued p-dimensional vector of the parameters that can be manipulated to change the performance of the modeled system. In the context of micro simulation calibration, θ is the vector of selected parameters to be calibrated, such as target headway, reaction time, driver aggressiveness, etc. It is common that a noise ε could occur when observing $L(\theta)$, that is, the observation $z(\theta)$ would be:

$$z(\theta) = L(\theta) + \varepsilon \tag{2.1}$$

Assuming $L(\theta)$ is differentiable over θ and the minimum is obtained at a zero point of the gradient, i.e.,

$$g(\theta) = \left|\frac{\partial L(\theta)}{\partial \theta}\right|_{\theta=\theta^*} = 0 \tag{2.2}$$

With an initial guess θ_0 (e.g., the default parameter values in Paramics), SPSA applies a series of "simultaneous perturbations" over the successive steps until the approximation of the gradient $g(\theta)$ converges to zero *almost surely* (*a.s.*), under several regularity conditions. The readers are referred to (Spall 1992) for the theoretical development of the regularity conditions. It is observed that for most engineering problems these conditions are almost automatically satisfied (Spall 1992) with only one exception of restricting the objective (fitness) function values not going excessively large in the calibration context. An excessively large fitness value implies that the simulated results cannot represent the real world traffic at all when replacing the default parameters in simulation with the estimated parameters (θ_k). Since it could be avoided by restricting feasibility ranges, this would be unlikely to occur in our calibration process.

Along the successive steps, θ_k is updated recursively in the following way:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}(\hat{\theta}_k) \tag{2.3}$$

where the gain sequence $\{a_k\}$ also needs to satisfy the regularity conditions.

The perturbation is performed upon deriving $\hat{g}(\hat{\theta}_k)$. First define a *p*-dimensional mutually independent mean-zero random variable vector $\Delta_k \in R^p = \{\Delta_{k1},...,\Delta_{kp}\}$ that is also independent of the estimated sequence of θ_k. The expectation of the (or higher) inverse moment of each component of Δ_k must be bounded, i.e., $E(|\Delta_{ki}^{-2}|) \le \alpha_2$. An optimal distribution of Δ_k is symmetric Bernoulli: $P(\Delta_{ki} = \pm 1) = \frac{1}{2}$ (Sadegh & Spall 1998). Let

$$z_k^{(+)}(\theta_k) = L(\hat{\theta}_k + c_k \Delta_k) + \varepsilon_k^{(+)} \tag{2.4}$$

$$z_k^{(-)}(\theta_k) = L(\hat{\theta}_k - c_k \Delta_k) + \varepsilon_k^{(-)} \tag{2.5}$$

where c_k is a positive scalar (A1), and $z_k^{(+)}(\theta_k)$, $z_k^{(-)}(\theta_k)$ are the outputs of the system under the perturbation $\hat{\theta}_k + c_k\Delta_k$, $\hat{\theta}_k - c_k\Delta_k$, respectively. The approximated gradient will read:

$$\hat{g}_k(\hat{\theta}_k) = \frac{z_k^{(+)} - z_k^{(-)}}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{2.6}$$

Spall (1992) shows that by recursively updating θ_k, the gradient will converge to zero that implies a local minimum, since it is unlikely that the approximation would settle down at a maximum or a saddle point because of the stochastic nature of the algorithm.

The gain sequences of a_k and c_k generally take the form of power functions:

$$a_k = \frac{A}{(1+k)^{\alpha}}, \qquad c_k = \frac{1}{(1+k)^{\gamma}} \tag{2.7}$$

where k is the iterator, and A is a constant introduced to stabilize the optimization process.

The above SPSA procedure is suitable for unconstrained optimization, and it has to be adapted to accommodate the constraints posed in our application, i.e., imposing a lower and an upper bound for each parameter (the so-called "box" constraints). Sadegh (1997) proposed a projection method to restrict $\theta_k \in R^p$ at iteration *k* to fall in the feasibility range. It simply replaces any violating $\hat{\theta}_k$ with the nearest $\theta_k \in G(\theta)$, where $G(\theta)$ is the feasibility set of the parameters to be calibrated:

$$\hat{\theta}_{k+1} = P_r(\hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k)) \tag{2.8}$$

The perturbed vectors $\hat{\theta}_k \pm c_k \Delta_k$ in evaluating the system performances will also be projected into the feasibility region in the same manner. By enforcing an additional regularity condition (Proposition 1 in Sadegh 1997) over the constraints, constrained SPSA is still able to converge to a Karash-Kuhn-Tucker point *a.s.*.

Based on the above constrained SPSA method, we develop our SPSA based calibration algorithm as follows:

2.4 SPSA Algorithm for Micro Simulation Calibration

Step 1: Initialization and Selection of Algorithmic Coefficients.

1.0 Set iterator $\kappa = 0$;

1.1 Select the set of parameters to be calibrated as θ *and normalize it;*

1.2 Pick an initial feasible solution of θ_0 *(e.g., default values in the simulation software);*

1.3 Select nonnegative algorithmic parameters $\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{A}, \alpha$ *and* γ.

Step 2: Simultaneous Perturbation.

Generate a p-dimensional random perturbation vector Δ_k, *where each component is mutually independent Bernoulli* ± 1 *distributed with probability of ½ for each* ± 1 *outcome.*

Step 3: Objective Function Evaluation by Running Simulation with Perturbed Parameters.

3.1 Perturb the vector $\hat{\theta}_k$ *with* $\pm c_k \Delta_k$ *as in (2.4-2.5);*

3.2 Project the perturbed vectors onto $G(\theta_k)$ *from (2.7);*

3.3 Evaluate the calibration performance by running simulation with perturbed parameters obtained in (2.4-2.5).

Step 4: Compute the Approximate Gradient.

Calculate the approximated gradient with (2.6).

Step 5: Parameter Update.

Update $\hat{\theta}_k$ *with (2.8).*

Step 6: Check convergence.

Check if convergence criterion is met or the maximum number of iterations has been reached.. If yes, stop. If not, set $\kappa = \kappa + 1$ *and go to step 2.*

Genetic Algorithm (GA)

Genetic algorithm (GA) is a popular calibration method for micro simulation, and has been shown to obtain satisfactory calibration results (e.g., Cheu et al. 1998, Lee, Yang& Chandrasekar 2001, Ma & Abdulhai 2002, Kim& Rillet 2004, Ma, Zhang & Dong 2006). We refer the readers to our Year-1 report (Zhang, Ma & Dong 2006) for an introduction of GA as well as detailed guidelines for its use in calibration applications.

Trial-and-error Iterative Adjustments (IA)

The trial-and-error iterative adjustment method used here first enumerates the feasible solutions by dividing the feasible region into equal intervals and picking a value from each interval, then runs the simulation based on combinations of selected parameter values, often one parameter at a time. One can make the intervals smaller to increase the precision. This process continues until both precision requirements and the performance target are met. This method is simple and easy to apply. Thus, many calibration efforts (Gomez, May & Horowitz 2004, Gardes et al. 2002) rely on trial-and-error to find a suitable set of model parameters. However, the choice of the feasible range and incremental steps of each parameter is quite *ad hoc*, often relying on the analyst's modeling experience and judgment.

2.5 Calibrating Global and Local Driving Parameters Using Various Algorithms

2.5.1 Global and Local Driving Parameters

As stated in the calibration framework (Zhang, Ma & Dong 2006), the driving behavior model parameters are categorized into two related groups: 1) global parameters that affect driving behavior throughout the network, 2) local parameters that are peculiar to bottleneck locations, such as lane-drop locations or junctions where several roads meet, e.g., on-ramp merging sections or intersections.

Global model parameters and local model parameters are calibrated separately in this study. This separation is akin to highway capacity analysis, where one first identifies a set of ideal conditions and obtain the ideal capacity under such conditions, then adjusts the ideal capacity for non-ideal conditions through discount factors to obtain the prevailing capacity. Similarly, we want to identify typical road sections for the calibration of global parameters, and road sections with special features (such as sharp curvatures, lane drops, on-ramps and intersections) for the calibration of local model parameters. Through such a two-step calibration process, we want to obtain a set of parameters that can reproduce the flow capacities of various types of road sections.

Different simulation packages have their own underlying driving behavior models and corresponding parameters; Paramics package (V5) has the following driving behavior related parameters (Zhang, Ma & Dong 2006).

Table 2.1 Global and Local Driving Behavior Parameters

Parameter Category	Parameter Name	Feasible Range	Unit	Description
Global Parameters	Mean Target Headway	0.6~2.4	sec	Mean headway between a vehicle and its following vehicle
	Mean Reaction Time	0.4~1.6	sec	Mean time lag between a change in speed in a leading vehicle and the following vehicle's reaction to this change
	Driver Aggressiveness	0.2~0.8 (mean)	N/A	A distribution that determines how long a headway is accepted by a DVU
	Driver Awareness	0.2~0.8 (mean)	N/A	A distribution that affects the use of a longer headway when a vehicle approaches a lane drop or a ramp
Local Parameters	Link Headway Factor	0.5~2.5	N/A	Adjustment factor for the mean headway on a link
	Link Reaction Factor	0.5~2.5	N/A	Adjustment factor for the mean reaction time on a link
	Ramp Headway Factor	0.5~2.5	N/A	Adjustment factor for the mean headway on a ramp
	Minimum Ramp Time	1~3	sec	Minimum time that a DVU remains on a ramp before considering merging into the freeway
	Ramp Awareness Distance	1~300	Meter	A distance at which a freeway DVU is aware of an approaching ramp
	Sign-posting	1~300	Meter	A distance from the hazard that the most aware vehicles become aware of the hazard ahead.

2.5.2 The Calibration Procedure

The calibration procedure has been describe in detail in Chapters 2 and 3 in our Year-1 report (Zhang, Ma & Dong 2006) and is not be repeated here.

2.5.3 Calibration target: Link Capacities

Both the flow profile (FP) and fundamental diagram (FD) at a specific location have been used as calibration targets (Chapter 2, Part II, Zhang, Ma & Dong 2006). However, most networks may not have well-placed detector stations suitable for the FP method, and finding such a place appropriate for global parameter calibration is even harder. Moreover, since the major goal of driving behavior model calibration is to ensure that the local capacity is faithfully reproduced, a fundamental diagram (FD) approach is used in our subsequent calibration effort.

In the FD approach, one tries to match both the shape and the flow capacity of the observed fundamental diagram of a particular location. If the car-following model is able to reflect reality, field observed capacities as well as critical densities/occupancies of those sections should be closely replicated. As capacity and critical density/occupancy are not influenced by traffic volumes, an accurate O-D demand matrix is not necessary. Rather, we use an artificial demand matrix to produce the whole range of traffic conditions in the simulation, so the shape of the fundamental diagram can be drawn (see Figure 2.5). Paramics provides a demand factor, which globally adjusts volume between each O-D pair by a certain percentage ranging from 0% to 200%. This factor is used to create the demand fluctuations for generating the shape of fundamental diagram under different parameter values. Then the maximum flow rate and the corresponding critical density/occupancy are estimated and compared with their counterparts obtained from field data. Readers can get the implementation details from Chapter 2 of Zhang, Ma & Dong (2006) and the user guidelines in the Appendix.

We use the using the following fitness function to measure the closeness between the simulated capacities and field observed ones:

$$F = \sum_{i=1}^{M} [GEH(Cap_i) + A \times GEH(Occ_i)] \tag{2.9}$$

Where:

M: number of data collection locations

Cap_i: capacity of all general purpose lanes in one direction on which the data collection location i is located;

Occ_i: critical occupancy of a link on which the data collection location i is located.

A: a weighting factor; in general, the GEH values of occupancy are found to be one magnitude lower than those of capacity, the value of A is chosen to be 10 in this work;

and GEH is a statistics by the British engineers (Highway Agency 1996) that reads:

$$GEH = \sqrt{\frac{(V_p - V_m)^2}{(V_p + V_m)/2}} \tag{2.10}$$

where V_p = value predicted by the model and V_m = value measured in the field. Note that a perfect match will result in a zero of the GEH value.

2.5.4 Application to a northern California network

The same network of SR-99 and the settings as reported in (Chapter 2, Zhang, Ma & Dong 2006) are used to compare different algorithms. The site for global calibration is selected based on those guidelines discussed in that report: a straight road section with no drastic changes in geometric features. The network is examined and a section between Florin Road and Mack Road meets the requirements. This section is 2-mile long with two general purpose lanes and one HOV lane. A sub-network is then constructed for this section.

The search space for global parameters calibration includes four dimensions: mean target headway (MTH), mean reaction time (MRT), driver aggressiveness (AGGR) and driver awareness (AWAR) (Table 2.1). Through an ordinary division of each dimension, e.g., a resolution of 0.02, the total number of feasible solutions can easily reach near 5 million. While enumerating all feasible solutions and then selecting the best set becomes impractical using the trial-and-error IA method, genetic algorithm (GA) and SPSA algorithm can generally obtain an optimal solution in much fewer number of iterations. For example, the genetic algorithm took only 600 simulation runs (population 30 times generation 20) to converge to a local optimal solution. Naturally one would wonder how good this solution is. Thus a trial-and-error IA process that searches exhaustively a reduced solution space (using a coarser division) is conducted to benchmark the calibration results.

The first exhaustive search keeps aggressiveness and awareness unchanged, and enumerates MTH from 0.6 through 2.1 and MRT from 0.6 through 1.8, with the increment of 0.02 for each parameter. The results of IA exhaustive search are shown in Figure 2.1 and Figure 2.2.

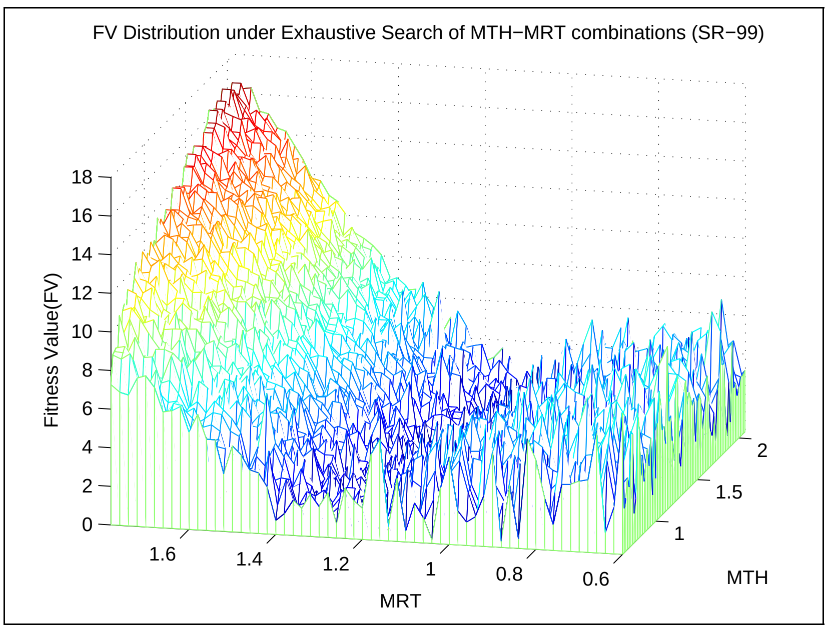


Figure 2.1 Mesh Figure from IA Exhaustive Search (Network SR-99) (with Fixed AGGR and AWAR)

Figure 2.1 visualizes the changes of the fitness value (FV) under various MTH-MRT combinations. First one can notice numerous local optima marked by small “valleys”, denoting the lowest FV within the close vicinity of the local minimizers. Even though these local optimal could be caused by the stochasticity of micro simulation rather than the changes of MTH-MRT pairs, this feature adds to the difficulty of seeking the global optima. Second, one can also notice that a certain range of MRT values produced a similar level of FVs, marked by a big “valley”. In the contour plot of Figure 2.2, correspondingly, a downward-bending band of low FVs is clearly shown, and the global optimal MTH-MRT combinations falls in the center of the band (the intersection of the two white dashed lines). These two figures indicate that in this network context, the simulation performance is more sensitive to the changes of MRT than those of MTH; and similar link capacities can be obtained from a certain range of the combinations.

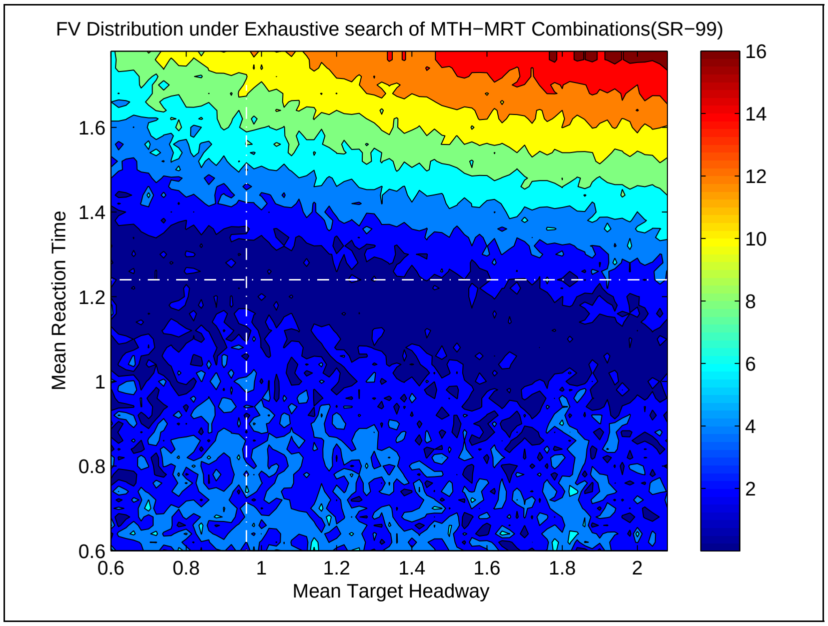


Figure 2.2Contour Plot (MTH-MRT) from IA Exhaustive Search (SR-99) (with Fixed AGGR and AWAR)

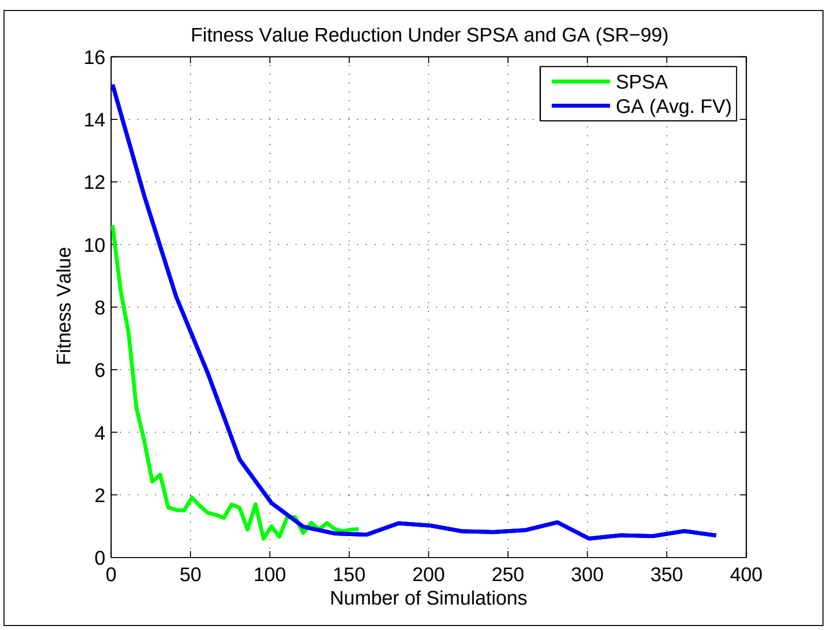


Figure 2.3 GA and SPSA based global parameter calibration

Both GA and SPSA are run on the same reduced search space. Their FV convergence patterns are shown in Figure 2.3 and the results are summarized in Table 2.2. Generally, both GA and SPSA can reach stable solutions that are very close to the global optimum obtained from the exhaustive enumeration. However, the fitness values in the SPSA calibration process show a quick drop during the first few dozens of iterations. Its FV drops to an acceptable value (lower than 2) in less than 50 simulation runs. Because the overwhelming proportion of computation time is spent on simulation runs, here the SPSA calibration algorithm can obtain an acceptable solution in significantly less time (Table 2.2). But GA manages to reach a better solution after many more iterations.

Table 2.2 Numbers of Performance Evaluations under Various Algorithms at Convergence

Algorithm	Total Number of FV Evaluations	Optimal Solution			CPU Time Taken (hour)
		Fitness Value (FV)	Mean Target Headway (second)	Mean Reaction Time (second)	
IA	2,400	0.15	0.96	1.24	12.30
GA	600	0.61	0.96	1.25	3.1
SPSA	150	0.70	0.87	1.27	0.9

2.5.5 Calibration of Local Driving Behavior Parameters

Based on the global parameters calibrated above, local driving behavior model parameters are calibrated subsequently. In the SR-99 network, the southbound section between Fruitridge Road and Mack Road, including the Florin Road / SR-99 interchange, has three general purpose (GP) lanes to the north of the interchange but only two GP lanes to the south. The lane drop section, two on-ramps and two off-ramps, have frequent merging and weaving maneuvers. This site has also been calibrated earlier (see Chapters 3, 4 of part II in this report series, Zhang, Ma & Dong 2006).

2.5.6 Calibration Results of the Local Parameters

One bottleneck link (lane drop), two onramps in the sub-network generate altogether 15 local parameters to be calibrated, and the calibration target is the truncated capacity of the link upstream of the bottleneck. The target link has one PeMS detector group (VDS 312513) that provides flow and occupancy data in five-minute intervals.

The IA trial-and-error method becomes impractical in this context because of the large parameter space. Only GA and SPSA calibration algorithms are implemented and compared, shown in Figure 2.5 and Table 2.3. In Figure 5, the FVs from SPSA again show a quick drop during the first few iterations. However, it becomes quite oscillatory during the remaining iterations. A close examination of the successive solutions indicate that only very minor changes occur from one iteration to the next, and generally it falls within the close region around the parameters' values in Table 2.3. It implies that the capacity of the target link is very sensitive to the changes of local parameter values.

As to the GA method, a smoother convergence pattern is observed in terms of average FVs (Avg. Fitness in Figure 2.4). The ranges between the maximum and minimum fitness values within successive GA generations become smaller during the process, which implies that the GA method manages to reach a better solution.

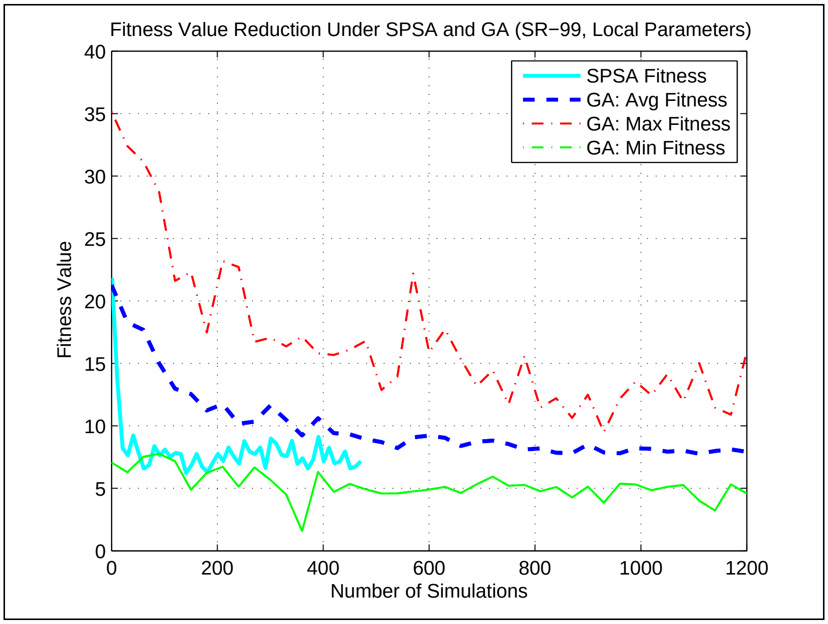


Figure 2.4 Local Parameter Calibration Convergence Diagram

Table 2.3 shows the optimized local parameters. The fitness value associated with the set of parameters from GA is 1.59, smaller than that from SPSA (4.94). According to certain application guidelines for micro simulation (Dowling, Holland & Huang 2002), both are considered acceptable. But one can notice that these two sets of parameter values are not close. The fundamental diagrams under the set of best-matching parameters and field observations are plotted in Figure 2.5, where the field observed link capacity and critical occupancy are estimated to be 5,020 veh/hr and 0.10. The counterparts from SPSA and GA are (4,715 0.083) and (4,972 0.13), respectively. The GA results are slightly better and thus selected in the final application; but the GA algorithm took much longer to produce those results than the SPSA algorithm. Unlike the calibration of global parameters (Figures 2.1, 2.2 and Table 2.2), different search algorithms yielded quite different parameter values in some parameters yet these different parameter sets produced similar simulation results. This is a clear indication that the parameter space is quite complex and many local minima exist in this space. As reported in the pat II report (Chapter 3, Zhang, Ma & Dong 2006), the calibrated local parameters can be verified to provide better simulation results against different data sets. The verification implies that it is reliable to use such heuristic methods in micro simulation. It should be pointed out that the fit to the observed flow-density plot produced by the GA search algorithm is better than that produced by the SPSA, although the GEH values produced by the two algorithms are similar, which indicates that GEH may not fully capture the variances in traffic flow characteristics.

Table 2.3 Best Optimized Local Parameters

	Bottleneck (mainline north of the interchange)		On-Ramp from Fruitridge Road WB		On-Ramp from Fruitridge Road EB	
	GA	SPSA	GA	SPSA	GA	SPSA
Link Headway Factor	0.63	0.93	1.96	0.91	0.53	1.18
Link Reaction Factor	1.58	0.99	0.93	1.13	0.80	1.06
Sign-posting (meter)	3043	801	924	802	759	805
Ramp Headway Factor	---	---	1.13	1.13	1.08	0.99

Minimum Ramp Time (sec)	---	---	2.94	1.44	1.28	1.64
Ramp Awareness Distance (meter)	---	---	212	484	8.72	245

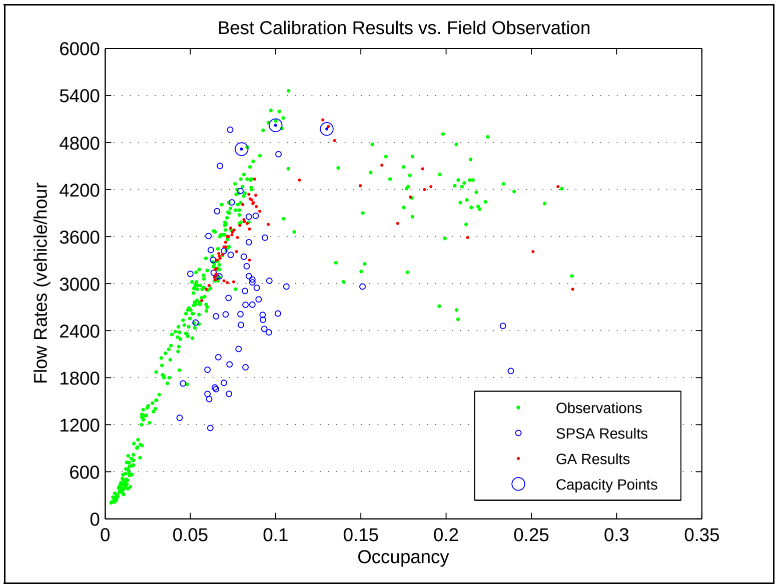


Figure 2.5 Best Simulation Results vs. Field Observation

2.6 Application Guidelines of SPSA in Micro Simulation Calibration

SPSA is a method easy to implement, and generally shows good performance, especially when there are observation errors in data. Similar to other heuristic methods, proper selection of appropriate algorithmic parameters (including ***a***, ***c***, ***A***, α and γ in SPSA) is of crucial importance to its performance. For example, two different initial values of ***a*** leads to quite different convergence performances when searching the best MTH-MRT pair (Figure 2.6), although they both start with the same guess (MTH = MRH = 1.60 seconds). However, more experimentation indicates that choosing too large $\{a_k\}$ and $\{c_k\}$, i.e., larger ***a*** and ***c*** that aims at an even faster convergence could lead to drastic changes in $\{\hat{\theta}_k\}$ and the calibration process may not even converge. To assist the further application of this method, therefore, some general guidelines from (Spall 1998) and based on our own experiences are summarized here.

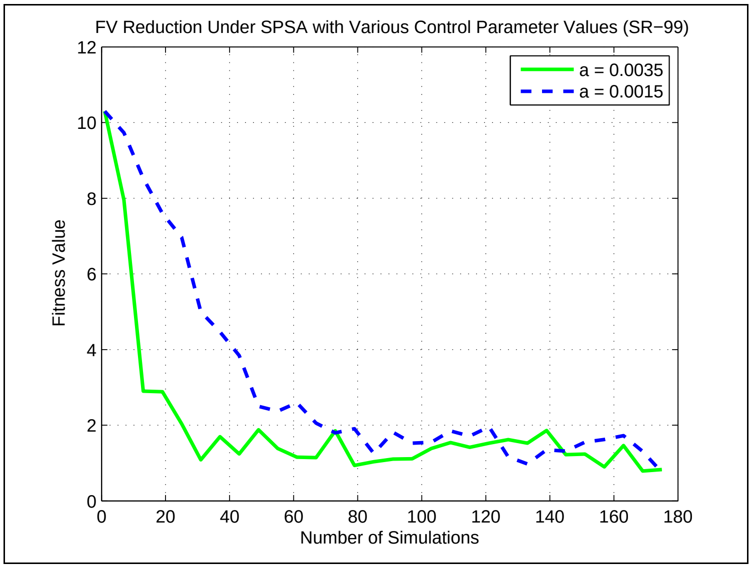


Figure 2.6 Experimentation with SPSA Algorithmic Parameters (SR-99)

For the value of ***c***, Spall (1998) recommended using the standard deviation of the performance measure (fitness function value) from a few runs under the initial value θ_0, so that the initial perturbation steps do not go excessively large. For $\{a_k\}$, a stabilization parameter A is introduced as $a_k = \frac{a}{(A+k+1)^{\alpha}}$, and 10% (or less) of the predefined maximum number of iterations proves practically effective for the value of *A*. And the value of *a* is chosen such that $\frac{a}{(A+1)^{\alpha}}$ (i.e., the initial step of change) times the magnitude of $\hat{g}_0(\hat{\theta}_0)$ would be equal to the smallest change in the magnitude of θ_0 during the early iterations. We recommend that the changes of $\{\hat{\theta}_k\}$ during the first few iterations not exceeding 2-4%.

The other issue concerns the selection of stop criteria. A commonly used criterion is the percentage change of the objective function value below a certain threshold, e.g., 2%. It performs well for problems with no observation errors ($\varepsilon_k^{(\pm)}$); but in the case of micro simulation calibration where performance measures such as the FV defined in equation (9) can easily have a variation larger than 2% even under the same set of parameters. Therefore, a predefined maximum number of iterations and an acceptance level of objective function values combined are better indicators to decide when to stop the calibration process.

2.7 A Brief Summary

The calibration of a micro simulation is a complex problem that often defies conventional optimization methods, and calls for reliable and more efficient heuristic optimization methods. Some heuristic methods, such as the GA method introduced in Report II (Zhang, Ma & Dong 2006), usually take considerable computational time to obtain satisfactory calibration results. In this chapter, we introduced another heuristic optimization method, simultaneous perturbation stochastic approximation (SPSA), to calibrate driving behavior model parameters. Compared to other heuristic methods such as GA, this method can generally obtain an acceptable set of parameters in much less time.

Although SPSA speeds up the calibration process considerably compared with the GA method, our experiments indicate that GA is more capable of reaching finer solutions. We thus recommend that when the number of parameters to be calibrated is not large, and the network concerned is small, GA should be used. When either the network or the number of parameters to be calibrated is large, SPSA should be used. This is a trade-off between solution accuracy and solution speed.

CHAPTER 3 TRAFFIC DEMAND ESTIMATION AND REFINEMENT

3.1 Preparation of Traffic Demand for Paramics: the O-D Estimator

Setting up the appropriate traffic demand inputs for any micro simulation is vital to its success, but preparing the traffic demand table can be a very difficult task. In general, both the temporal and spatial scope of micro simulation has been considerably expanded to go beyond a freeway section (Cheu et al. 1998, Lee, Yang& Chandrasekar, 2001) or an isolated location such as an interchange or an intersection (ISAC 1999) to include corridor networks (Gomez, May & Horowitz 2004, Gardes & May 1993, Lee & Kim 2004) and city or region wide transportation networks over 520 km^2 (Smith & Sadek 2006, Bridges 2003). Preparing quality input data for these networks, especially the O-D demand, takes a large portion of the project time. Occasionally this leads to unmet project goals because time was run out when it took an unexpected amount of time to prepare good O-D demand data (Bacon, Windover & May 1995).

A remedy to this problem is to take advantage of readily available data sources from other models (Dowling, Holland & Huang 2002). For instance, regional planning models could provide an initial O-D demand input. As a matter of fact, many early micro simulation applications applied O-D demand from planning models as the input. Because planning models do not capture the dynamic features of traffic flow, particularly peak spreading and queuing, such O-D inputs did not provide the micro simulation models with good structural information about the time-dependent O-D demands, therefore simulations with such O-D inputs often generated unacceptable errors compared to the observed macro measurements such as link counts (Gardes & Bloomberg 2003). Some micro simulation packages, such as Paramics (Quadstone Ltd. 2004a), developed their own demand estimation modules to assist the analysts obtaining O-D inputs. It was reported that O-D matrices obtained from these modules were generally better than that from planning models (Gardes & Bloomberg 2003).

The O-D Estimator in Paramics can take as much, or as little data input as the user can provide to generate and optimize an O-D matrix. There are four categories of input data and a subset of these categories will be enough to start an O-D estimation process in the O-D Estimator:

- Pattern O-D (or seed O-D) which serves as a starting reference and target demand;
- Link counts, hourly (or finer) link volume observed in the middle of road sections;
- Turning counts, hourly (or finer) turning volume observed at signalized intersections;
- Cordon counts, hourly (or finer) traffic volume that crosses a given screen line.

Paramics' O-D Estimator also provides the level of confidence, or *weight*, associated with each data set, e.g., seed matrix or each entry of the traffic counts. The user can adjust the default weight values to make them commensurate with the data reliability. For instance, a low level of confidence is more appropriate for the traffic counts if they were assembled from some outdated sources.

3.2 Guidelines on O-D Estimation Using Paramics' O-D Estimator

To note that demand estimation process may not be a once-for-all task; sometime it may be done repeatedly or iteratively with other calibration steps. This is particularly common when the network or dataset are updated, e.g., rearrangement of zoning structures, more available data and identification and elimination of severe coding errors during the calibration. The demand estimation process could actually start when the initial network has been coded and the coding has been checked against possible errors (Zhang, Ma&Dong 2006). This is especially useful when traffic counts (e.g., link counts, turn counts or cordon line counts) are the only available traffic data and an initial demand pattern must be provided. In this section, some application guidelines are summarized from past demand estimation experiences with several networks (Gardes & Bloomberg 2003, Gardes et al. 2001).

3.2.1 Data Checking

A common O-D demand estimation practice is to use traffic counts assembled from various sources in different times. For instance, traffic counts can be obtained online in real-time as fine as every five minutes (or even finer at 30-second level) for freeway sections covered by loop detectors, while the turn counts for city streets may only be updated periodically in a few years. Even though both data sets are valuable, they may be inconsistent. Past experiences tell that the O-D Estimator could get trapped in a loop and fail to converge to a stable O-D matrix if fed with inconsistent data. Therefore, the first step of using the O-D estimator is to check for data consistency.

Paramics' O-D Estimator provides a "Data Validation" (menu "Tools->Validate Data") module to fulfill this task. When the network is loaded in the O-D Estimator, the module will check whether the link counts or turn counts are consistent with the neighboring data entries. For instance, the link flow in Figure 3.1 in theory should equal the sum of turn counts from approaches 1, 2 and 3 and any error in this relation will be reported in the Data Validation window. When inconsistency occurs, usually the data entries with lower weights are corrected.

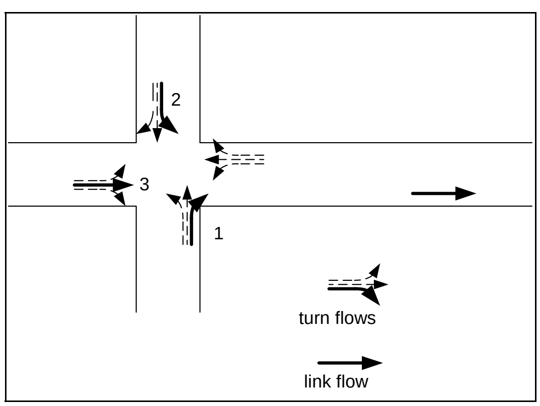


Figure 3.1 Checking Traffic Counts Data Consistency

It should be noted that two types of traffic demand input exist, i.e., O-D matrix or intersection turn counts. Simulations packages like Paramics, AIMSUN, and CORSIM accept O-D demand matrices, while SimTraffic and others accept turn counts as O-D demand. Some packages, such as VISSIM accept both types of traffic demand input. The data consistency issue does not occur when turn counts are used as demand inputs, for traffic can virtually be generated or absorbed on each link. It shows up, however, when O-D matrices are the required format for the demand input, and one must check the consistency of the input data, because flow conservation must be maintained at both the inter-zonal and link levels.

Another commonly neglected problem in preparing the input for O-D estimation is the locations from which the data were collected. In Paramics, a sink link is used to receive traffic coming from other zones, and its traffic statistics, such as traffic counts, are treated differently than a common link (e.g., the link count given by Paramics' Analyzer is zero for any sink link). Accordingly, sink links should not be used to observe traffic flow.

3.2.2 Zone Structuring

Even though traffic demand is given in the format of zone-to-zone O-D table, a Paramics simulation actually generates and absorbs traffic on connector links associated with each zone. When there are more than one link associated with a zone, Paramics simulation will randomly generate vehicles and put them onto each outgoing link of the zone. This vehicle releasing mechanism, though capturing the stochastic nature of real traffic, could lead to various path choices between certain O-D pairs and thus create ambiguity in the O-D estimation process. It is then recommended in O-D Estimator that such ambiguity should be eliminated by keeping only one single generating and receiving link section for each zone.

When a zone is too large to accommodate the traffic with only one incoming and outgoing link, it is suggested to break up the large zone into smaller ones, each of which holds only one incoming and outgoing link pair. This is especially true when the zoning structure is "borrowed" from regional planning models, where a single TAZ could easily cover a large community block with exit road facing different directions. Breaking the large TAZs into smaller zones in micro simulation is suitable in this context (Gardes and Bloomberg 2003), and the pattern demand for the original large TAZ from the planning model should also be evenly divided among these smaller zones in the micro simulation model.

This, of course, has its own drawbacks: 1) the number of zones could increase dramatically for large networks and this can considerably increase the computational time of traffic assignment in the simulation; 2) limiting just one incoming and outgoing link from a zone may not work for some situations. For example, a large parking lot may have various exits connecting to different streets. In this situation, "*car parks*" must be used instead of ordinary zones. Nevertheless, a clearer zoning structure can generally lead to faster convergence and easier control of the O-D estimation process.

3.3 O-D Estimation Settings and Process

3.3.1 Basic Settings for the Estimation Process

Two factors are critical when setting up the O-D estimation process in the O-D Estimator, one is the assignment method, and the other is the length of the simulation time. Determining the assignment methods, which include all-or-nothing, stochastic and dynamic feedback, or their combinations, is up to the analyst's judgment based on his/her own knowledge of the simulated traffic. Meanwhile, following the calibration procedure, the familiarity and perturbation factor (when stochastic and dynamic feedback assignment methods are used) should be calibrated using the departure time and route choice (D-R) calibration module. At the same time, the simulation time length for each O-D estimation iteration is recommended to be no less than the longest trip occurring within the network.

Other key parameters that affect calibration performance include the minimum and maximum trip demand values that can be adjusted dynamically while the estimation process is running, flow intensity and the “revert to best” parameter. The minimum and maximum trip demand values define the absolute bounds for the variation of the demand between each O-D pair (Quadstone Ltd. 2004a). The minimum value plays a critical role by allowing for manual adjustment of the O-D pairs with very low expected demand. The flow intensity refers to the percentage of the overall demand that is used in the estimation process; the “revert to best” percentage indicates the fraction of the best O-D matrix to be used in the next iteration.

3.3.2 The Estimation Process

There are two basic strategies to run the O-D estimation in Paramics’s O-D Estimator: starting from low demand and increasing the demand upward (the upward strategy), and the other is just the opposite (the downward strategy). The upward strategy can avoid traffic gridlock in the early stage of the estimation process and is therefore usually recommended. This stragety is controlled by dynamically setting the “flow intensity” factor during the estimation process. The higher the flow intensity, the more traffic will be released onto the network during estimation. Besides setting the flow intensity upward gradually, two other ways are available to ensure the success of the upward strategy. The first is through adjusting the minimum trip parameter. This value cannot be set very high if a network has a large number of zones, for too many fake traffic will be otherwise released artificially onto the network and possibly distort the estimation of the true traffic demand. The second way is through adjusting the scale factor. This is one value that the O-D Estimator prompts when opening the network, which is usually larger than one. If the value is accepted, this factor will scale the pattern O-D table with that value. In this case, the O-D Estimator could load too much traffic onto the network and possibly cause traffic gridlocks.

The error between the estimated traffic counts with a given O-D table and the traffic counts from field measurements is measured in either average GEH or the sum of least squares. Usually, the average GEH is applied and a value of lower than 5 is generally considered acceptable.

3.4 Inefficiencies in Paramics O-D Estimator

Although being a powerful and easy to use calibration tool, Paramics’ O-D Estimator inherits a significant deficiency of microscopic simulation: it usually takes considerable time to finish multiple estimation runs because the O-D Estimator must call the simulation engine to evaluate the performance of the estimated O-D tables. This issue becomes more pronounced when the network gets larger and consequently route choice sets between O-D pairs grow exponentially. Furthermore, complex networks (e.g., a general corridor network) usually apply multiple periods to accommodate the demand variations during the whole simulation horizon. One O-D table must be estimated for each period, which requires more demand estimation and analysis time.

The modeling and estimation time is further elongated when using the recommended coalition strategy of scaling up the demand gradually: increased flow intensity implies that more traffic needs to be simulated and when this happens the estimation process slows down dramatically.

Another important observation is that a poor seed matrix fed into the O-D Estimator inevitably leads to zigzagging in the estimation process, and consequently considerably slows down the convergence or at times even leads to no convergence.

3.5 A Methodology of Accelerating Traffic Demand Estimation

As we mentioned in the previous section that a good seed O-D matrix can speed up the convergence of Paramics' O-D estimator. In this section, we propose a methodology that can produce a good seed O-D matrix. The essential idea is to exploit the speed and accuracy of time-dependent macro O-D estimators to obtain a good reference (seed) O-D matrix for a micro simulation. Figure 3.2 shows the three major steps of this method. The first step converts the coded micro simulation network into a simplified macroscopic network, the major task being the translation of the network features into corresponding link capacities. Taking this simplified macro network as well as the observed measurements (link counts) as inputs, the second step develops an extended Logit path flow estimator (LPFE) to come up with an improved pattern O-D matrix. This pattern O-D matrix will be used in Paramics' O-D Estimator as the seed OD pattern. The O-D Estimator then performs its own estimation with this seed O-D in the third step.

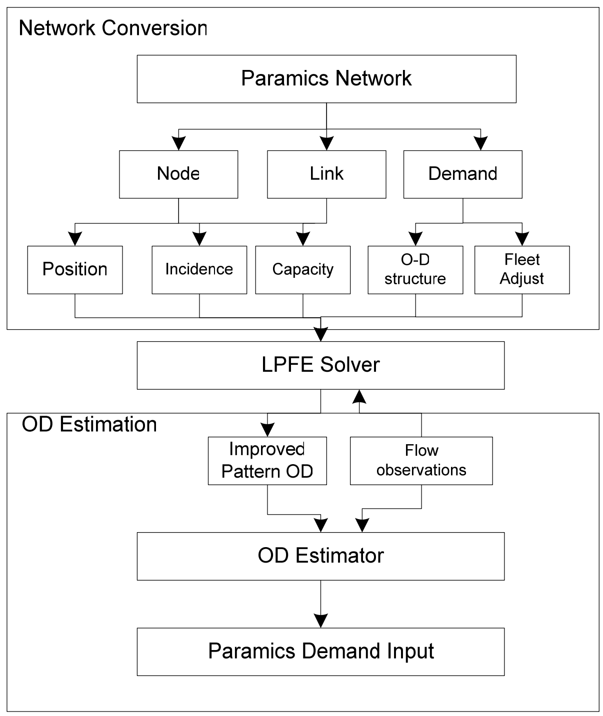


Figure 3.2 An LPFE-based O-D Estimation Enhancement Procedure for Paramics

3.5.1 Network Conversion: Link Capacity Translations

Micro simulation and macro traffic models differ in many ways in their description of traffic flow. Micro simulation, for example, does not have explicit specifications of flow capacity on a road section. Rather, the capacity of a road section is determined driving behavior parameters and the special road features of that section, such as its curvature, and presence of lane drops and ramps. As a matter of fact, calibrating these model parameters to replicate the traffic flow characteristics at these locations is one vital step in building a reliable micro simulation. In contrast, a macro traffic flow model often requires the capacity of a road section as its basic input. For example, LPFE takes a link's capacity as one major factor to calculate the travel cost for that particular link. It is then a fundamental task to translate the detailed network information in a micro simulation into the information required by a macro model (see Figure 3.2). In this process, the link capacity translation is a critical task as it affects the path choice in LPFE solutions.

Three categories of information in a micro simulation are critical in the translation:

- General links and critical sites: typical road sections and sections with special features such as work zones, lane drops, and sharp curvatures;
- Controlled links: road sections adjacent to a signalized intersection or a metered ramp.
- Vehicle fleet: demand adjustment according to vehicle type features.

General links and critical sites

We use two supplemental approaches to retrieve the capacity for a general link, one is the Highway Capacity Manual (Transportation Research Board 2000) capacity adjustment method and the other is the direct interpretation of driving behavior model parameters.

For a basic freeway section with free flow speed (FFS) of 70 mph or higher, HCM 2000 suggests a base capacity value of 2,400 pc/hr/ln; and with FFS of 55-70 mph, 2,250 pc/hr/ln. These basic capacity values will be adjusted from local geometric conditions (number of lanes, lane width, lateral clearance, interchange density, grade and FFS) and demand characteristics (peak hour factor, heavy vehicle percentage and driver population factor). The basic formula for calculating this adjusted capacity is:

$$C_{adj} = C_{base} N \prod_i f_i \qquad (3.1)$$

where C_{adj} is the adjusted capacity in veh/hr/ln; C_{base} is the base capacity and N is the number of lanes; and f_i is the adjustment factor for condition i , which can be found from HCM (2000).

The other approach is to deduce the capacity from flow-density plots based on field data or the outputs from the coded micro simulation. One should exercise caution when using this approach to obtain the flow capacity for a link (ISAC 1999, Gardes et al 2001), for the flow-density plot usually has a lot of scatter and it is not straightforward to decide on an appropriate value for capacity. Doing this in a micro simulation also requires that the driving behavior parameters have been properly calibrated.

At certain locations such as lane drops, sharp geometric changes, weaving sections or work zone sections, the driving behavior model parameters must also be adjusted locally to reflect the special characters of traffic at these locations (Ma, Zhang & Dong 2006). Correspondingly, the capacity values for these links in the translation have to be adjusted. For example, the lane capacity of a construction work zone section used in our example below takes a value of 1,500 pc/hr/ln after the calibration (Lee & Kim 2004).

Controlled links

A controlled link refers to one that has a control device at the end of the link, including a traffic signal or a ramp meter. The capacity treatment for these two types of controlled links is different in our translation.

Two basic signal controllers are common in a network, i.e., pre-timed and vehicle actuated (VA). For any given link *i* that approaches a pre-timed signalized urban intersection and occupies an independent phase, its link capacity is computed as:

$$C_i = S_i \frac{g_i}{c} \qquad (3.2)$$

where C_i is the link capacity and S_i is the saturation flow rate for that particular link; g_i is the green time duration for the phase and c is the cycle length. The saturation flow rate calculation applies a similar adjustment approach as in formula (3.1) according to HCM2000: adjusting an ideal SFR of 1,900 pc/hr/ln with such factors as lane width, heavy vehicle percentages, grade, on street parking, bus blockage, area types and so forth. All these values could also be retrieved easily from the coded micro simulation model. For a controlled link with more than one lane group, the capacity for the particular link will be the sum of the capacity for every lane group on the same link. This intersection capacity utilization concept is generally applied in the traffic signal optimization models.

HCM provides an averaging process called queue accumulative polygon (QAP) to estimate the green time for VA controllers (HCM Chapter 16, Transportation Research Board 2000). However, this QAP process is too complicated to be used for calculating link capacity in the capacity translation. We take advantage of micro simulation to estimate the average green time for a VA controlled link: the overall green split allocated to a particular link is calculated over the entire simulation horizon after a trial-run simulation and taken as the green ratio, and the procedure of dealing with the pre-timed controllers is applied to obtain the link capacity for the subject VA controlled link. Links with other control facilities such as priority control via yield signs or stop signs could be treated similarly.

For metered ramp links, the capacity values just take the (average) ramp metering rate. When adaptive ramp metering algorithms are used, the same trial-run and averaging process as in the above VA controllers could be applied to retrieve the capacity values for these links.

Vehicle fleet and demand data from sketch planning models

In the network conversion process, vehicle fleet information can be used for two purposes: 1) link capacity interpretation for both freeway sections (HV percentage) and urban street sections (HV percentage, bus blockages) as described in the previous sections; 2) the adjustment of the structure of the pattern demand obtained from sketch planning models.

Even though an O-D matrix extracted from a travel forecasting model is too coarse to be directly used in a micro simulation, it could serve as the historical O-D for LPFE, i.e., vector $\bar{\mathbf{q}}_o$ in equation (3.7) in Section 3.5.2. However, as the historical O-D is generally an aggregated measure that does not differentiate vehicle types, it is subject to adjustment according to the vehicle fleet information. For any given O-D trip rate $\bar{q}_{o,i}$ in the historical O-D matrix, the adjusted value becomes:

$$\bar{q}_{o,i} = \bar{q}_{o,i} \times \sum_{j=1}^{J} E_j p_j \qquad (3.3)$$

where p_j is the percentage of a certain vehicle type j in the fleet, and E_j is the passenger car equivalent for type j, and J is the total number of vehicle types defined in the micro simulation network. In this manner, the pattern O-D matrix with multiple vehicle types is transformed into one with only one vehicle type, which can then be fed into LPFE.

3.5.2 Estimation of a Seed Matrix from Logit Path Flow Estimator (LPFE)

Bell's logit path flow estimator (LPFE) (Bell & Shield 1995) estimates path flow rates between an O-D pair, thus its trip rates after summation over path flow rates. It incorporates the logit route choice model while maintaining a one-level optimization structure. LPFE has been used in a number of studies as a tool to estimate O-D flows as well as path travel times (Bell & Shield 1995, Bell & Grosso 1998) and was shown to yield reasonable O-D estimates.

In our study we extended LPFE in the following way:

$$\min \sum_{a \in A_u} \int_0^{x_a} t_a(w)dw + \sum_{a \in A_o} \bar{t}_a x_a + \frac{1}{\theta}\langle \mathbf{f}, \ln \mathbf{f} - \mathbf{I} \rangle \quad (3.4)$$

subject to:

$$\Delta_u \mathbf{f} \le C_u \quad (3.5)$$

$$\langle \mathbf{x}_o, \mathbf{I}_o - \delta_o \rangle \le \mathbf{\Delta}_o \mathbf{f} \le \langle \mathbf{x}_o, \mathbf{I}_o + \gamma_o \rangle \quad (3.6)$$

$$\langle \overline{\mathbf{q}}_o, \mathbf{I}_o - \phi_o \rangle \le \mathbf{M}_o \mathbf{f} \le \langle \overline{\mathbf{q}}_o, \mathbf{I}_o + \eta_o \rangle \quad (3.7)$$

where
x_a is the traffic volume on link a;
$t_a(.)$ is the travel time on link a, which is a non-decreasing function of x_a;
$\bar{t}_a$ is the observed travel time on link a;
$\mathbf{f}$ is a vector of path flows;
$\mathbf{x_o}$ is a vector of measured link traffic volumes;
$\overline{\mathbf{q}}_o$ is a vector of historical O-D flows;
C_u is a vector of link capacities for all unobserved links;
Δ_u is the portion of the path-link incidence matrix corresponding to unmeasured links;
$\mathbf{\Delta}_o$ is the portion of the path-link incidence matrix corresponding to measured links;
$\mathbf{M}_o$ is the path-OD incidence matrix;
A_u is the set of unmeasured link;
A_o is the set of measured links.

In this study, the BPR function is used for $t_a()$. There exist a number of algorithms for solving a nonlinear optimization problem like the extended LPFE (e.g., the Frank-Wolfe algorithm). However, the complex constraint structure involved in the problem makes it unappealing to use those classical procedures. The iterative balancing algorithm (IBA) offers a competitive alternative by directly exploiting the Kuhn-Tucker conditions. Zhang, Nie & Shen (2005) described a solution framework for the extended LPFE, which consists of three major components: a column generation scheme to avoid path enumeration, a descent direction algorithm, and a revised IBA algorithm for solving the restricted sub-problems (i.e., with a given path set and fixed link travel costs). Readers are referred to Zhang, Nie & Shen (2005) for details of this solution framework.

Once the seed matrix is generated from LPFE, the analyst can apply this seed O-D matrix to Paramics' O-D Estimator to further refine the O-D demand estimates. One should note that LPFE does not replace the estimation module within Paramics. Instead, the results from LPFE serve as an improved seed O-D trip table for the O-D Estimator in Paramics. Nevertheless, when one does not have Paramics' O-D estimator at hand or one wants a rough O-D table quickly, one can use the O-D trip table produced by LPFE in Paramics simulation. The latter was done in the calibration of SR-41 in our case study, to be described in Chapter 5 of this report.

3.5.3 A Demonstration of the Procedure with a Southern California Network

A medium-size network is used to investigate the effectiveness of the proposed procedure. The network is a 12-mile long section in the Interstate 15 freeway corridor in San Bernadino County, near the City of Devore in southern California (Figure 3.3). The simulation was established to analyze the traffic management plans around a construction work zone, such as mobile barrier positioning and truck restriction and rerouting. The network consists of 1,600 plus links connecting 32 zones. The simulation study period is three hours. The driving behavior model parameters for the I-15 network and some specific locations have been adjusted based on previous studies (Gardes & Bloomberg 2003). The Southern California Association of Governments (SCAG) developed and maintained a travel demand forecasting model for the entire LA basin (GIS/Trans 1996). The traffic analysis zone structure used in the study was extracted from the SCAG model, and the O-D matrix obtained from the SCAG model was adjusted based on the method in (Formula 3.3).

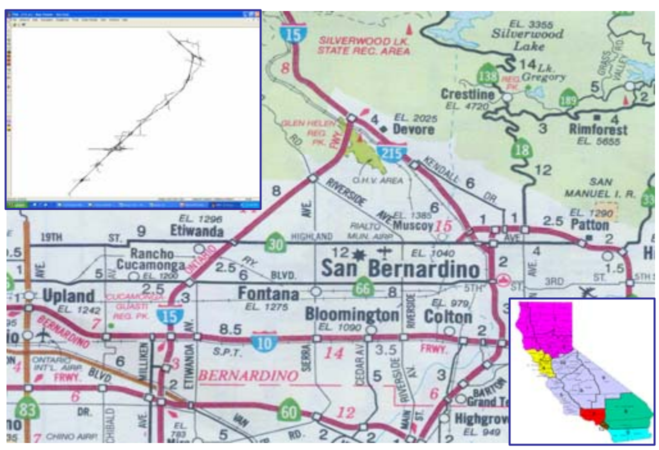


Figure 3.3 The Layout of the Interstate 15 Network and the Paramics Network

Based on the procedure in Section 3.5.1, the network is converted into a macro network that can be solved by the LPFE module. Then LPFE was run and an improved O-D trip table was generated and fed into Paramics' O-D Estimator as a seed O-D matrix. Both scenarios of O-D estimation with and without the improved seed O-D trip table were run; and their convergence performances were recorded and compared in Figure 3.4.

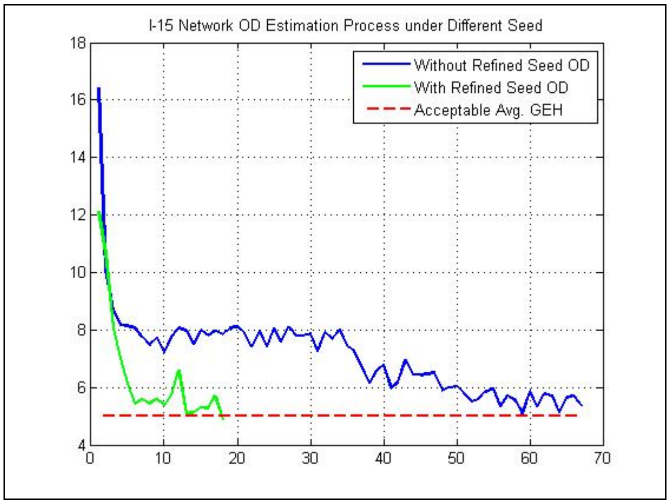


Figure 3.4 Paramics O-D Estimation Convergence Performance under Different Pattern O-Ds for the I-15 Network

As recommended by the calibration guidelines (Dowling, Holland & Huang 2002), an average GEH below 5 over link count observations is acceptable for a micro simulation. It can be seen from Figure 3.4 that with the improved seed O-D matrix, Paramics' O-D estimation process went through a rapid drop to reach a GEH below the acceptance level at the 18th iteration (Avg. GEH = 4.86). With the seed O-D matrix from the SCAG model, the estimation process took a longer and oscillatory path to reach an average GEH of 4.97 at the 78th iteration.

A good seed O-D matrix also helps save significant computational resources. It took about one minute to complete a single estimation run on an up-to-date PC (2G CPU, 1G RAM) and the estimation process with the seed O-D provided by a planning model took about 1.5 hours before it was terminated. In contrast, LPFE produced the seed O-D matrix in less than a second; and Paramics' O-D estimator with this seed O-D matrix reached the target GEH (<=5) in less than 20 minutes. It is expected the computational time savings will be even more significant when the network becomes larger, where one estimation run takes longer to complete and more estimation runs are needed to account for the more complex traffic interactions in a large network.

To investigate the reliability of the estimated demand under both cases (i.e., a LPFE seed O-D vs a planning seed O-D), the O-D matrix obtained from each case was applied to Paramics' Modeller to obtain link flow rates at the locations where measurements were taken in the field. Freeway mainline, on/off ramps and arterial street counts are categorized and compared in Figure 3.5. As indicated by the average GEH, the simulated link counts match closely observed link counts with both O-D matrices. This indicates that the three-step procedure through LPFE can considerably speed up the O-D estimation in Paramics without sacrificing the accuracy of the estimated O-D demand.

Figure 3.5 I-15 Simulated Link Counts under Different Estimation Process vs. Field Observation

CHAPTER 4 HANDLING TRAFFIC SIGNALS IN CALIBRATION

4.1 Introduction

Generally, modes of traffic signal operation can be divided into three primary categories: pre-timed, actuated and adaptive. In the field, the widely used signal mode in the US is actuated signal control with complex NEMA logic or type-170 logic.

Paramics comes with a fixed-time signal control module and some programming capabilities that can be used to emulate various actuated (and adaptive) signal control logic. There are two programming methods to emulate an actuated (or adaptive) signal in Paramics:

(1) Using plan / phase language, which is a kind of script language that can be used to simulate some simple actuated signal control logic, such as those for transit operation.

(2) Using Application Program Interface (API) provided by Paramics to develop plugins that can work together with Paramics.

Between them, the first option has limitations to emulate the complex actuated signal control schemes used in the real world and replicate these schemes to multiple signalized intersections.

This section describes our efforts to enhance Paramics' functionality in actuated signal control and explains what is needed for arterial calibration. The assumption of this chapter is that user/reader already has some knowledge of actuated signal control.

4.2 Actuated Signal Control Enhancement

4.2.1 Background

The commercial Paramics software can't directly emulate all signal operations deployed in the field but Paramics' functionalities can be enhanced and complemented via API programming. As a result, a plugin program needs to be developed and the plugin should be able to emulate major functions of actuated signal control. Considering that there are many different traffic controllers and their signal control logic may be similar but not exactly the same, the plugin program should be able to take parameters from different types of traffic controllers and emulate related functions.

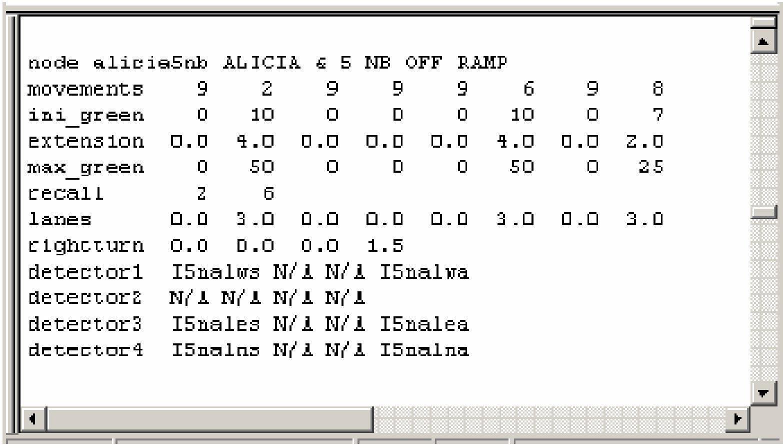
Based on signal operation practices in District 12 (Orange County), District 4 (Bay Area), and District 6 (Fresno), it is found that Caltrans is controlling signals close to freeways and most Caltrans signals are operated using 170 Controller with C8 program (some with C4 program). For example, a typical freeway interchange has two signals controlled by Caltrans, as shown as red dots in Figure 4.1. According to timing plans from Caltrans, Caltrans signals are usually operated under actuated signal coordination during day time and under free mode of actuated signal control during night time. For some un-congested locations, signals may be always operated under free mode.

Figure 4.1 Two Caltrans controlled signals at a freeway interchange

In addition, many signals controlled by cities are also operated under actuated signal control. The popular traffic signal controllers/programs include (1) Bi-Tran 233; (2) SEPAC / EPAC from Simens/Eagle controllers; (3) Multisonics 820A; (4) Econolite. Based on our understanding, C8 and Bi-Tran are 170-type controllers and others are NEMA type controllers.

The signal control module should be able to emulate major functions provided by Caltrans 170 Controller's C4/C8 program. In addition, the developed plugin should be compatible with Bi-Trans and SEPAC (previously called EPAC) programs. It should be able to take signal data of these controllers and emulate associated functions.

A plugin was developed to model actuated signal control within Paramics in early 2001 based on version 3.0 of Paramics. However, this plugin only implemented basic functions of actuated signal control due to the limited resources and knowledge regarding how actuated signals are operated in the field. Its input data includes minimum green, extension, maximum green, recall, phasing, geometry, and detector information. Its functions can be described as follows. If a phase has vehicle calls, the phase will be given with a certain minimum green. If there are vehicles passing the phase's extension detector after minimum green ends, green will be extended with the maximum of maximum green specified. For those recall phases, they won't be bypassed but non-recall phases will be bypassed if there is no vehicle call.



```
node alicia5nb ALICIA & 5 NB OFF RAMP
movements     9    2    9    9    9    6    9    8
ini_green     0   10    0    0    0   10    0    7
extension   0.0  4.0  0.0  0.0  0.0  4.0  0.0  2.0
max_green     0   50    0    0    0   50    0   25
recall        2    6
lanes       0.0  3.0  0.0  0.0  0.0  3.0  0.0  3.0
rightturn   0.0  0.0  0.0  1.5
detector1   I5nalws N/A N/A I5nalwa
detector2   N/A N/A N/A N/A
detector3   I5nales N/A N/A I5nalea
detector4   I5nalns N/A N/A I5nalna
```

Figure 4.2 Input file of the previous version of the plugin

Compared to the previous plugin, the following new functions have been added to the new plugin:

(1) Volume-density control

(2) Yellow and red time handling

(3) Max green handling

(4) Recall modes

(5) Actuated signal coordination

(6) Time-of-day (TOD) control

(7) Output actual green split per cycle

These functions will be then explained in order to connect real-world signal timing charts with the new plugin.

4.2.2 Volume-density control

Volume-density control is recommended when approach speeds are higher than 35 mph. It needs to work with detectors set back considerable distances from the stopline. Volume-density control has two features: (1) variable initial timing; (2) gap reduction.

The developed plugin accepts the following parameters for volume-density control.

(1) minimum green (or minimum initial)

(2) added initial (or added seconds per actuation)

(3) maximum initial

(4) minimum gap (or minimum extension)

(5) maximum gap

(6) passage (or extension)

(7) reduce by (seconds of gap reduced)

(8) reduce every (per seconds of interval)

(9) time to reduce

(10) time before reduce

The first three parameters are used for variable initial timing and the rest are used for gap reduction. For variable initial timing, Caltrans C8 program and Bi-Tran use “minimum green” and “added initial”. Both of them do not have “maximum initial” in the timing chart but they have the “maximum initial” parameter. Their values can be set through controller (Caltrans C8 program: RAM location F-0-E; Bi-Tran: F/1+9+Y). Other programs and controllers have “maximum initial” in the timing charts.

Caltrans C8 program uses parameters “minimum gap”, “maximum gap”, “passage”, “reduce by” and “reduce every” in gap reduction. BiTran is very similar to Caltrans C8 and the difference is that it uses a fixed “reduce by” (i.e. 0.1 sec). SEPAC and other NEMA controllers use “passage”, “minimum gap”, “time to reduce” and “time before reduce” in gap reduction. Figures 4.3-4.7 show associated signal timing data from Districts 4, 6 and 12 of Caltrans and the Cities of Fresno and Irvine.

0			CONTROL CODE 'F' PHASE TIMING 1	2	3	4	5	6	7	8
	0	WALK	O	10				O		0
	1	FLASHING DON'T WALK	O	10				O		O
	2	MIN GREEN	5	10				10		5
	3	TYPE 3 DET (CALL)	O	0				0		0
	4	ADDED/ACT	O	2.0				2.0		0.0
	5	PASSAGE TIME	1.8	3.5				3.5		2.0
	6	MAXIMUM GAP	1.8	4.5				4.5		2.0
	7	MINIMUM GAP	1.8	1.5				1.5		2.0
	8	MAXIMUM EXT 1	15	40				40		20
	9	MAXIMUM EXT 2	15	40				40		20
	A	MAXIMUM EXT 3	15	40				40		20
	B									
LONG PWR FAIL	C	REDUCE BY	O	0.5				0.5		0.0
SHORT PWR FAIL	D	EVERY	O	50				5		00
MAX VAR INIT	E	YELLOW	3.2	3.9				3.9		3.6
RED REVT	F	RED	1.0	1.0				1.0		1.0
		DIRECTION	SAND CANYON WBLT	SAND CANYON EB				SAND CANYON WB		MARINE WAY

Figure 4.3 Timing chart from District 12

INTERVAL	TIMING FUNCTION	φ1	φ2	φ3	φ4	φ5	φ6	φ7	φ8
0	WALK				7				
1	FLASHING DON'T WALK				18				
2	MINIMUM INITIAL		10		4		10		
3	TYPE 3 DET. DISCONNECT		16				16		
4	ADDED SEC./ACTUATION		1		0		1		
5	PASSAGE		3		2		3		
6	MAXIMUM GAP		4		3		4		
7	MINIMUM GAP		2.5		1.2		2.5		
8	MAXIMUM EXTENSION I		35		21		35		
9	MAXIMUM EXTENSION II								
A	MAXIMUM EXTENSION III								
B									
C	SEC. OF GAP REDUCED		0.1		0.1		0.1		
D	PER SEC. OF INTERVAL		1.5		1		1.5		
E	YELLOW		5 ~~4~~		3		5 ~~4~~		
F	RED CLEARANCE		1		0.5		1		

Figure 4.4 Timing chart from District 4

	CALTRAN'S C-8 PROGRAM Version 3 - 11/18/95 F-PAGE								
	Direction		EB		SB		WB	SB	N\S
	Det. Dist.		150		L/TH		130	RT	WALK
	Phase	1	2	3	4	5	6	7	8
0	Walk		7				7		7
1	D. Walk		9				15		17
2	Min Grn		8		8		8	6	8
3	Type III		20		20		20		20
4	Add Intial								
5	Passage		4.1		4.0		3.5	4.0	0.5
6	Max Gap		5.5		6.0		4.7	6.0	0.5
7	Min Gap		2.0		2.5		2.0	2.5	0.5
8	Max 1		30		20		30	20	18
9	Max 2		40		30		40	30	28
A	Max 3								
B	PHASES	1	2	3	4	5	6	7	8
C	Red by		0.1		0.1		0.1	0.1	
D	Every		0.9		0.9		1.1	0.9	
E	Yellow		3.5		3.5		3.5	3.5	3.0
F	Red		0.5		1.0		0.5	0.5	0.1
	C/O w/mid								
	Speed	*0*	*35*	*0*	*35*	*0*	*35*	*35*	*0*

Figure 4.5 Timing chart from District 6

		NBL	SB	WBL	PHASE		NB	EBL	WB
		1	2	3	4	5	6	7	8
0	WALK	0	7	0	7	0	7	0	7
1	DON'T WALK	0	18	0	18	0	18	0	18
2	MIN INITIAL	4	7	4	7	4	7	4	7
3	TYPE 3 LIMIT	0	0	0	16	0	0	0	16
4	ADD PER VEH	0	0	0	0	0	0	0	0
5	VEH EXT	2	5	2	4.5	2	5	2	4.5
6	MAX GAP	2	5	2	6.0	2	5	2.	6.0
7	MIN GAP	2	5	2	2.0	2	5	2	2.0
8	MAX LIMIT	16	23	16	27	16	23	16	27
9	MAXIMUM 2								
A	ADV/DLY WALK								
B	MIN PED CLEAR								
C	COND SRV MIN								
D	REDUCE EVERY	0	0	0	0.5	0	0	0	0.5
E	YELLOW	3.1	3.9	3.1	3.9	3.1	3.9	3.1	3.9
F	RED CLEAR	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Figure 4.6 Timing chart of a signal using Bi-Tran software

PHASE TIMINGS								
PHASE	1	2	3	4	5	6	7	8
Ped Walk:		5		5		5		5
Ped Protect:		22		30		22		30
Add/Actuation:								
Initial:	5	8	5	5	5	8	5	5
Max Initial:								
Extension:	3	5	3	3	3	5	3	3
Min Extension:								
Before Reduce:								
Time to Reduce:								
Max Green:	24	48	24	24	24	48	24	24
Max II Green:	24	48	24	24	24	48	24	24
Yellow Change:	4	5	4	4	4	5	4	4
Red Clearance:	2	2	2	2	2	2	2	2
Bike Timing:								

Figure 4.7 Timing chart of a signal using 820A controller

4.2.3 Yellow and red time

According to Figures 4.3 to 4.7, each phase of a signal has its own yellow and red time.

Due to Paramics' model restriction, the previous actuated signal plugin only has red light and the length of red time is 4 sec (equivalent to the total of yellow plus red). The length of red time is the same for all phases of all signals. Now, the developed plugin still shows red light only but its length is equal to the total of yellow plus red time.

4.2.4 Max green handling

According to Figures 4.3 to 4.7, Caltrans C8 has three "maximum extension" inputs and other signal programs only have two "maximum green" inputs.

As illustrated in Figure 4.8, Caltrans timing charts have a "Control Code 7" part of data explains time-of-day Activity Code. User can enable the use of a certain "maximum green" for a certain time. The default maximum green is "Maximum ext I" (or "max 1" or "Maximum ext 1" or "Max green").

For Bi-Tran signals, there is a "TOD funct." Part of data that may be used to specify which maximum green is used by a time period. As shown in Figure 4.9, "Maximum 2" is used for "maximum green" for all time periods.

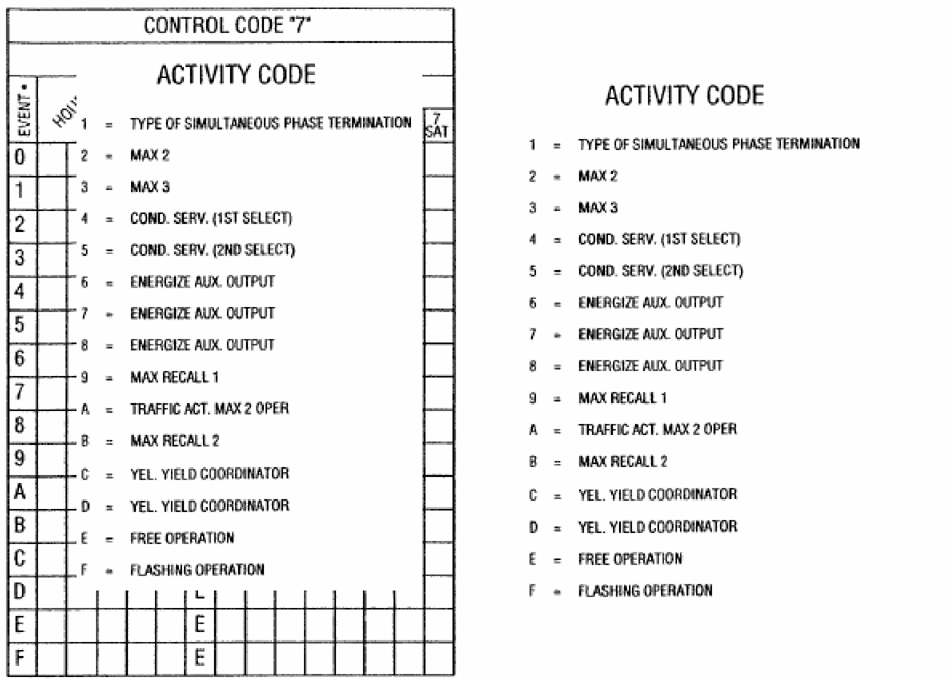


Figure 4.8 Selection of "maximum green" for Caltrans signals

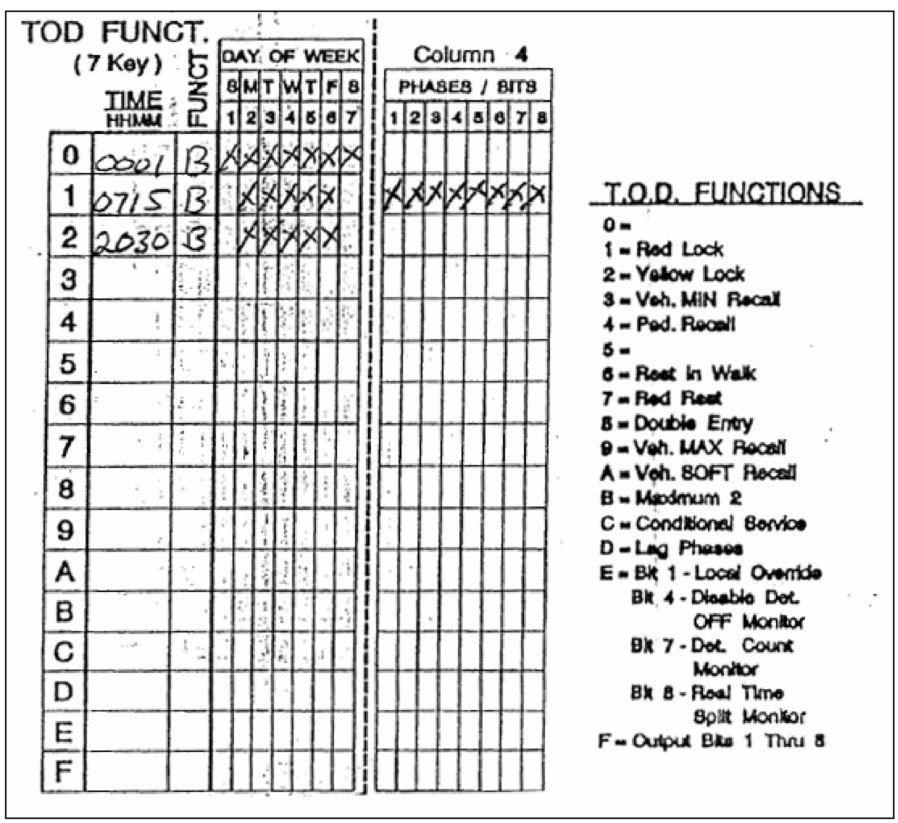


Figure 4.9 Selection of "maximum green" for Bi-Tran signals

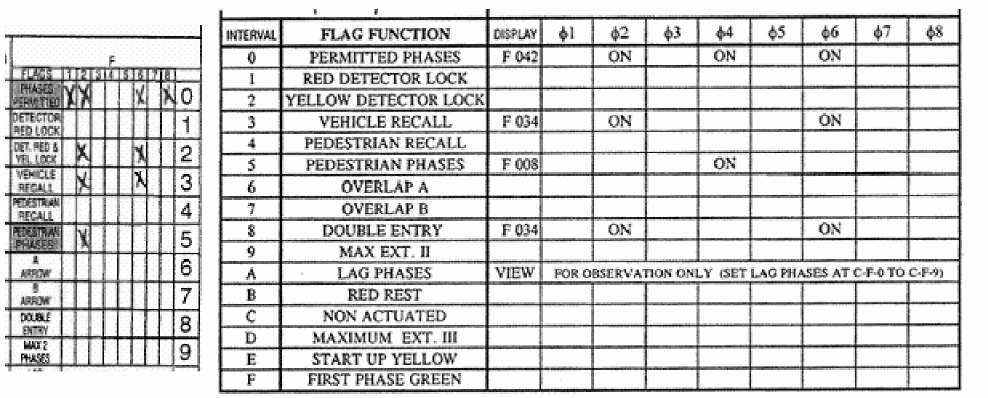
How to start max green timer is different for different programs. For Caltrans C8 and Bi-Trans controllers, max green timer starts at the end of variable initial or min green. For NEMA type controllers, max green timer starts when there is a conflicting call.

4.2.5 Recall mode

There are three widely used recall modes for all controllers: minimum recall, maximum recall and pedestrian recall. The developed plugin is able to emulate these three recall modes.

The previous plugin only allows two phases to be recall phases. The new developed plugin allows user to set any phase to be recall phase.

Caltrans C8 program can set any a phase to be vehicle recall in the "Control code F" in District 6 and 12's timing charts and on bottom of the first page of District 4's timing charts, as shown in Figure 4.10. The default recall in Caltrans C8 program is minimum recall. In order to switch to maximum recall, user can set it up in "Control Code 7", as shown in Figure 4.9. Bi-Tran signal's recall setting is very similar as Caltrans C8, as shown in Figure 4.10.



FLAGS	1	2	3	4	5	6	7	8	F
PHASES PERMITTED	X	X				X		X	0
DETECTOR RED LOCK									1
DET. RED & YEL. LOCK		X				X			2
VEHICLE RECALL		X				X			3
PEDESTRIAN RECALL									4
PEDESTRIAN PHASES		X							5
A ARROW									6
B ARROW									7
DOUBLE ENTRY									8
MAX 2 PHASES									9

INTERVAL	FLAG FUNCTION	DISPLAY	φ1	φ2	φ3	φ4	φ5	φ6	φ7	φ8
0	PERMITTED PHASES	F 042		ON		ON		ON		
1	RED DETECTOR LOCK									
2	YELLOW DETECTOR LOCK									
3	VEHICLE RECALL	F 034		ON				ON		
4	PEDESTRIAN RECALL									
5	PEDESTRIAN PHASES	F 008				ON				
6	OVERLAP A									
7	OVERLAP B									
8	DOUBLE ENTRY	F 034		ON				ON		
9	MAX EXT. II									
A	LAG PHASES	VIEW	FOR OBSERVATION ONLY (SET LAG PHASES AT C-F-0 TO C-F-9)							
B	RED REST									
C	NON ACTUATED									
D	MAXIMUM EXT. III									
E	START UP YELLOW									
F	FIRST PHASE GREEN									

Figure 4.10 Recall settings in District 4 (right) and District 6 and 12 (left)

An actuated signal can thus be operated like semi-actuated signal control by setting the main street to be maximum recall and the crossing street has no recall.

When pedestrian recall is applied to a signal, the "walk" (or "Ped walk") and "Flashing Don't walk" (or "Ped Protect") time must be provided to the input file of the developed plugin.

4.2.6 Actuated signal coordination

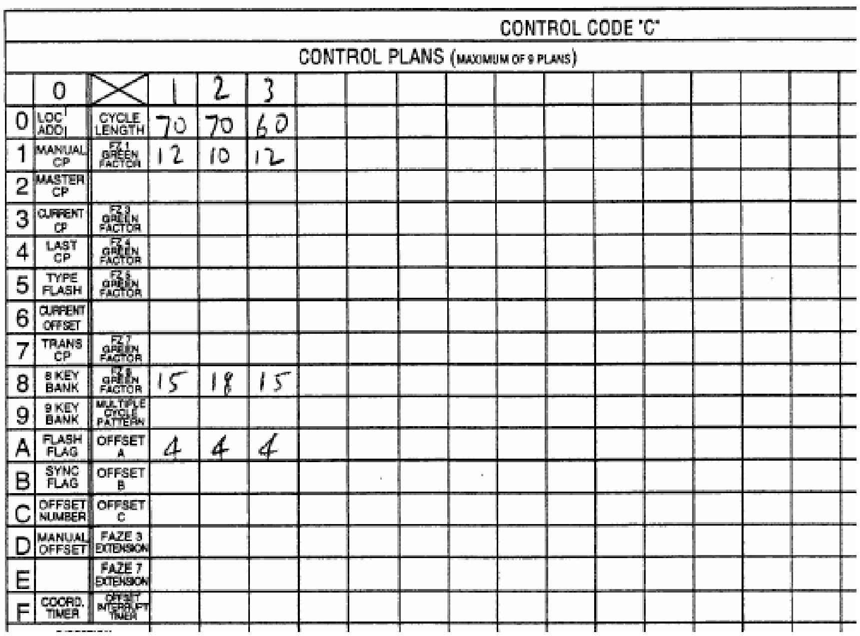
The developed plugin implements the coordination method of Caltrans C8 program. The theory needs any phase of a cycle to have a force-off point. The start of a cycle represented using offset (or yield point) is the end of green of the coordinated phases. The force-off points for coordinated phases are always zero. For non-coordinated phases, force-off points can be calculated based on the green splits or green factors assigned to the phase and other phases and the cycle length.

The signal timing charts provide cycle length, green factors or green splits for all non-coordinated phases, and three offset settings. A sample coordination plan from District 12 is shown in Figure 4.11. The coordinated phases can be easily derived, which are 2 and 6 if taking a look at data in "Control Code F" shown in Figure 4.3. District 6's coordination plan data are almost the same as

District 12. However, District 4's timing charts show coordination plan in a different way, as illustrated in Figure 4.12.

It is found that Bi-Tran signals may have similar coordination logic as Caltrans C8 program. The coordination data for Bi-Tran are cycle length, force-off points for all phases, and three offsets, as shown in Figure 4.13. The developed plugin can accept both green factors and force-offs in order to be work with both controllers.

However, SEPAC program has different coordination methods. The current plugin does not implement them but user can convert SEPAC coordination data in the format that can be accepted by the plugin. Then, the plugin will operate signals using Caltrans C8's logic.



CONTROL CODE 'C'

CONTROL PLANS (MAXIMUM OF 9 PLANS)

	0		1	2	3
0	LOC'L ADD	CYCLE LENGTH	70	70	60
1	MANUAL CP	FZ 1 GREEN FACTOR	12	10	12
2	MASTER CP				
3	CURRENT CP	FZ 3 GREEN FACTOR			
4	LAST CP	FZ 4 GREEN FACTOR			
5	TYPE FLASH	FZ 5 GREEN FACTOR			
6	CURRENT OFFSET				
7	TRANS CP	FZ 7 GREEN FACTOR			
8	8 KEY BANK	FZ 8 GREEN FACTOR	15	18	15
9	9 KEY BANK	MULTIPLE CYCLE PATTERN			
A	FLASH FLAG	OFFSET A	4	4	4
B	SYNC FLAG	OFFSET B			
C	OFFSET NUMBER	OFFSET C			
D	MANUAL OFFSET	FAZE 3 EXTENSION			
E		FAZE 7 EXTENSION			
F	COORD. TIMER	OFFSET INTERRUPT TIMER			

Figure 4.11 Sample coordination plans from District 12

PATTERN 1			
CODE	FUNCTION	ENTER	DISPLAY
C-1-0	CYC. LENG.	65 E	C 065
C-1-1	φ 1 SPLIT	E	C
C-1-2	φ 2 SPLIT	E	C
C-1-3	φ 3 SPLIT	E	C
C-1-4	φ 4 SPLIT	16 E	C 016
C-1-5	φ 5 SPLIT	E	C
C-1-6	φ 6 SPLIT	E	C
C-1-7	φ 7 SPLIT	E	C
C-1-8	φ 8 SPLIT	E	C
C-1-A	OFFSET A	5 E	C 005
C-1-B	OFFSET B	E	C
C-1-C	OFFSET C	E	C

PATTERN 3			
CODE	FUNCTION	ENTER	DISPLAY
C-3-0	CYC. LENG.	75 E	C 075
C-3-1	φ 1 SPLIT	E	C
C-3-2	φ 2 SPLIT	E	C
C-3-3	φ 3 SPLIT	E	C
C-3-4	φ 4 SPLIT	18 E	C 018
C-3-5	φ 5 SPLIT	E	C
C-3-6	φ 6 SPLIT	E	C
C-3-7	φ 7 SPLIT	E	C
C-3-8	φ 8 SPLIT	E	C
C-3-A	OFFSET A	8 E	C 008
C-3-B	OFFSET B	E	C
C-3-C	OFFSET C	E	C

PATTERN 2			
CODE	FUNCTION	ENTER	DISPLAY
C-2-0	CYC. LENG.	75 E	C 075
C-2-1	φ 1 SPLIT	E	C
C-2-2	φ 2 SPLIT	E	C
C-2-3	φ 3 SPLIT	E	C
C-2-4	φ 4 SPLIT	12 E	C 012
C-2-5	φ 5 SPLIT	E	C
C-2-6	φ 6 SPLIT	E	C
C-2-7	φ 7 SPLIT	E	C
C-2-8	φ 8 SPLIT	E	C
C-2-A	OFFSET A	0 E	C 000
C-2-B	OFFSET B	E	C
C-2-C	OFFSET C	E	C

PATTERN 4			
CODE	FUNCTION	ENTER	DISPLAY
C-4-0	CYC. LENG.	E	C
C-4-1	φ 1 SPLIT	E	C
C-4-2	φ 2 SPLIT	E	C
C-4-3	φ 3 SPLIT	E	C
C-4-4	φ 4 SPLIT	E	C
C-4-5	φ 5 SPLIT	E	C
C-4-6	φ 6 SPLIT	E	C
C-4-7	φ 7 SPLIT	E	C
C-4-8	φ 8 SPLIT	E	C
C-4-A	OFFSET A	E	C 000
C-4-B	OFFSET B	E	C
C-4-C	OFFSET C	E	C

Figure 4.12 Sample coordination plans from District 4

		PLAN NUMBER								
		1	2	3	4	5	6	7	8	9
0	CYCLE LENGTH	78	92	105						
1	FORCE OFF 1	33	41	47						
2	FORCE OFF 2	18	58	67						
3	FORCE OFF 3	49	22	26						
4	FORCE OFF 4	0	0	0						
5	FORCE OFF 5	33	41	47						
6	FORCE OFF 6	18	58	67						
7	FORCE OFF 7	49	76	87						
8	FORCE OFF 8	0	22	26						
9	RING OFFSET	6	0	0						
A	OFFSET A	0	0	0						
B	OFFSET B	0	0	0						
C	OFFSET C	0	0	0						
D	END PERM 1	5	5	5						
E	HOLD RELEASE	255	255	255						
F	ZONE OFFSET	0	0	0						

Figure 4.13 Sample coordination plans from Bi-Tran signals

4.2.7 Time-of-day control

An actuated signal's operation modes can be set to be (a) free mode actuated signal control; (b) actuated signal coordination.

Signal timing charts usually provide time-of-day plans. The information provided in the time-of-day plan associates a certain timing period with signal operation mode, either free mode or coordination mode. If signal is operated under coordination, the coordination plan number and its offset setting are specified.

A sample time-of-day plan is shown in Figure 4.14. Control plan "E" means that signal is operated under free mode. If control plan is a digit, it corresponds to a coordination plan that can be found in the coordination plan (examples are Figure 4.9 or 4.10).

CONTROL CODE "9"

TIME OF DAY SELECTION FOR COORDINATED CONTROL PLANS

C-0-9 = 1 OR 0

DATE	BY	EVENT *	TIME SETTING FOR CONTROL PLAN TO BE OPERATED: HOUR	MINUTE	CONTROL PLAN	OFFSET	DEPRESS "E"	DAY OF WEEK SET DISPLAY LIGHTS 1-7: 1 SUN	2 MON	3 TUES	4 WED	5 THU	6 FRI	7 SAT
4/11/02	DP	0	06	00	1	A	E		X	X	X	X	X	
4/11/02	DP	1	09	30	1	A	E		X	X	X	X	X	
4/11/02	DP	2	14	30	2	A	E		X	X	X	X	X	
4/11/02	DP	3	19	00	E	A	E		X	X	X	X	X	
4/11/02	DP	4	08	00	1	A	E	X						X
4/11/02	DP	5	19	00	E	A	E	X						X
		6					E							
		7					E							
		8					E							
		9					E							
		A					E							
		B					E							
		C					E							
		D					E							
		E					E							
		F					E							

Figure 4.14 Caltrans TOD plan

4.2.8 Lag phases

Lag phase information can be found in signal timing charts, as shown in Figure 4.15. Lag phases are set up through the "priorities" file for a simulation model. Paramics provides a way to set up time-dependent phase sequences. Please check Paramcis manuals for the method.

C-F-0	LAG FAZES "FREE"	2 4 6 8	2 4 6 8	C 170
C-F-1	LAG FAZES "PATTERN 1"	2 4 6 8	2 4 6 8	C 170
C-F-2	LAG FAZES "PATTERN 2"	2 4 6 8	2 4 6 8	C 170
C-F-3	LAG FAZES "PATTERN 3"	2 4 6 8	2 4 6 8	C 170
C-F-4	LAG FAZES "PATTERN 4"			
C-F-5	LAG FAZES "PATTERN 5"			
C-F-6	LAG FAZES "PATTERN 6"			
C-F-7	LAG FAZES "PATTERN 7"			
C-F-8	LAG FAZES "PATTERN 8"			
C-F-9	LAG FAZES "PATTERN 9"			

FLAGS									
F	1	2	3	4	5	6	7	8	
LAG FZ FREE		X		X		X		X	0
LAG FZ CP 1		X		X		X		X	1
LAG FZ CP 2		X		X		X		X	2
LAG FZ CP 3		X		X		X		X	3
LAG FZ CP 4									4
LAG FZ CP 5									5
LAG FZ CP 6									6
LAG FZ									7

Figure 4.15 Phase sequence from District 4 (left) and District 12 (right)

4.2.9 Output data

In order to analyze signals' performance, the developed plugin outputs actual green splits per cycle for each signal when a cycle is completed. An example of the output file is illustrated in Figure 4.16. Using the output data, the following analysis can be performed:

- Whether demands/routing of the simulation model make sense
- Whether signal timing needs to be optimized

Time	signal	cycle	phase 1	phase 2	phase 3	phase 4	phase 5	phase 6	phase 7	phase 8
6:03:43	eltoro5sb1	98	14	84	0	0	0	98	0	0
6:04:27	eltoro5sb1	44	18.2	25.8	0	0	0	44	0	0
6:05:25	eltoro5sb1	58	14.6	43.4	0	0	0	58	0	0
6:06:24	eltoro5sb1	58.4	14	44.4	0	0	0	58.4	0	0
6:07:24	eltoro5sb1	60.8	27	33.8	0	0	0	60.8	0	0
6:08:09	eltoro5sb1	44.2	15	29.2	0	0	0	44.2	0	0
6:08:51	eltoro5sb1	42.4	22.4	20	0	0	0	42.4	0	0
6:09:34	eltoro5sb1	43	23	20	0	0	0	43	0	0
6:01:32	eltoro5sb2	91.8	0	76.8	0	15	0	76.8	15	0
6:02:07	eltoro5sb2	35.4	0	21.4	0	14	0	21.4	14	0
6:02:49	eltoro5sb2	41.6	0	25.2	0	16.4	0	25.2	16.4	0
6:03:23	eltoro5sb2	34	0	20	0	14	0	20	14	0
6:04:02	eltoro5sb2	39	0	25	0	14	0	25	14	0
6:04:41	eltoro5sb2	39.8	0	21.4	0	18.4	0	21.4	18.4	0
6:05:27	eltoro5sb2	45.4	0	20	0	25.4	0	20	25.4	0
6:06:05	eltoro5sb2	38.4	0	24.4	0	14	0	24.4	14	0
6:07:07	eltoro5sb2	61.8	0	35.4	0	26.4	0	35.4	26.4	0
6:08:01	eltoro5sb2	54	0	31.2	0	22.8	0	31.2	22.8	0
6:08:55	eltoro5sb2	54	0	20	0	34	0	20	34	0
6:09:41	eltoro5sb2	46	0	26.6	0	19.4	0	26.6	19.4	0
6:10:16	eltoro5sb2	35	0	20	0	15	0	20	15	0

Figure 4.16 Output file

4.3 Coding Signal Timing

Coding an actuated signal in Paramics may be as complicated as setting up signals in the real world. User needs to understand the control logic, data shown in signal timing charts, and also follows procedure and rules of the developed plugin.

In the last section all functions are explained of the developed plugin and also help user understand the signal timing charts from the field. This section will explain how to set up actuated signals using the plugin. There are four steps:

(1) Geometry checking

(2) Add detectors

(3) Prepare signal_control file

(4) Prepare priorities file

(5) Load pluign with network for simulation

(6) Signal coding checking

4.3.1 Geometry checking

The first step is to verify if all necessary geometric data have been coded correctly. It involves the checking of the following data:

(7) Number of lanes

(8) Lane use, i.e. which lane is assigned to left turn, through, and right turns for each approach of a signalized intersection

(9) Lane assignment (or *next lane* information), i.e. assigning a lane on the upstream link to specific lanes of the downstream link. For example, an approach has two left turn lanes. There are three lanes on the downstream link. So, the next lane for the median side left-turn lane is assigned to the median side of the lane (i.e. lane 3) on the downstream link. The other left-turn lane is assigned to the other two lanes on the downstream link.

Please note:

(1) Not only links around the signal node but also those nodes / links close to the intersection need to be checked.

(2) If there is a left-turn bay for an approach of an intersection, the approach is usually modeled as two links and the upstream link has less number of lanes compared to the downstream link that ends at the intersection. The shared node of the two links is important. Usually, *next lane* needs to be set for the node. If there are 3 lanes for the upstream links and 5 lanes for the downstream link, the only *next lane* setting required for the node is "from lane 3 to lane 3".

(3) For a busy intersection, an approach may need to set up lane choices to ensure left-turn vehicles not to occupy through lanes.

4.3.2 Add detectors

The second step is to add detectors around the intersection. The layout of a typical actuated signal intersection is shown in Figure 4.17. The vehicle detection is an important part of the actuated signal system. There are three groups of detectors in each approach for a typical intersection in the real world:

(1) Stopline detectors, located on the through lanes and very close to the stop line, for the presence detection of through vehicles. There may be 2-3 presence detectors for a lane;

(2) Advance loop detector (or extension detector or setback detector), located at almost 150—300 feet from the stop line, used to detect vehicles for the extension of the through movement phase; and

(3) Long loop detector (or several single loops) for left turns, with the length of about 50-70 feet, for the presence detection of left turn vehicles.

For some intersections, there may be no advance detector or stopline detector at some approaches of an intersection. If presence detectors are only placed on the minor streets, it is a semi-actuated signalized intersection.

To better simulate the functionality of detectors, ideally, detectors should be modeled in Paramics according to the real-world configuration. However, in Build 3 of Paramics, detectors are not lane specific. A detector covers all lanes of a link and thus a Paramics detector represents a detector station. Therefore, we cannot model a separate long loop (for left turn use) in the actuated signal system. As a result, we use three small detectors instead of a long loop, as shown in Figure 4.18. Three 2 m or 6.6 ft detectors are used to mimic one 50ft long loop detector.

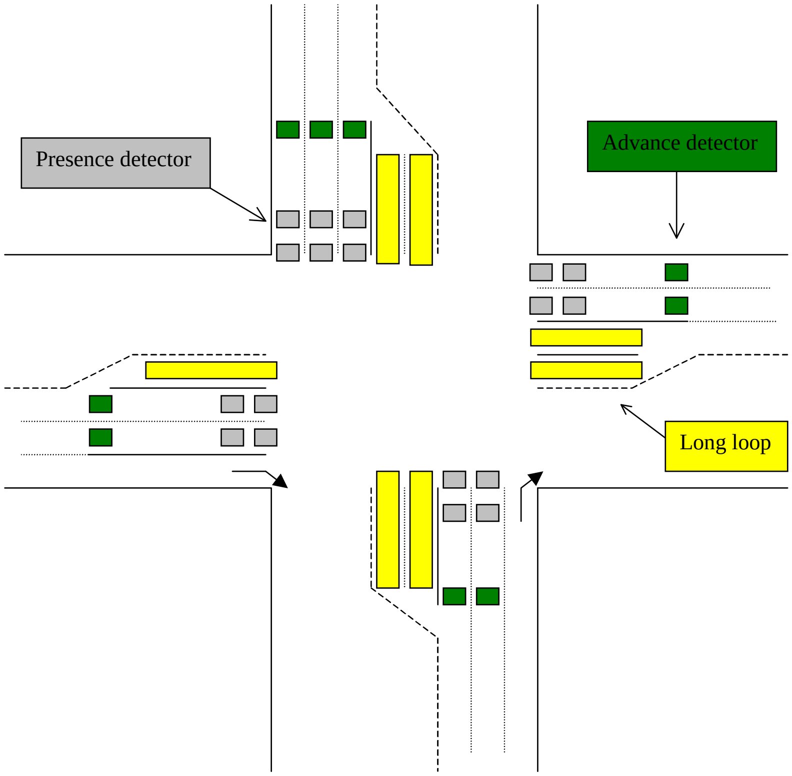


Figure 4.17 Typical Intersection Layout

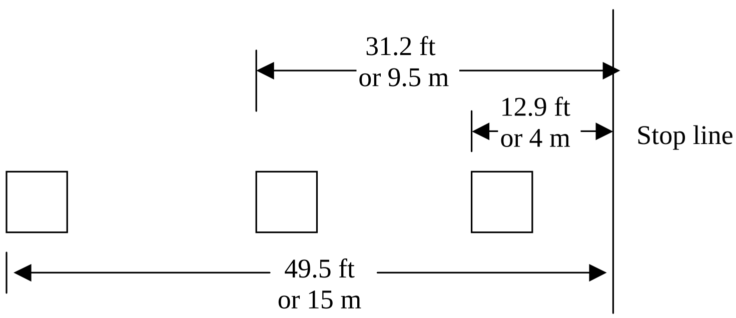


Figure 4.18 Modeling the left turn long loop detector

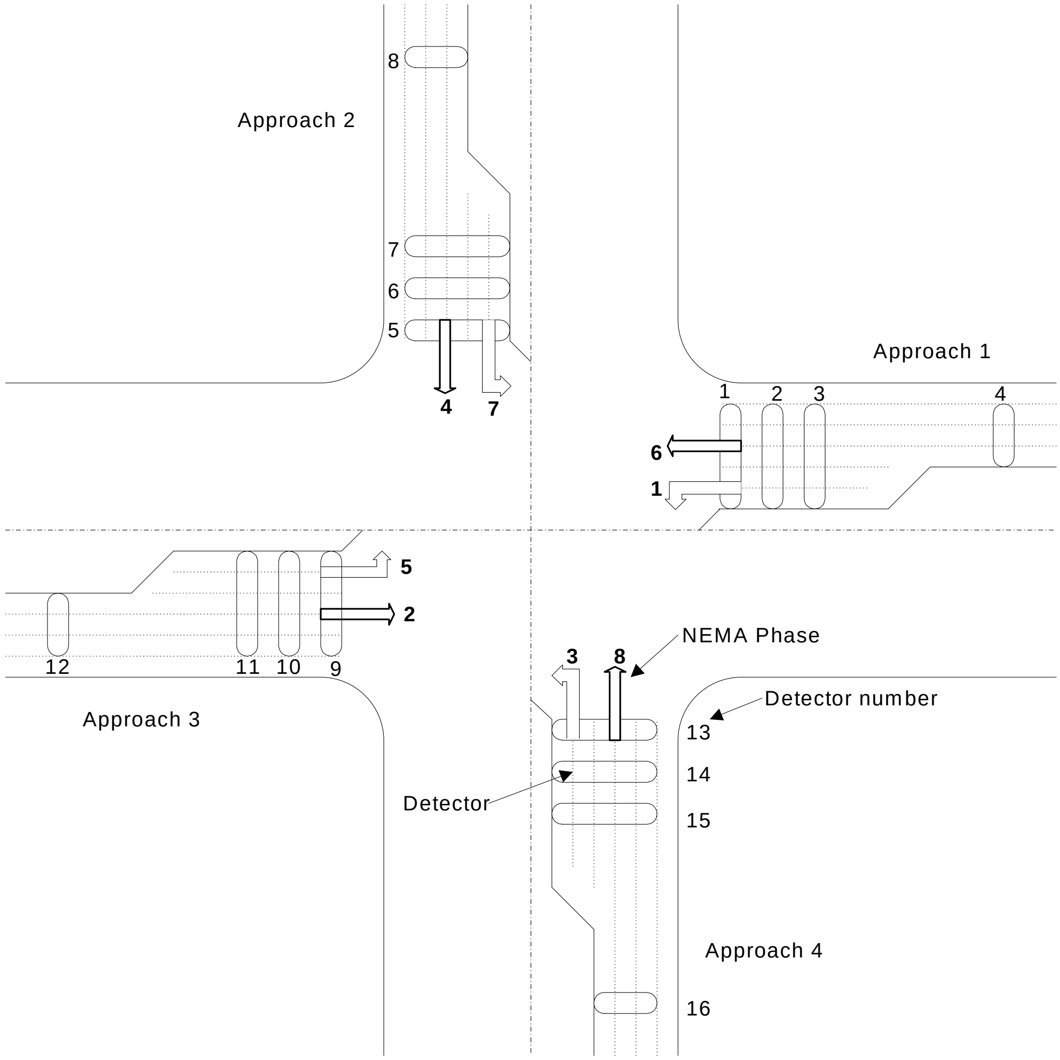


Figure 4.19 Typical Intersection Layout with NEMA phases

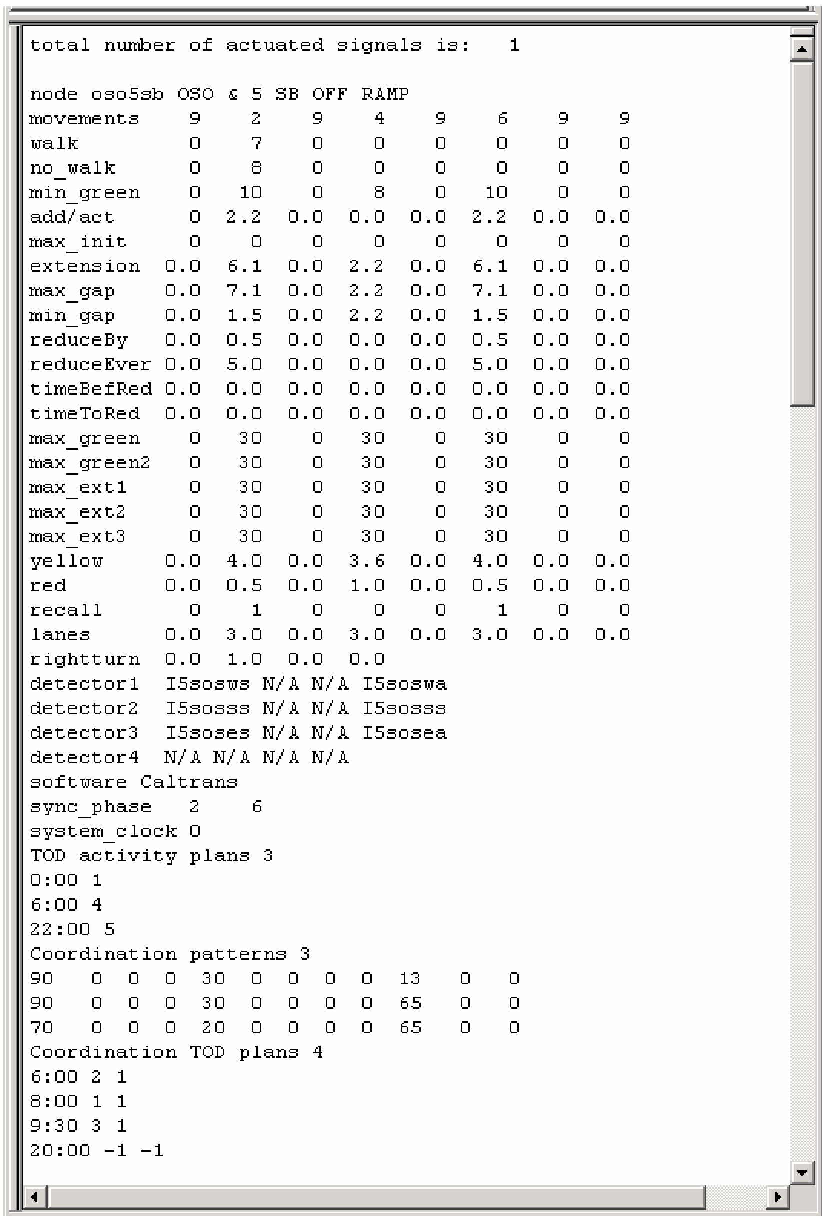
As illustrated in Figure 4.19, we modeled 16 detectors for a typical intersection in Paramics and each detector covers all lanes of a link. For each approach, there are three detectors close to the stop line for through and left-turn vehicle presence detection, and one advance detector located at about 150-300 feet to the stop line for detecting vehicles for the extension of the through movement phase. For stopline detectors, all three of them employ the vehicle presence of left turn lanes; the two detectors close to the stop line are used for detecting the presence of through vehicles.

Due to the improvement of long loop detection in a later version of Paramics (later than Build V.3.0.7), one long loop can be used instead of three stopline loop detectors for vehicle presence. As a result, we only need to model 8 detectors for an intersection. That is to say, detectors 1, 5, 9 and 13 are long loops (with a typical length of 50 feet), and there is no need to code detectors 2, 3, 6, 7, 10, 11, 14, and 15. This is our recommended method to model detectors of an actuated signal intersection.

After version 4.0 of Paramics, detectors can be lane specific. This plugin does not support the use this type of detectors.

4.3.3 Prepare signal_control file

The plugin requires a file titled “signal_control” to be in the Paramics network directory. An example of the “signal_control” file is shown in Figure 4.20. The first line of this file specifies the number of actuated signals modeled in the network. The remainder of the file contains the signal timing information for each signal.



```
total number of actuated signals is:   1

node oso5sb OSO & 5 SB OFF RAMP
movements    9    2    9    4    9    6    9    9
walk         0    7    0    0    0    0    0    0
no_walk      0    8    0    0    0    0    0    0
min_green    0   10    0    8    0   10    0    0
add/act      0  2.2  0.0  0.0  0.0  2.2  0.0  0.0
max_init     0    0    0    0    0    0    0    0
extension  0.0  6.1  0.0  2.2  0.0  6.1  0.0  0.0
max_gap    0.0  7.1  0.0  2.2  0.0  7.1  0.0  0.0
min_gap    0.0  1.5  0.0  2.2  0.0  1.5  0.0  0.0
reduceBy   0.0  0.5  0.0  0.0  0.0  0.5  0.0  0.0
reduceEver 0.0  5.0  0.0  0.0  0.0  5.0  0.0  0.0
timeBefRed 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
timeToRed  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
max_green    0   30    0   30    0   30    0    0
max_green2   0   30    0   30    0   30    0    0
max_ext1     0   30    0   30    0   30    0    0
max_ext2     0   30    0   30    0   30    0    0
max_ext3     0   30    0   30    0   30    0    0
yellow     0.0  4.0  0.0  3.6  0.0  4.0  0.0  0.0
red        0.0  0.5  0.0  1.0  0.0  0.5  0.0  0.0
recall       0    1    0    0    0    1    0    0
lanes      0.0  3.0  0.0  3.0  0.0  3.0  0.0  0.0
rightturn  0.0  1.0  0.0  0.0
detector1  I5sosws N/A N/A I5soswa
detector2  I5sosss N/A N/A I5sosss
detector3  I5soses N/A N/A I5sosea
detector4  N/A N/A N/A N/A
software Caltrans
sync_phase   2    6
system_clock 0
TOD activity plans 3
0:00 1
6:00 4
22:00 5
Coordination patterns 3
90   0  0  0  30  0  0  0  0  13   0   0
90   0  0  0  30  0  0  0  0  65   0   0
70   0  0  0  20  0  0  0  0  65   0   0
Coordination TOD plans 4
6:00 2 1
8:00 1 1
9:30 3 1
20:00 -1 -1
```

Figure 4.20 Sample signal_control file

Please note that

(1) “movement” line represents phase sequence information, which should be based on Figure 4.15. If a phase doesn’t exist, movement is written as 9. If phase 2, 4, 6, and 8 are lagging phases, timing data should be written with the sequence of “1 2 3 4 5 6 7 8”. If phase 1 , 4, 6, and 8 are lagging phases, the timing data should be with the sequence of “2 1 3 4 5 6 7 8”.

(2) Not all data are needed to operate a signal. How to prepare “signal_control” file needs to be based on what data are provided by the signal timing chart. However, all lines should be written in the sequence shown in Figure 4.20. For example, “max_gap” should be written before “min_gap”.

(3) Software could be “Caltrans”, “Bi-Tran”, or “SEPAC”.

(4) Data in “coordination pattern” include cycle length, eight green factors (for”Caltrans” signals) or force-off points (for “Bi-Tran” signals) or green split percentage (for “820A”, “FPAC” and “SEPAC” signals), and three offsets (Caltrans signals have three offsets but others may only have one. If there is only one offset, the other two need to be written as 0).

4.3.4 Prepare priorities file

The “priorities” file defines what movements are allowed under each phase of an intersection. For pre-timed signal control, the priorities information can be edited through the Paramics GUI. However, for the actuated signal, the file “priorities” must be edited directly with a text editor.

To prepare priorities file, “movement” information needs to be used. If phase sequence is from 1 through 8, “priorities” information needs to be prepared based on Figure 4.21.

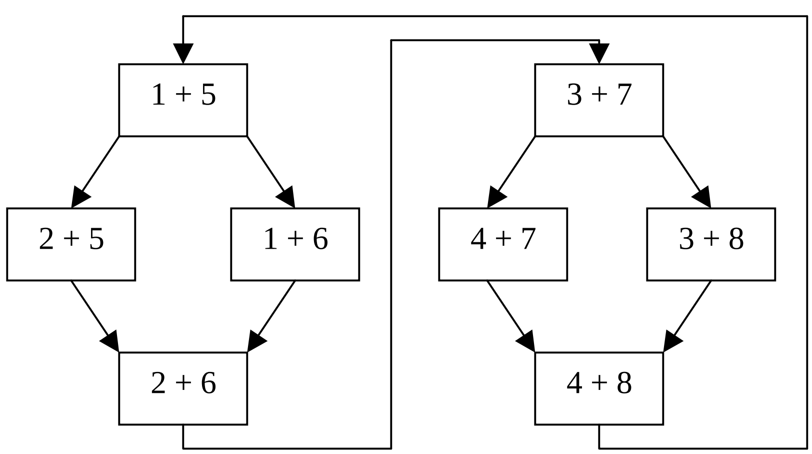


Figure 4.21 Eight phases of the four-legged full-actuated signal intersection

Figure 4.22 is an example of the node designations for a four-legged intersection. “approach 1” is considered to be in the direction starting at node 7511 and heading towards the junction node 528z.

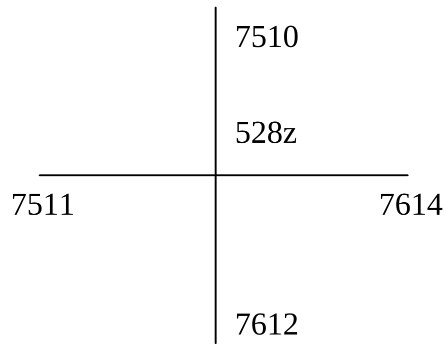


Figure 4.22 Intersection Layout

The “priorities” for a four-legged full-actuated intersection will have eight phases. As illustrated in Figure 4.21, “Phase 1” will correspond to the situation where the left-turning NEMA movements 1 and 5 show green. “Phase 2” will account for the situation where movements 5 and 2 show green, and “phase 3” will be for movements 1 and 6. “Phase 4” will be for the through movements 2 and 6. The last four phases will follow the pattern of the first four phases, starting with the left-turn movements 3 and 7.

For the intersection in the previous figure, the definition of phases and actions (movements) in “priorities” file would be:

```
actions 528z
phase offset 0.00 sec
phase 1
      0.00
      max 100.00
red phase 0.00
fill
all barred except
from 7510 to 7511 minor
from 7511 to 7612 minor
from 7511 to 7510 major
from 7612 to 7614 minor
from 7614 to 7612 major
from 7614 to 7510 minor
phase 2
      0.00
      max 100.00
red phase 0.00
fill
all barred except
from 7510 to 7511 minor
from 7511 to 7612 minor
```

```
from 7511 to 7614 major
from 7511 to 7510 major
from 7612 to 7614 minor
from 7614 to 7510 minor
phase 3

…

phase 8
      0.00
      max 100.00
red phase 0.00
fill
all barred except
from 7510 to 7511 minor
from 7510 to 7612 major
from 7511 to 7612 minor
from 7612 to 7614 minor
from 7612 to 7510 major
from 7614 to 7510 minor
```

In this example, the movements of each phase are "major" while all right turns are "minor". We set the default signal time of each phase as 0 sec (This is the reason that we cannot edit these "actions" information through GUI). The plugin will assign a certain length of time to each phase based on the presence of vehicles.

Then, update the above priorities information of the corresponding signalized node in the "priorities" file of the network.

Please note that the network with modified "priorities" file must use together with this actuated signal plugin. Without this plugin, all movements of those actuated signal intersections are in red light.

4.3.5 Loading plugin

After the completion of the "signal_control" file and the update the "priorities" file, you can load the simulation network together with this plugin. The name of this plugin file is "actuated_signal.dll", which can be saved in the Paramics root folder.

There should be a file called "programming" in the simulation network folder and the content of the file is "actuated_signal.dll" if the network only needs one plugin. Then, load the network using Modeller and then run simulation. If there is no problem with signal coding, you will see that this plugin is used to emulate the actuated signal control. If there is any problem occurring when the network and the control file is loaded, error messages may be shown in the reporter window. There is an output file of the plugin, called Log-signal.txt. It can be found under the network folder. User may need to check it.

4.3.6 Checking signal coding

A signal needs to be roughly checked after it is coded. Signals coded should perform in the way they perform in the field. Inappropriate signal coding may cause a signal to perform weirdly.

Basic signal control knowledge, which can be obtained from textbooks, signal control program manuals or from traffic engineers in charge of traffic signals, is required during signal checking.

When simulation starts, the first two un-conflicted recall phases show green first. Then, the green light will be given to appropriate phases based on phase sequence and the presence of vehicles. If gridlocks occur at the target signalized intersection after running simulation for a while, the coding of the signal may have some problems.

4.4 Arterial calibration

4.4.1 Method of Calibration

Arterial calibration involves the comparison of observed and simulated data. The performance measures that are usually used include:

(1) Queue length

(2) Delay

(3) Number of stops on an approach

(4) Turning counts / percentages

If there is more than one signal in the target network, these signals may need to be checked in sequence since signal progression may affect the performance of downstream signals.

If the comparison of the above-mentioned measures shows significant difference between observed and simulated data, it means that the signal doesn't perform as expected. It may be caused by several issues:

(5) Signal coding

(6) Demand

(7) Routing

(8) Inappropriate signal timing

4.4.2 Signal coding

This step involves the checking of signal coding, including stopline and advance detector location, lane usage and assignment, length of the left turn bay, and signal timing data, etc.

The placement of stopline and advance detectors deserves to be mentioned specifically since it is very important for signal operation. Inappropriate placement of a stopline detector may cause a signal phase not to be called. For signals using volume-density control, gap is changed from the extension time to the minimum gap. If the advance detector is not placed at a right location, it affects a signal's operation since gap out and max out are two major reasons for ending a signal phase.

It is noted that it is hard to place a detector at an exact location through Paramics GUI. This is the reason that a user may need to edit detectors file directly. The advance detector needs to be placed at a location setback distance to the stopline. If the setback distance is unknown, it is equal to extension * (link speed limit). The stopline detector needs to be located at the end of a link. It should be written as follows in the "detectors" file.

```
loop "1011d1" at 25.0 ft on link 1010:1011 length 25.0 ft
loop "1016d1" at 25.0 ft on link 1015:1016 length 20.0 ft
```

For the above example, the second detector is actually placed beyond the end of the link, which is allowed by Paramics to model those practices used in the field.

If there is no signal coding issue, the signal's performance may need to be checked in more details based on the following measures.

(1) Green time assigned to each phase

(2) Max_out

(3) Gap_out

(4) v/c ratio

If any of the following occur, problems can then be identified:

(1) If a phase of a signal always ends with maximum green while other conflicting phases are not.

(2) If a phase's green always ends with gap out.

(3) If green splits of some phases are close to maximum green but other conflicting phases are not.

4.4.3 Demand and routing

Demand and routing are two factors that need to be considered at the same time during calibration. For a large network, in order to know O-D and turning information of vehicles passing an intersection, the detector aggregator plugin can be used to capture these data. A sample input file for the detector data aggregator plugin is shown in Figure 4.23. The third line tells the plugin that the O-D information collection interval is 300 sec. In addition, the plugin has another input file that can be used to report turning information. The input file is called "turning_control". Figure 4.24 shows an example of the file.

The O-D output of the detector data aggregator plugin tells user which O-D pairs' vehicles pass the intersection and what is the proportion of the O-D flow over the total flow for a certain time period. The turning information output tells user the turning counts of each specified turn in the input file for each time interval.

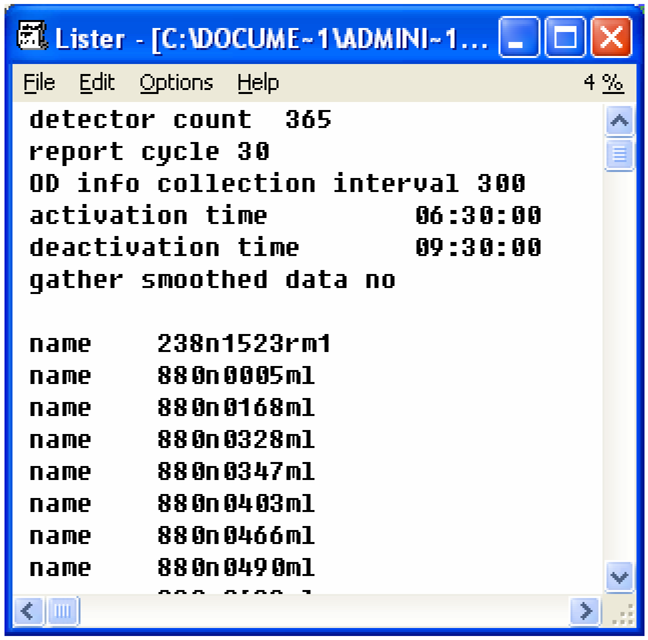


Figure 4.23 Sample input file "loop_control" for the detector data aggregator plugin

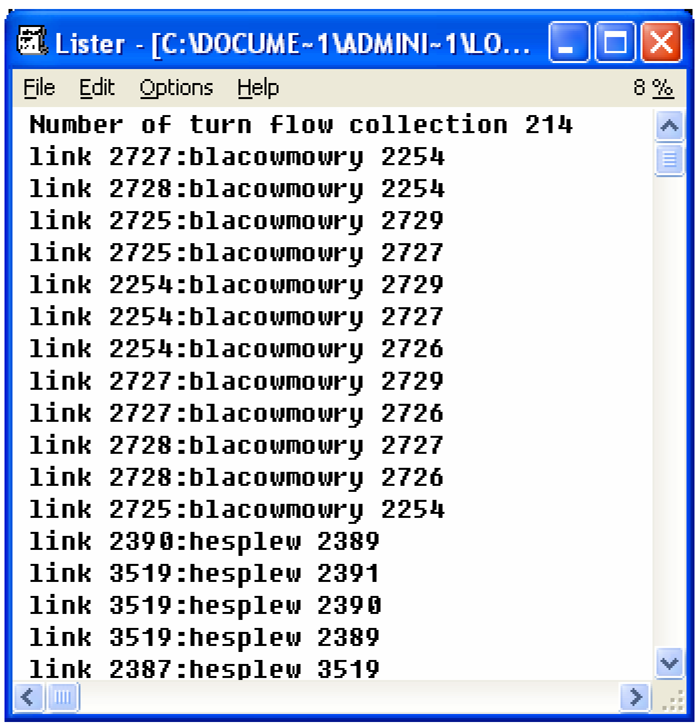


Figure 4.24 Sample input file "turning_control" for the detector data aggregator plugin

4.4.4 Inappropriate signal timing data

If there is no problem found with signal coding, demand and routing, the problem could lie with inappropriate signal timing. Users may need to check first whether signal timing data are the right ones. Then, users may need to check with the local transportation agency if the timing data provided are correct and complete. Sometimes, signals are operated under signal coordination but the coordination related data may not be provided.

4.4.5 Signal optimization

If no problems are found with signal coding, demand, routing, and signal timing, one may still need to optimize signal timing in order to obtain satisfactory calibration results. When demand or network characteristics change, the signal timing provided may no longer be adequate. For the baseline network, this step may not be needed. However, users may need to consider it under the alternative scenarios.

Based on observed turning count data, signal optimization can be done using the Synchro software. Users may need to re-code the network in Synchro or generate the corresponding Synchro network using the converter developed by University of Minnesota. After Synchro performed optimization, the signal parameters need to be implemented in the "signal_control" and "priorities" files. Currently, there is no interface between this plugin and Synchro. It is noted that University of Minnesota developed a preliminary version of the interface between the previous version of the plugin and Synchro. However, this interface does not work with our current plugin. We leave this to our future work.

4.5 Summary

An actuated signal control plugin was developed to enhance the actuated signal control capabilities of the Paramics software. These include adding a volume-density control function, detector placement and output diagnoses, and a conversion of SEPAC coordination data to a format acceptable by Caltrans C8's logic. We also provide a detailed, step by step procedure to code, diagnose and optimize arterial traffic signals in Paramics. These can all be used together to ease the effort of calibrating traffic signal operations and improve simulation performance.

CHAPTER 5 THE SUMMARY STATISTICS TOOL

Introduction

Paramics provides a good graphical user interface to view the results and other network related statistics. For example, Paramics Analyser provides many features to analyze network and the data associated with it. Some of the statistics that Analyser provides are displaying turning counts, delay and journey time data, Origin and Destination demand data using statistics display manager. Besides, it also provides detailed reports for further analysis. However, Paramics does not provide a tool to diagnose the potential problems encountered in a calibration.

The purpose of developing a summary statistics tool is to provide a simple and easy to use graphical user interface for monitoring the calibration process in progress, viewing calibration results and generating a final report on the calibration results. The advantage of having this tool is that the user need not look at the output files created by the calibration and assemble them to assess whether a calibration is successful.

5.2 Features of The Tool

The summary statistics tool provides a summary of the calibration results of global and local driving behavior parameters and the basic statistics to verify the results. The foremost criteria to ascertain the proximity to the actual values is the overall convergence curve. This curve is saved in the spreadsheet, on using the ‘Generate Report’ option, to view and, based on the user’s judgment the results can be used for further verification like comparing the simulation results with other observed dataset. Before comparing the simulation results with observed data, user can look at the so-called Fundamental Diagram produced by the simulation outputs and make sure it reproduces the actual capacity and the shape of the flow-occupancy plot observed at that location. Figure 5.1 shows a Fundamental Diagram in progress for a test network which can give a fair idea if the capacity and the shape are being reproduced with the calibrated parameters. In this case, both were not reproduced well, so further calibration is needed.

The convergence curve is of the utmost importance in the calibration process. On a very first look at the convergence curve a user can tell how the calibration is progressing and if the optimal solution has been reached. This, in actual, is controlled by GA’s control parameters while running the Global and Local Calibration tool. It shows the pattern of fitness values along the generations of the GA algorithm. As the curve flattens the solution moves towards optimality. Figure 5.2 shows the convergence curve.

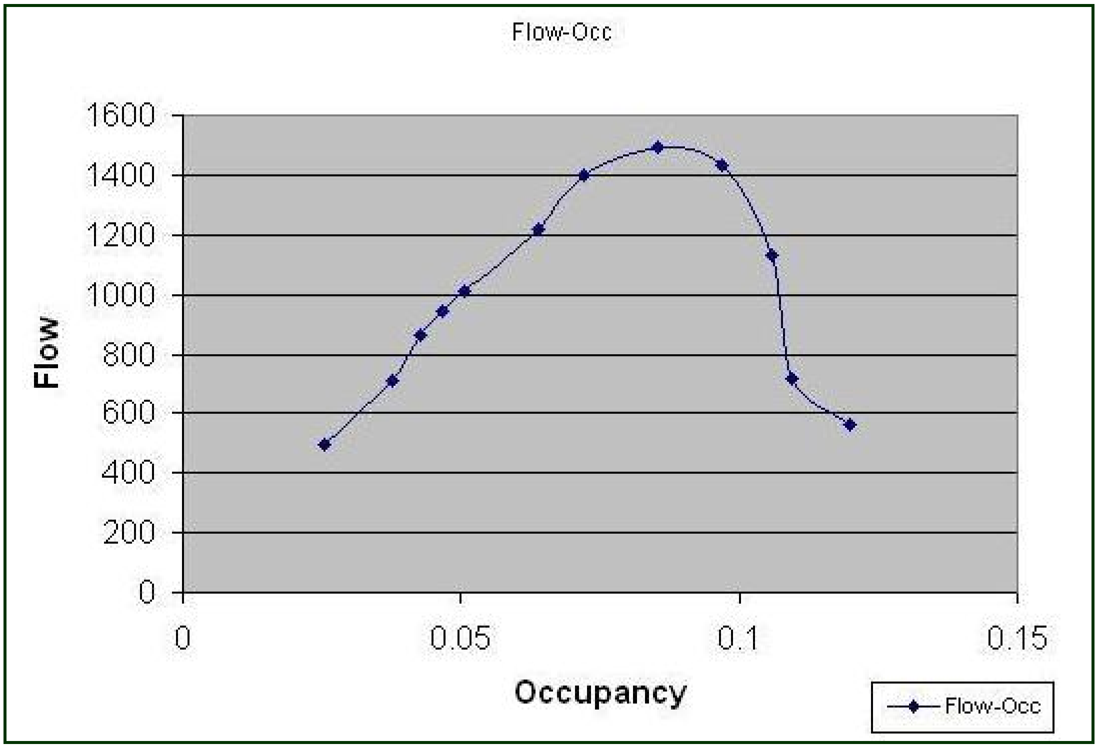


Figure 5.1 The Fundamental Diagram

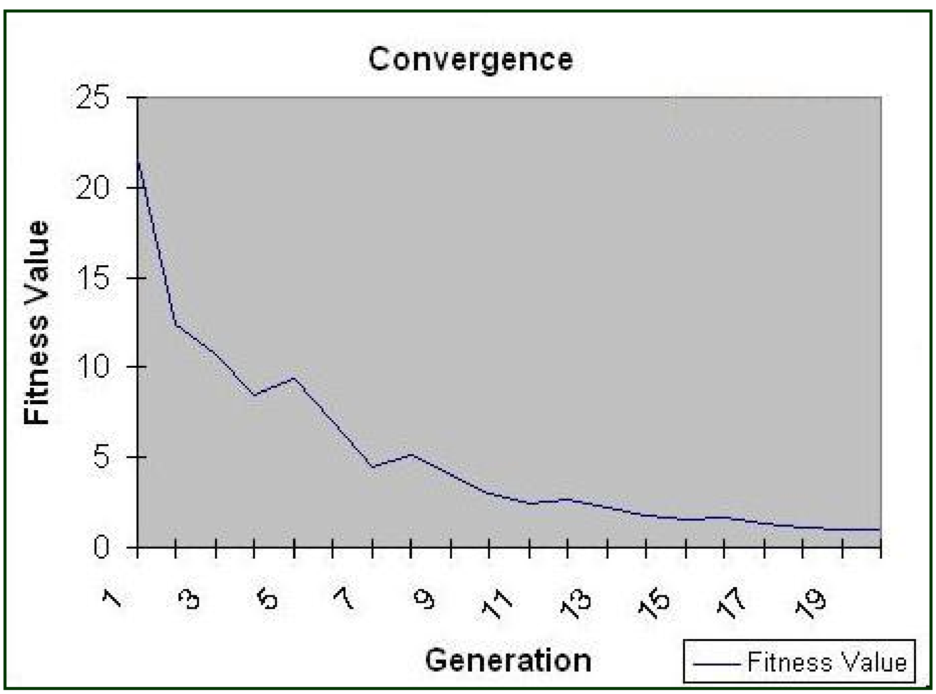


Figure 5.2 The convergence curve

Besides these features of the summary statistics tool, it also gives the final calibrated parameter values as well as their values in each generation. These generation-wise parameter values can be useful, when used in conjunction with the convergence curve, in detecting any discrepancy in the fitness value. Figure 5.3 gives a short view (in actual, it reports for user-defined number of generations and populations) of the generation-wise parameter values along with its fitness values. Using this spreadsheet, any abrupt change in fitness value, that might have affected the overall convergence curve, can be identified.

		Generation 1			
Index	Mean Target Headway	Mean Reaction Headway	Driver Aggressiveness	Driver Awareness	Fitness
Pop. 1	1.06421	0.707731	0.351635	0.79682	16.881
Pop. 2	1.77295	0.56865	0.464146	0.319832	23.2042
Pop. 3	1.96434	1.49305	0.517072	0.658566	24.9953
Pop. 4	2.02749	1.18518	0.29709	0.279328	28.2927
Pop. 5	1.69319	0.791132	0.673627	0.267987	21.6908
		Generation 2			
Index	Mean Target Headway	Mean Reaction Headway	Driver Aggressiveness	Driver Awareness	Fitness
Pop. 1	1.06421	0.704623	0.269427	0.274707	25.0482
Pop. 2	1.06421	0.707731	0.351635	0.79682	23.7482
Pop. 3	1.57113	0.725023	0.269427	0.274707	19.8693
Pop. 4	1.61973	1.03896	0.711911	0.485569	26.604
Pop. 5	0.876645	1.23586	0.29709	0.309331	24.7642
		Generation 3			
Index	Mean Target Headway	Mean Reaction Headway	Driver Aggressiveness	Driver Awareness	Fitness
Pop. 1	1.61489	0.515693	0.711911	0.49679	22.0427
Pop. 2	1.57113	1.03896	0.559316	0.350615	21.2435
Pop. 3	1.06775	1.0677	0.559316	0.220582	26.1432
Pop. 4	0.961444	0.59229	0.651545	0.350615	20.8891
Pop. 5	1.06421	0.704623	0.269427	0.319832	26.2895
		Generation 4			
Index	Mean Target Headway	Mean Reaction Headway	Driver Aggressiveness	Driver Awareness	Fitness
Pop. 1	1.06701	0.465581	0.227783	0.628683	21.1995
Pop. 2	1.62221	0.707695	0.559316	0.350615	24.2541
Pop. 3	0.786284	1.3077	0.711911	0.49679	22.9517
Pop. 4	1.61489	0.513581	0.227783	0.268587	21.8633
Pop. 5	1.61501	0.704623	0.651545	0.220582	23.2788
		Generation 5			
Index	Mean Target Headway	Mean Reaction Headway	Driver Aggressiveness	Driver Awareness	Fitness
Pop. 1	1.21118	1.18518	0.464146	0.316772	19.8557
Pop. 2	1.06421	0.707731	0.351635	0.748695	21.2635
Pop. 3	1.21112	1.03896	0.711911	0.448785	22.9751
Pop. 4	1.21112	1.03884	0.711911	0.448785	28.3798
Pop. 5	1.04495	1.28159	0.351635	0.79682	26.0088

Figure 5.3 Generation-wise results extracted from the final report

The summary statistics tool is not limited just for viewing the results. It also provides a better platform to identify any discrepancy, if at all, using the 'auto check' option or by manually observing the fitness values for each generation. Besides viewing the summary on the tool viewer there is an option of generating the report in a single click and presented in an excel file. All the different reports are saved on different sheets in one excel file for easy access.

The graphical user interface (GUI) of the tool is shown in the Figure 5.4, and Figure 5.5 shows the final report generated by the summary statistical tool in Microsoft Excel format.

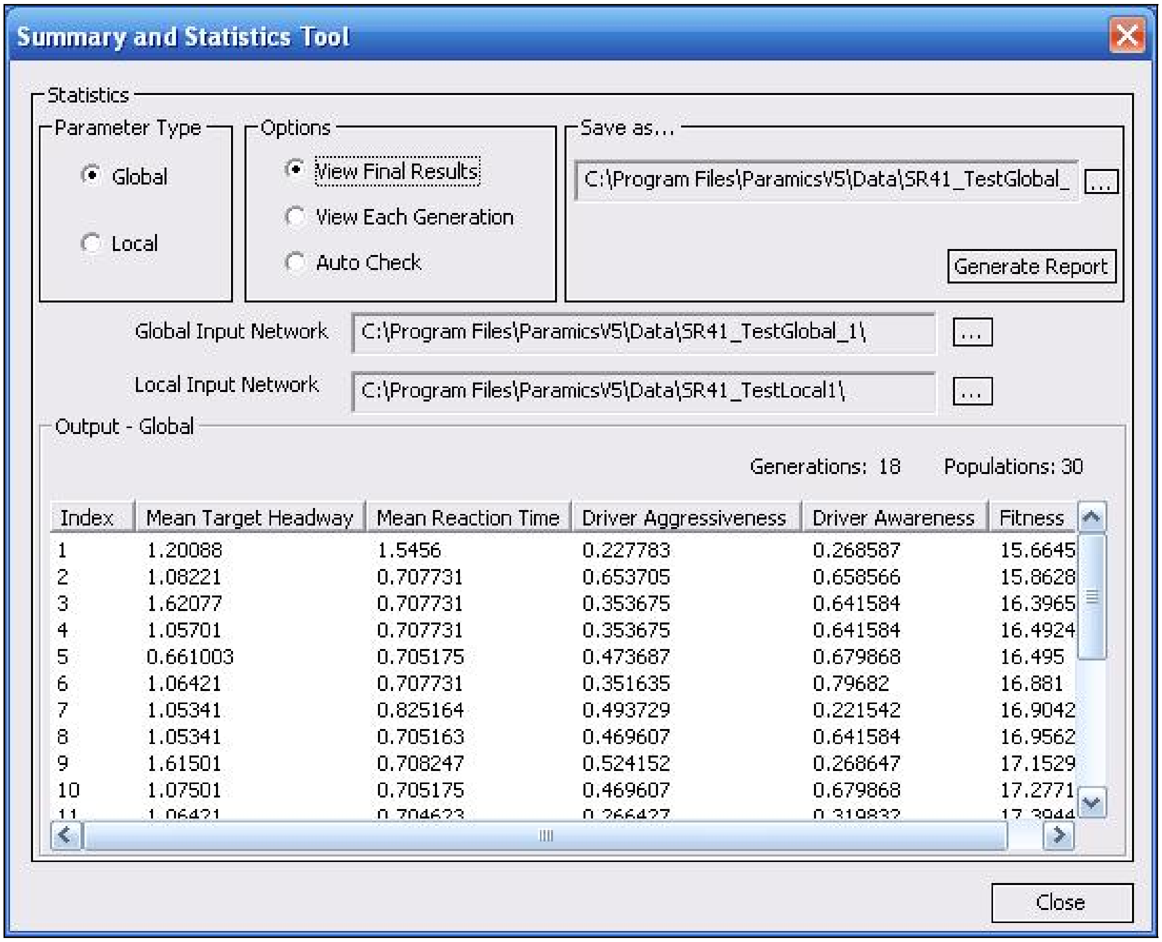
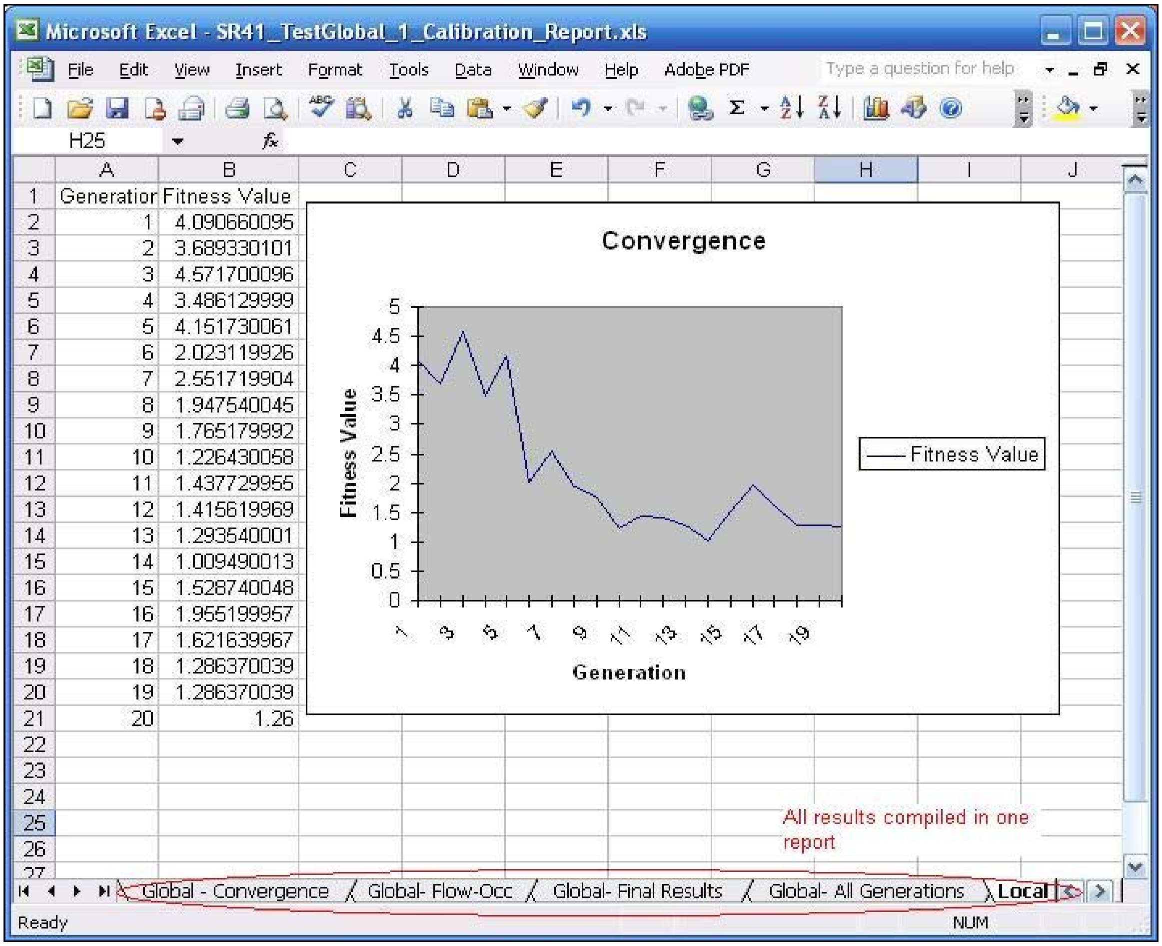


Figure 5.4 GUI for Summary Statistical Tool



Generation	Fitness Value
1	4.090660095
2	3.689330101
3	4.571700096
4	3.486129999
5	4.151730061
6	2.023119926
7	2.551719904
8	1.947540045
9	1.765179992
10	1.226430058
11	1.437729955
12	1.415619969
13	1.293540001
14	1.009490013
15	1.528740048
16	1.955199957
17	1.621639967
18	1.286370039
19	1.286370039
20	1.26

Figure 5.5 Final Report Generated by the Summary Statistical Tool

CHAPTER 6 CASE STUDY: CALIBRATION OF THE FRESNO SR-41 CORRIDOR NETWORK

6.1 Introduction

This chapter presents a case study of applying the developed calibration framework and tools to the State Route 41 corridor network in the city of Fresno, California.

Since the 1980s the Council of Fresno County Governments (COG) has developed and maintains a regional planning model, which consisted of the Fresno-Clovis metropolitan area, a portion of southern Madera County along SR 41, and a portion of northeastern Fresno County along Friant Expressway and SR 168 (Fresno COG 2006). In addition to traditional planning efforts, the local administration wants to study the multi-modal (e.g., HOV lanes) performance and traffic operations using micro simulation.

The data for the micro simulation study was collected and the original network was coded in Paramics by Fresno city engineers. The microscopic simulation model consists of the corridor of about 16 miles long in north-south and 4 miles wide in east-west. Two frontage arterials, Fresno Avenue and Blackstone Avenue are parallel to SR-41. Two other freeways, State Route 180 and State Route 99 are connecting with SR-41 in downtown Fresno (Figure 6.1). Because of the complexity of the network, an extensive data collection was conducted, including the traffic counts (freeway mainline, ramps and urban streets), traffic operational performance data (travel time on both SR-41 and two frontage roads and travel speeds).

Another research team conducted calibration on the SR-41 network when this project (TO5308) began, and the results were documented in (Liu et al. 2006). In that study, weekday PM peak of three hours (4pm-7pm) were selected, with a major focus on estimating a good traffic demand pattern. Global parameters, including time step, speed memory, target headway and reaction time, were selected for calibration and their combinations were tested to reproduce the flow capacity values for a certain location. A combination of time step of 5, speed memory of 8, target headway and reaction time of 0.75 second and 0.7 second, respectively, were picked through a trial-and-error process. A seed O-D matrix estimator was also developed to come up with a better seed O-D matrix to feed into Paramics' O-D Estimator; the estimated demand from the improved seed O-D matrix could reproduce closely the link counts obtained from field observations. Other traffic operation performance measures including travel time along SR-41 freeway mainline and speed contour maps showed a good match as well and thus the network was considered calibrated.

In our case study, the same network is used, but the network settings we start with were the original before the extensive calibration made in (Liu et al. 2006). The entire data set, including the initially coded network, traffic counts and travel time data, were generously shared with us by the Fresno city engineers.

This case study serves several purposes. The first one is to test the effectiveness of the developed tools within this project scope. Since other sites (e.g., SR-99 network) have also been tested within the scope, it would be insightful to compare the variation of the calibrated parameters against their geographical differences. This is especially useful to determine appropriate driving behavior parameters when such calibration data (e.g., traffic flow profile in time resolution of as fine as 30-second or 1-min intervals) are not available. The second one is to get a better sense of the potential difficulties one can encounter in calibrating a large network, and the third purpose is to get a sense of the amount of effort needed to calibrate a network of this size.

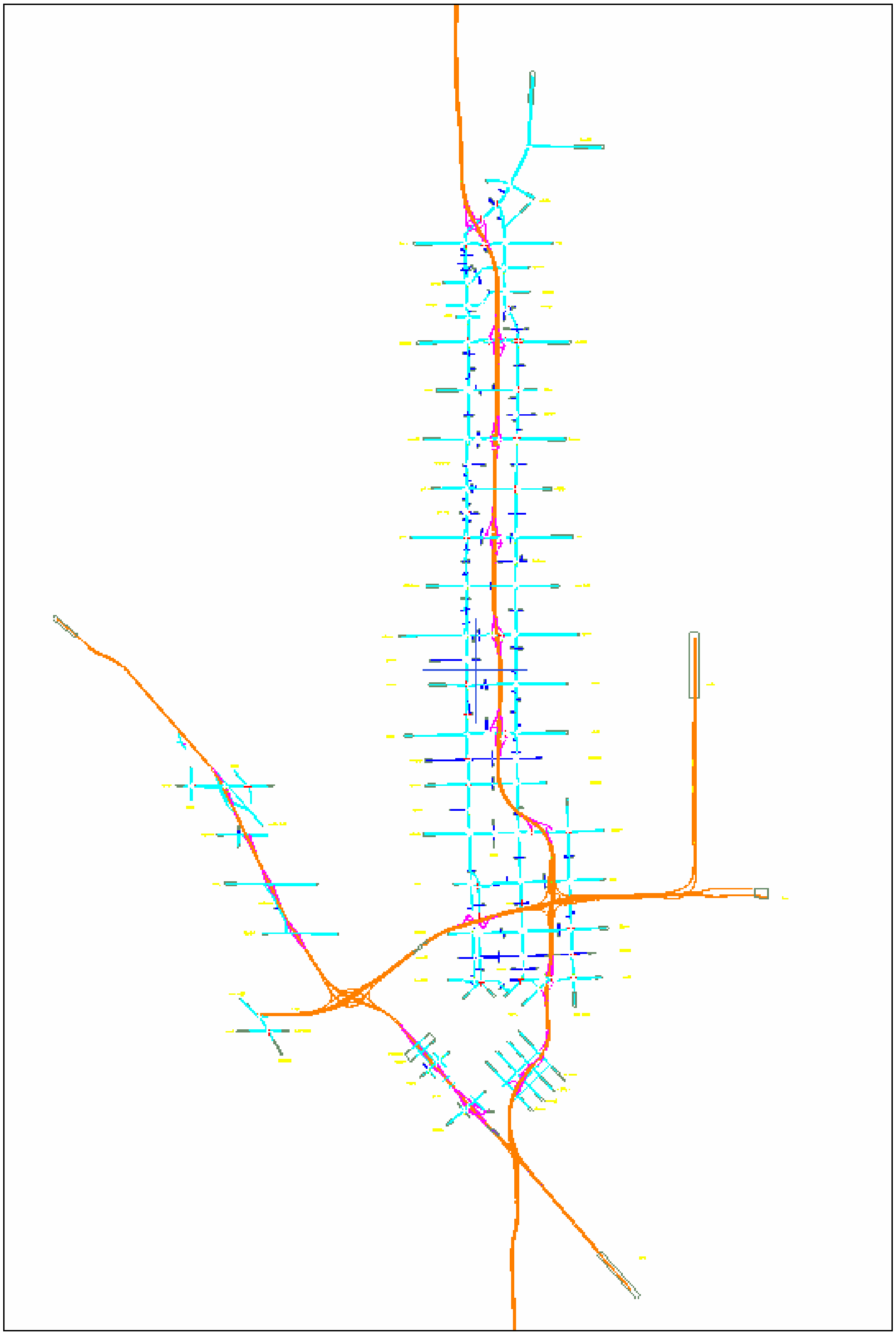


Figure 6.1 SR-41 Fresno Corridor Network Layout

6.2 Calibration of the Fresno SR-41 Network

The calibration procedure proposed in this research is followed to perform the following calibration tasks: network coding error checking, global and local parameter calibration, and demand refinement.

6.2.1 Obstacle #1: Traffic Gridlocks

Once the network data were received, a quick check revealed that the network was almost completely coded: all necessary files are complete and no connectivity problems were identified. But when running the model in Paramics Modeller with the coded network and given demand data, it was found that traffic gridlocks formed in several interchange areas along SR-41 and these local gridlocks quickly propagated upstream of the congested areas. After about 55 minutes of simulation time (the simulation starts at 3:30pm with a warm-up period of 30 minutes), gridlocks spread over the entire network and very few vehicles could move because of the severe congestion (Figure 6.2; circles mean "hot spots", i.e., long traffic queues at those sites).

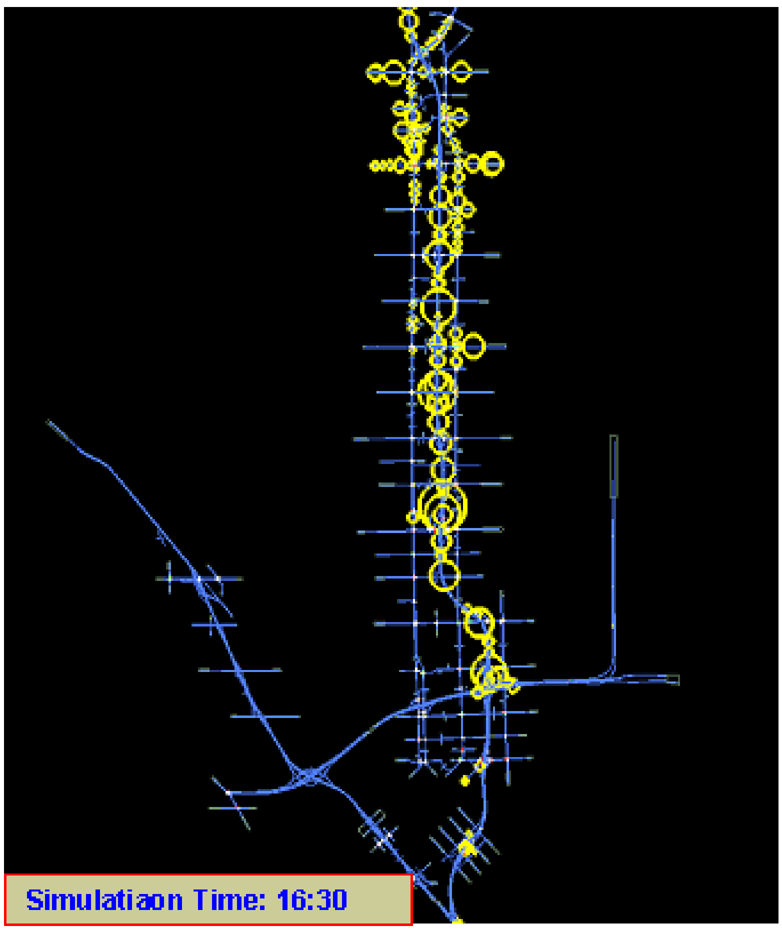


Figure 6.2 Network Wide Traffic Gridlock in Simulation with Original Network Settings

To find out if this is also the case in real life for that network, we checked with the traffic data, e.g., traffic flow and speed, from the PeMS database, and found that this network did not have total gridlock except for only a few interchange areas suffering from moderate congestion (e.g., SR-41 and SR-180 system interchange). Moreover, the congestion does not block traffic in up- and downstream interchange areas (Figure 6.4 & Figure 6.5).

Table 6.1 PeMS Detector and Interchange locations on SR-41

Postmile (Abs)	Postmile (CA)	VDS ID	Location	Description	Interchange
125.153	23.4	600064	Huntington	ML 3 Lanes	After
125.953	24.2	600065	Belmont	ML 3 Lanes	After
126.653	24.9	600232	Floradora	ML 3 Lanes	Before
127.653	25.9	600233	Clinton	ML 3 Lanes	After
128.653	26.9	600054	Dakota	ML 3 Lanes	Before
129.653	27.9	600053	Gettysburg	ML 3 Lanes	After
130.653	28.9	600231	Barstow	ML 3 Lanes	Before

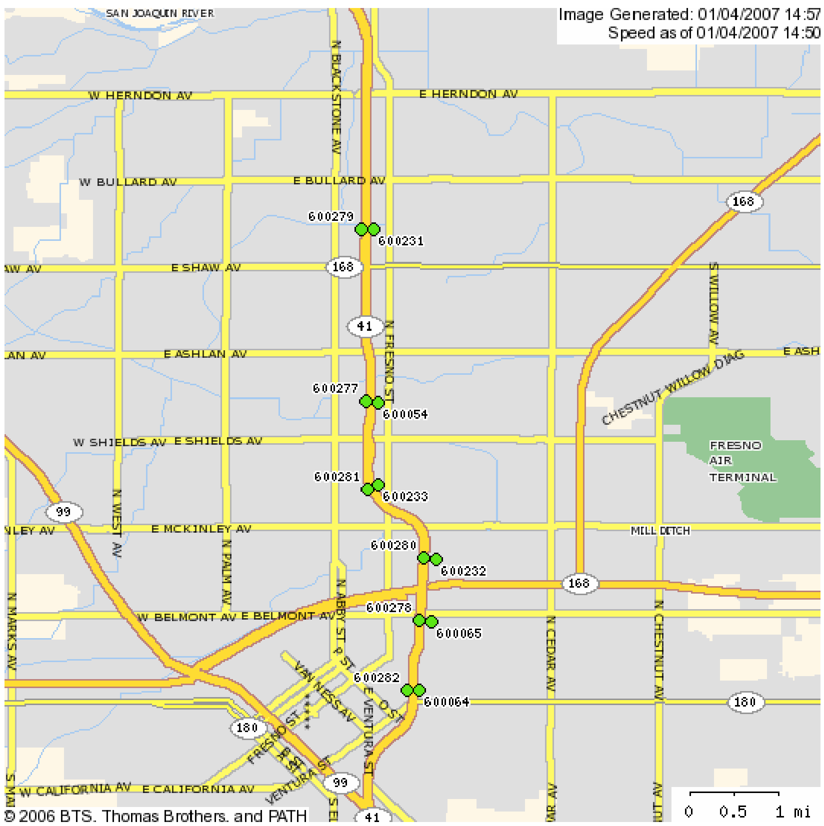


Figure 6.3 PeMS Detector Locations

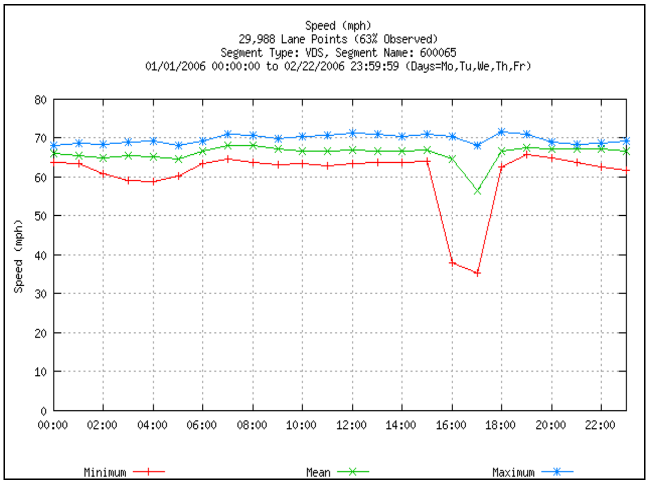


Figure 6.4(a) SR41 Mainline Speed Profile(VDS600065)

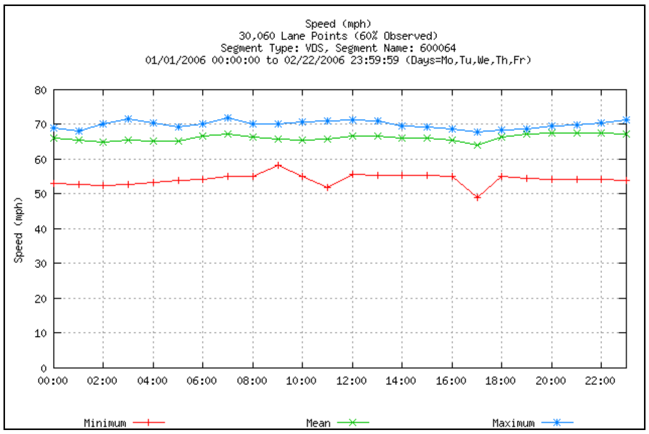


Figure 6.4(b) SR41 Mainline Speed Profile (VDS600064)

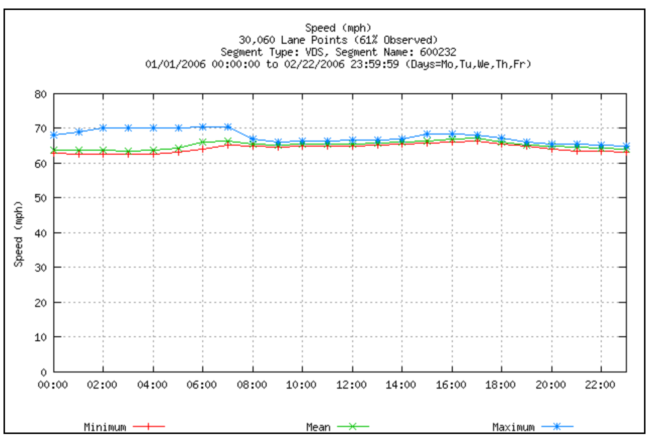


Figure 6.4(c) SR41 Mainline Speed Profile (VDS600232)

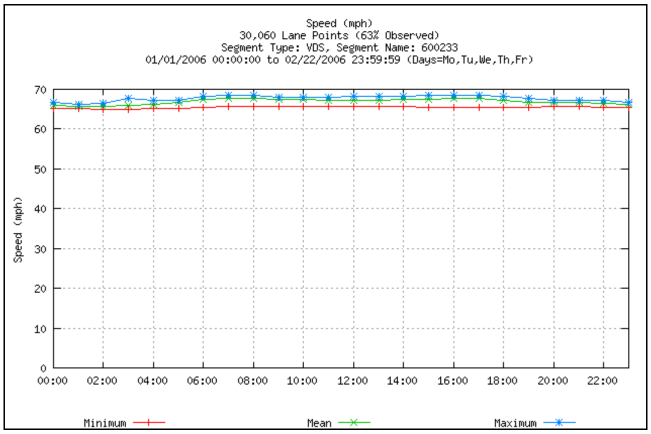


Figure 6.4(d) SR41 Mainline Speed Profile (VDS600233)

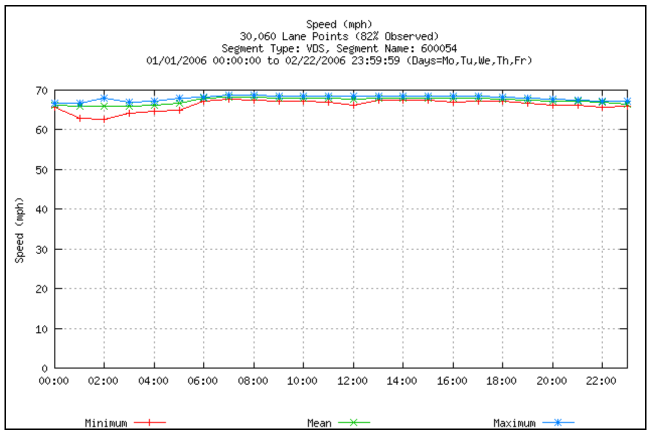


Figure 6.4(e) SR41 Mainline Speed Profile (VDS600054)

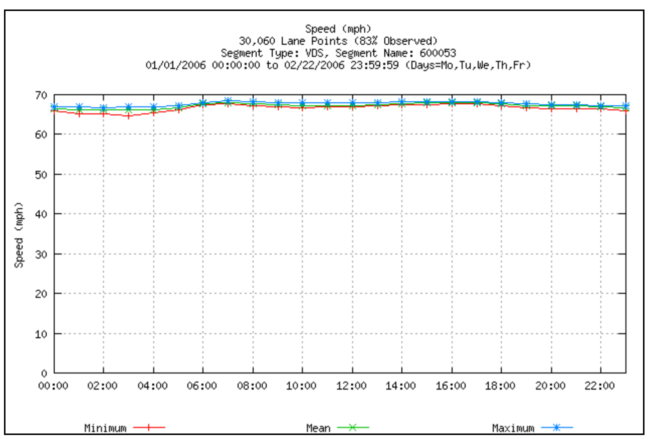


Figure 6.4(f) SR41 Mainline Speed Profile (VDS600053)

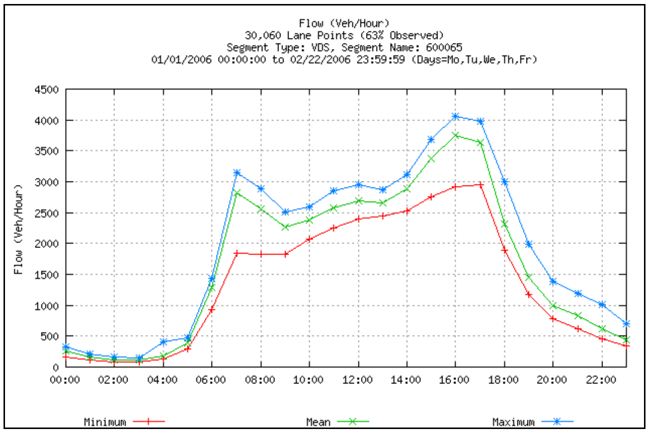


Figure 6.5(a) SR41 Mainline Flow Profile (VDS600065)

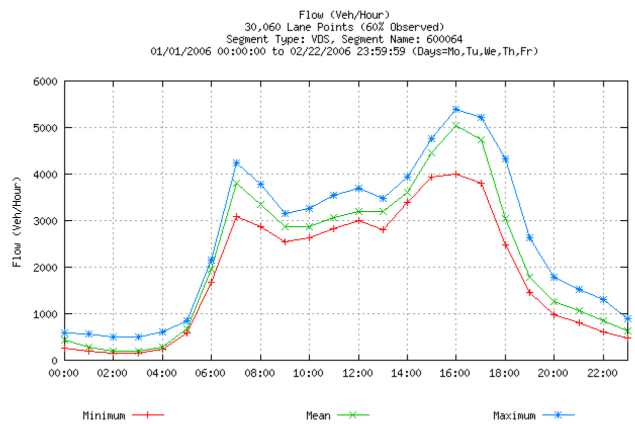


Figure 6.5(b) SR41 Mainline Flow Profile (VDS600064)

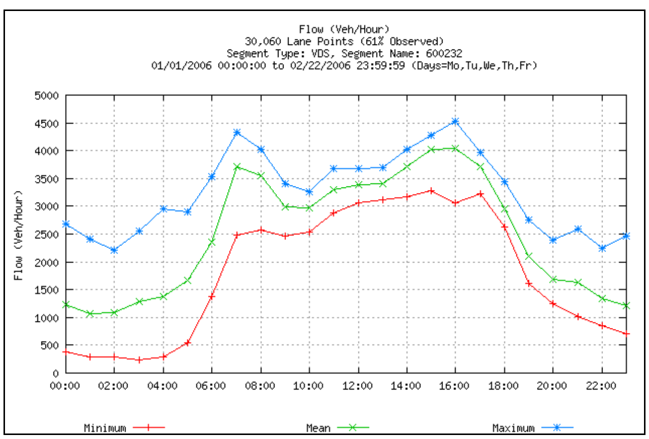


Figure 6.5(c) SR41 Mainline Flow Profile (VDS600232)

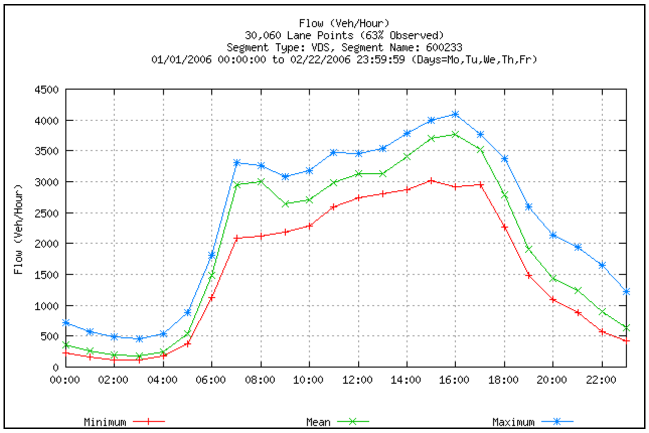


Figure 6.5(d) SR41 Mainline Flow Profile (VDS600233)

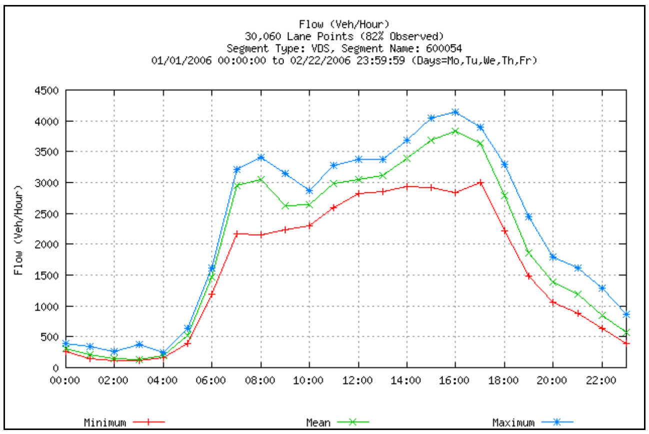


Figure 6.5(e) SR41 Mainline Flow Profile (VDS600054)

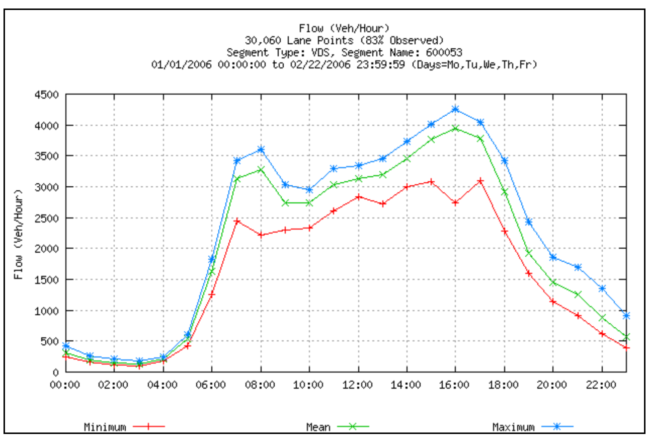


Figure 6.5(f) SR41 Mainline Flow Profile (VDS600053)

Figure 6.4 and Figure 6.5 clearly show that there is no major network wide gridlock during a PM peak. Consequently, the first calibration step is to eradicate the gridlocks in the Paramics network. After careful examinations, the following problems are identified that are considered the possible causes of the network wide gridlocks:

- *Unusually high U-turn volume at some turning bays*. It is observed that at both frontage roads, Blackstone Avenue and Fresno Avenue, too high traffic volumes for the U-turns are present at some left-turning bays to be accommodated by the given green time. Since both left-turn and U-turn traffic shares the same turning bays and the same signal phase, the mixed traffic could easily fill the storage space and the queue formed spread upstream when its allocated green time is not long enough. Successively, the through and right turning traffic upstream of the left-turn bay were unable to be serviced due to the blocking by left- and U-turn traffic, and local gridlocks occurred and spread.

- *Traffic blocking at nodes*. Paramics provides a function of "Non-blocking compliance rate" to specify the proportion of the vehicles that would not enter the conflict zone (i.e., the commonly used area to perform the turns at an intersection) of a signalized intersection when this behavior is generally prohibited (Quadstone Ltd. 2004b). However, it is found that even if this rate is set to 100%, ie., not occupied by stopped vehicles, some vehicles are still observed enter the conflict zone when the downstream link has queues (Figure 6.6). These stopped vehicles block the cross traffic. Such intersection blocking wastes green time for the cross traffic and triggers local congestion and gridlock.

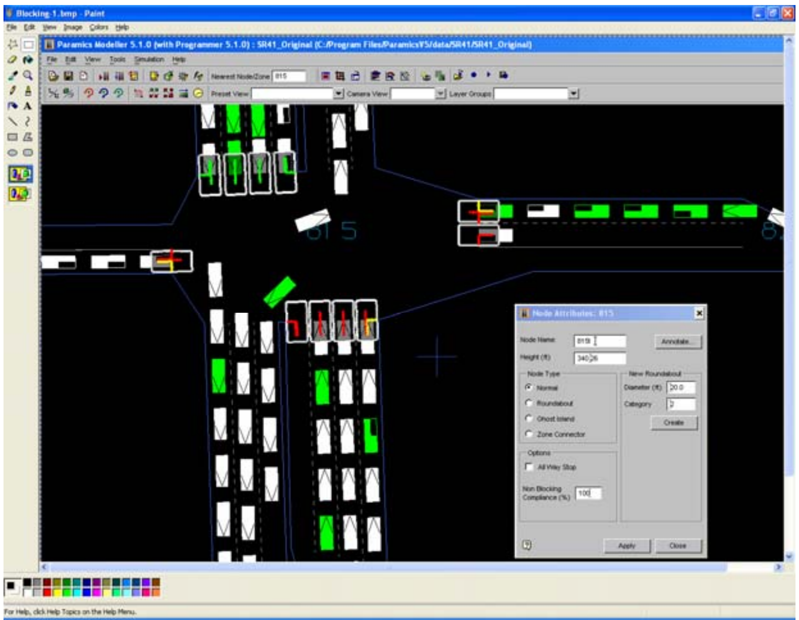


Figure 6.6 Traffic Blocking at Signalized Intersections

- *Improper lane utilization settings at signal approaches.* Improper lane utilization was also observed at some critical intersections, e.g., the intersection of Fresno Avenue at Heardon Avenue (Figure 6.7). At these places in the original network, traffic would either favor a particular lane for certain turning movements or perform the lane changing maneuvers too late to be able to enter the specified lanes. Consequently, the green time as well as the storage capacity of the lanes for certain movements are underutilized. The underutilization could easily lead to queue spilling back to other streets and creating the "chain reaction" at both urban streets and freeway mainline (blocked by off-ramp traffic).

- *High demand.* The original network has a total of fourteen periods in 15-minute intervals of demand data, corresponding to a 3.5 hour simulation. Generally, if properly calibrated, these successive 15-minute periods of demand can capture the departure time choice behavior of the drivers. However, it is observed that even if there are no local gridlocks at some places (the downtown portion of SR-41 corridor), congestion still occurs on the freeway. This implies that the initial demand could be overestimated.

Some of these problems are caused by the modeling limitations in Paramics (e.g., intersection blocking), but most of them are due to improper network settings and limited calibration. Therefore, we proceed to deal with the identified problems one by one.

Figure 6.7 Improper Lane Utilization under the Original Network Settings (Fresno Avenue @ Heardon Ave.)

6.2.2 Error Checking and Network Fine-tuning

As identified in the previous section, some network settings were not properly coded and led to or aggravated the traffic gridlock problem. Our first calibration effort is to check for coding errors and fine-tune the network settings at critical locations.

6.2.2.1 Elimination of local blockings and spillback

A major mean to check for possible coding errors is to visually monitor the simulation runs to identify the locations that have serious local congestion. The simulation is watched in Paramics' Modeller continuously until persistent queues occur at some sites (using the "Hot Spot" function in Paramics' Modeller). Those sites whose congestion level does not match what was observed in the field need further diagnosis.

First, the intersections with exceptionally high U-turn traffic volumes are examined one by one, including Fresno Ave NB@Ashlan, Blackstone@Bullard and a few other intersections adjacent to these two. At those intersections, all signals are operated in the vehicle actuated mode. Their timing parameters were all checked and found to be consistent with the signal plan data provided to us.

Exceptionally high U-turn traffic volumes at some locations imply that trips were disproportionately assigned to those paths that go through these locations, and because all-or-nothing and/or stochastic traffic assignment methods were used in the original simulation setting, no vehicle re-routing to avoid the long queues at these locations occurred during the simulation.

We adjusted the trip rates on those paths in the later demand estimation process to reduce the number of U-turns on those paths. However, no field trip could be made to validate if the number of U-turn trips observed in the simulation were actually close to that observed in the field.

Second, links with poor lane utilization are identified and corrected for imbalanced lane utilization. To do this, we first watch the simulation and identify those places where traffic congestion builds up quickly. We then investigate if the built-up of congestion is caused by poor lane utilization, e.g., some lanes in a lane group are used heavily while others are not. If this is true, we fine tune the relevant model parameters to achieve a balanced lane distribution. Besides sign-posting designation that can only affect vehicles' lane choice within the same link, Paramics provides two more functions to fine tune how vehicles make use of the lanes within certain sections: *lane choices* and *next lanes* (Quadstone Ltd. 2004c). *Lane choices* let the modeler specify the associated lanes for certain turning vehciles to take in a chain of links when they enter the first link in the chain. This a priori knowledge of lane usage allows the vehicles to change lanes more realistically, especially when the concatenating links are short and sign-posting alone cannot provide realistic lane-changing maneuvers. *Next Lanes* let the modeler specify for a vehicle the lane to take in the downstream link. Users are referred to (Qadstone Ltd. 2004c) for the details of how to apply *lane choices* and *next lanes* to to affect lane distribution.

The most critical intersections and links that we found to have lane utilization problems include

- Herdon Ave @ Fresno Ave concatenated with Herdon Ave @ Blackstone Ave
- Bullard Ave @ Fresno Ave
- Shaw Ave @ Fresno Ave
- SR41 diamond interchange @ Ashlan Ave
- SR41 NB freeway mainline sections leading to the off-ramp connecting to the above streets.

We applied the *lane choices* and *next lanes* functions to adjust the imbalanced lane distributions at those sites. It should be noted that these functions have a few parameters of their own to be calibrated. They include vehicle types and lane usage proportions. Their calibration, however, was done on a trial-and-error basis and requires good engineering judgment.

6.2.2.2 Applying Various Traffic Assignment Methods

Another way to affect the flow distribution in the network is trying various traffic assignment methods. Paramics offers three basic traffic assignment methods: all-or-nothing, stochastic or dynamic feedback assignment (see Paramics' Modeller Manual). One can use any of the three methods or combinations of them in assigning traffic to the network. In Particular, stochastic and dynamic feedback assignment methods allow the simulated traffic between an O-D pair to take more than one route or even change their routes during their journey (dynamic feedback). It should be noted that in Paramics only familiar drivers can choose a route other than the static shortest path or switch to another path en route when dynamic feedback assignment is used. In our developed tool set, the D-R choice module can calibrate the corresponding route chouce parameters.

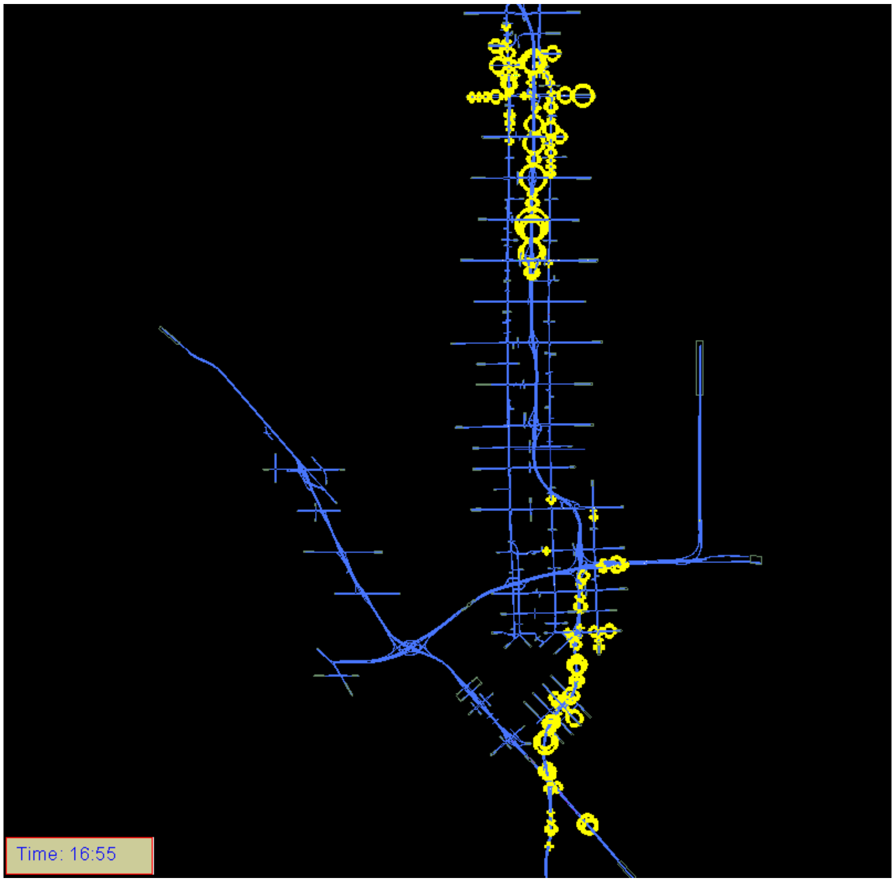


Figure 6.8 Network Gridlock Pattern under Dynamic Feedback Assignment

We applied the dynamic feedback assignment method in the simulation, but our simulation shows it does not completely eliminate gridlock. Rather, it postpones the formation of gridlock. As shown in Figure 6.8, gridlock occurs about one hour after the simulation starts. It is also noticed that with dynamic assignment, the locations with heavy congestion differ from those without dynamic assignment. Clearly traffic assignment has significant impact on the resulting traffic congestion pattern in the network, but before attempting a calibration with different assignment methods, we should first make sure that the local and global driving parameters be calibrated, as suggested by our calibration procedure. These are done in the next section.

6.2.3 Global and Local Parameter Calibration

We applied our local and global parameter calibration tools covered in Part II of this report series to several critical sites in the network. The calibration results are listed as follows:

Test site #1

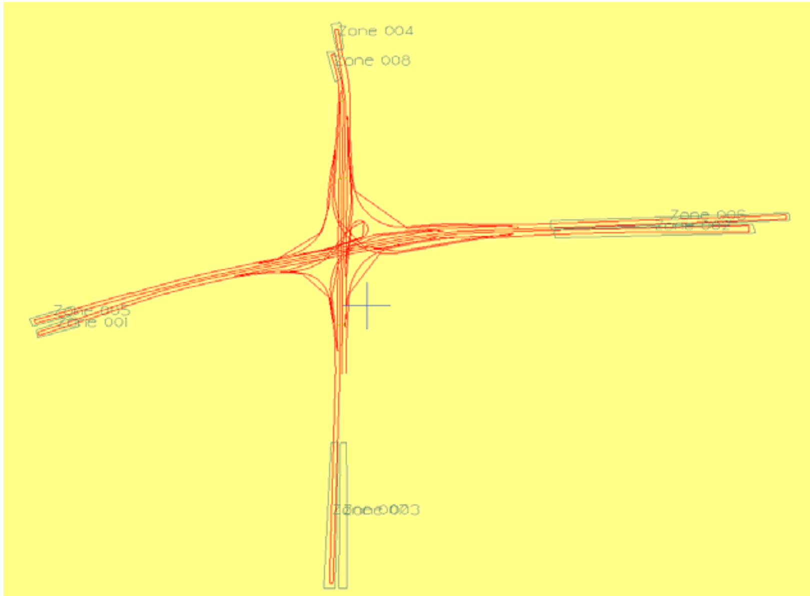


Figure 6.9 Sub-network for Test Site #1 (SR-180/SR-41 Interchange)

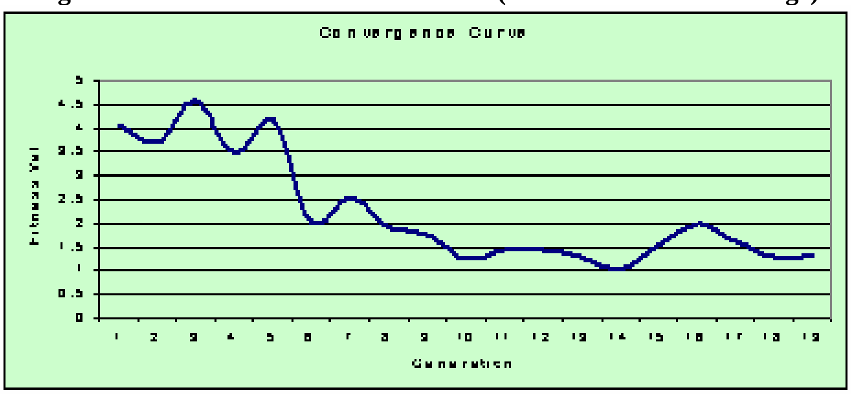


Figure 6.10 Convergence Curve for Test Site #1

Table 6.2 Calibrated parameters (Site #1)

	Bottleneck (mainline south of the interchange)		Bottleneck (mainline north of the interchange)	
	Best	Second Best	Best	Second Best
Link Headway Factor	0.92	1.35	0.96	1.07
Link Reaction Factor	0.58	1.35	1.79	1.18
Sign-posting (ft)	2634.4	3474.3	1493.9	3435.2

Test site #2

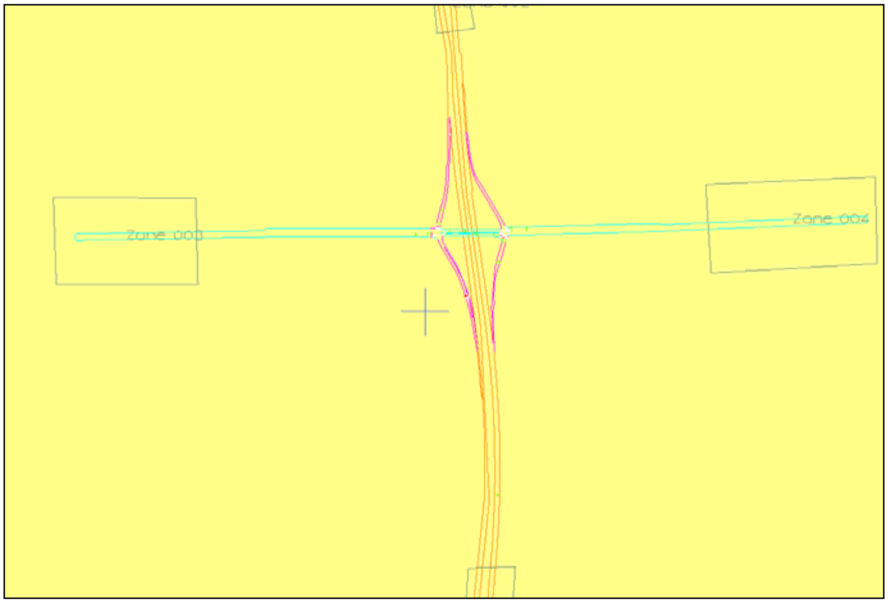


Figure 6.11 SR41@Ashlan (PeMS VDS 600054)

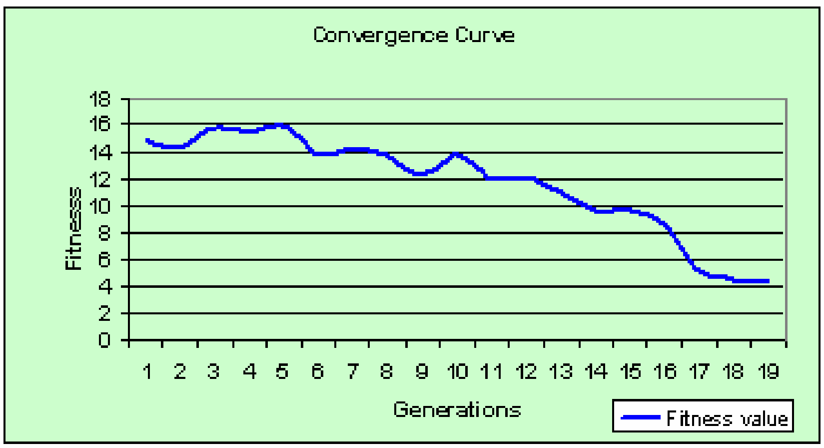


Figure 6.12 Convergence Curve for Test site #2

Table 6.3 Calibrated parameters (Site #2)

	Ramp (mainline north of the intersection)	
	Best	Second Best
LHF	1.57	1.31
LRF	0.752	.512
Sign-posting (ft)	1129	1130.43
Ramp HF	1.96	1.96
Min Ramp Time (sec)	2.40	2.35

These parameter values were used in the subsequent calibration of other model parameters in the simulation.

6.2.4 Calibration of Traffic Control Parameters

The network has 93 vehicle actuated signals and 23 ramp meters. Using the API developed for Paramics V5 (Liu, Chu & Recker 2001) and additional functional enhancements to the API (see Chapter 4 of this report), all control settings were specified to match timing data provided in the raw data set. It is found that the controls function properly. The remaining calibration task is therefore the refinement the O-D demand for the entire network.

6.2.5 Traffic Demand Refinement

The SR-41 network, considered a large scale or *strategic* network, takes a considerable amount of CPU time to simulate traffic for a 3-hour PM peak. Consequently, it will take Paramics' O-D Estimator even more time to refine the demand input because this estimation process requires numerous simulation runs. Since this network has 14 periods of demand to be estimated, the O-D estimation process becomes a huge burden in the calibration if we solely rely on the O-D Estimator to perform the demand estimation task. Fortunately, we have developed the LPFE based ODE (O-D estimator), which takes much less time to obtain an O-D trip table. We therefore first run LPFE, then use the O-D tables produced by LPFE as the seed O-D tables for Paramics's O-D Estimator, and use it to further refine the estimated OD trip tables.

6.2.5.1 Data Preparation

We explored the data provided to us by Caltrans District 6 engineers and found a total of 501 entries of traffic counts, including freeway mainline and ramp counts and intersection turn counts. These data were assembled and used in our O-D estimation effort. The assembled turn counts and link counts are shown in Figure 6.13. Because the effective data cover only a two-hour period (4:00pm-6:00pm), O-D trip tables were estimated for this period only.

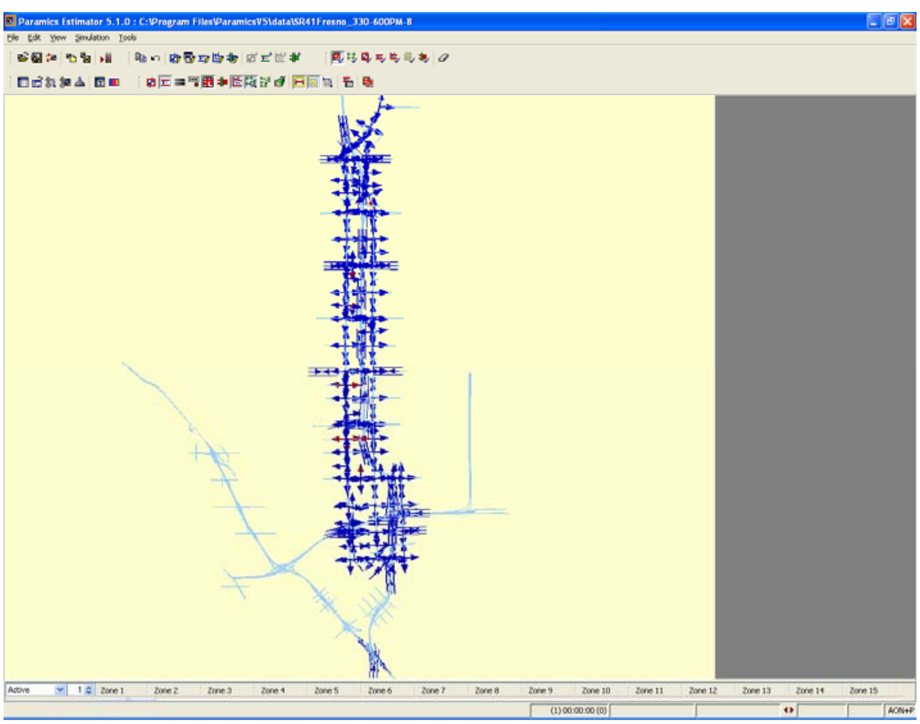


Figure 6.13 Assembled Link and Turn Counts locations for the SR41 Network

6.2.5.2 Obtaining the Seed O-D with the LPFE ODE Tool

Following the procedure described in Chapter 3 of this report, the seed O-D trip tables are estimated using the LPFE O-D estimator. It took LPFE a few seconds to compute the eight periods of O-D trip tables (4:00pm-4:15pm, 4:15pm-4:30pm,…, 5:45pm-6:00pm) from the assembled flow counts.

LPFE also provides count comparisons to evaluate the quality of the estimated O-D trip tables. Figure 6.14 shows the GEH values and the corresponding number of links in each category of GEH values. From this figure we can tell that the estimated O-D trip tables, when assigned to the network following LPFE's macroscopic model, produced link counts that match the observed ones well (The GEH values of 88% of all links with traffic counts are under the acceptable level of 5).

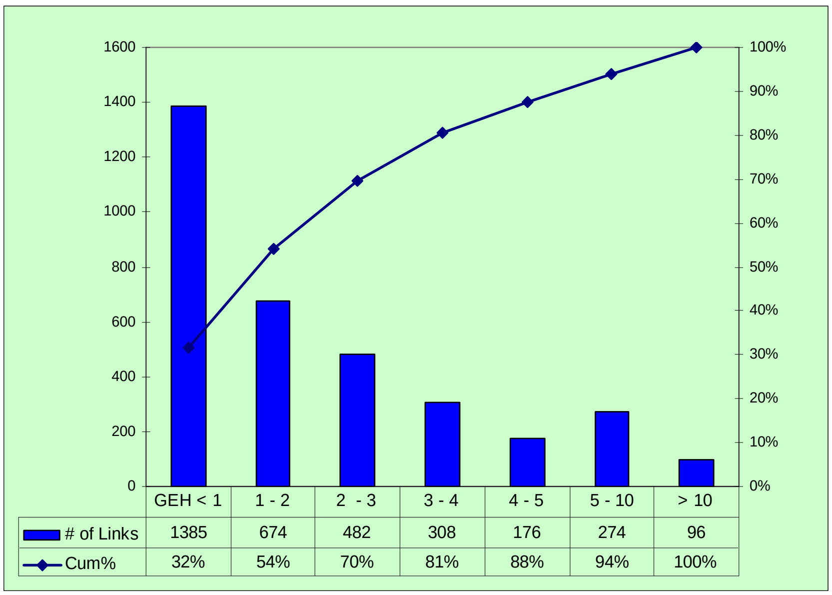


Figure 6.14 Quality of the Estimated O-D Trip Tables

In our normal O-D estimation procedure, we'll use the O-D trip tables produced by LPFE as seed O-D matrices to feed Paramic's own O-D estimator to obtain more refined O-D trip tables, but this was not completed due to a software problem between Paramics' O-D Estimator and our developed APIs[1]. We consulted the software developers (Quadstone) and were told that this software bug would be fixed in the next release of their software. Thus we did not perform further adjustments to the estimated O-D trip tables and used them in our subsequent simulations.

Simulation runs were conducted with a 15-minute warm-up period at the start and a 15-minute cool-down period in the end of the simulation. With Paramics' Analyzer, link volumes in 15-minute intervals were extracted from the simulation result for those links with field traffic counts, and the two sets of data are compared. The results are shown in Figure 6.15. This figure indicates that about 55% of the links with traffic counts has an acceptable GEH value (<5), which compares favorably with a prior calibration effort (Liu et al 2006), where 50% the links with traffic counts had a GEH <5. It should be noted, however, the match between simulated and observed traffic counts in the last two periods (5:30pm-5:45pm, 5:45pm-6:00pm) is not as close

[1] Paramics' O-D Estimator crashes when loaded with the control plug-ins (vehicle actuated control and freeway ramp metering).

as in the first six periods, owing to unexpected congestion at some locations. Were Paramics' O-D estimator not clash with the signal API, the O-D trip tables could be further improved to reduce mismatch between observed and simulated link volumes in the last two simulation periods.

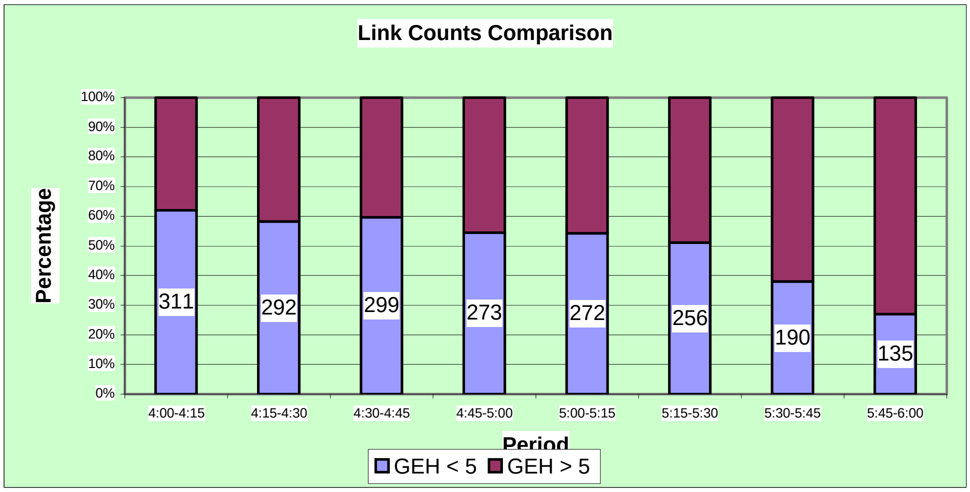


Figure 6.15 Link Counts Comparison under Pattern Matrix

6.3 Summary

Calibration of micro simulation models is both challenging and time-consuming when the network under calibration is as large as the SR-41 network because of the numerous parameters involved and their complex relations, and such an endeavor can easily take an experienced team of micro simulation users several months or more to complete. In this case study, we followed our five-step, divide-and-conquer strategy and the following calibration tasks: error-checking, global and local parameter calibration and O-D estimation. With the help of our developed tools, we were able to calibrate the SR-41 network within a reasonable amount of time (one person month). Here are some lessons that we learned in this case study:

Check for problematic areas, find the problem causes and study them one by one. In micro simulation every modeling detail matters. A minor problem at a critical location could trigger a gridlock of a large network area or even the entire network. For example, the initial improper lane-utilization settings at several critical locations (e.g., Herdon @ Fresno and Herdon @ Blackstone) in our case study caused a domino-like congestion buildup. Other factors, such as intersection blocking and high turning volumes, exacerbated the situation. Some of these problems can be solved locally (e.g., lane utilization and intersection blocking), but others have to be solved globally through route choice and demand refinement (e.g., high turning volumes at some locations). Nevertheless, each factor, starting from local to global ones, must be examined one by one as proposed in the calibration framework to remove the problems it causes.

Local driving parameters should be calibrated for each critical site to ensure that the simulated driving behavior reflect reality. When we examined the results of local parameter calibration from both the SR-41 network and the SR-99 network, we concluded that the calibrated results are reliable and could be verified using different data set. The calibrated parameters significantly improve the modeling accuracy at those critical sites such as weaving sections, interchange areas and lane drops. These results also show that the parameter values from different sites can be quite different even though the network characteristics of these sites are similar. Therefore, if time and data permit, one should calibrate the local parameters for each critical site..

Global calibration plays a major role in improving the micro simulation results. Global calibration refers to the calibration of route choice parameters and demand estimation, two sets of the parameters that affect the distribution of traffic flow in the network. In our case study, the use of a macroscopic demand estimation tool, LPFE, prove to be quite useful, for it helped us to obtain a good O-D trip table much more quickly than using Paramics' own O-D estimator. In fact, Paramics' own O-D estimator failed to run because of a software conflict with the signal timing API for the SR-41 network. With the O-D trip table estimated by LPFE, we were able to match 88% of the traffic counts with an acceptable GEH (<5) under LPFE's macro traffic model, and 55% of the traffic counts with an acceptable GEH (<5) under Paramics' micro simulation, both were better than what were achieved in a previous study for the same network. Still, it is desirable to combine LPFE and Paramics' own O-D estimator whenever this is feasible, for this can cut down estimation time (through LPFE) and further improve estimation reliability (through Paramics' own O-D estimator).

Overall, the case study indicates that our calibration framework and tools developed in this project can considerably ease the calibration effort, improve calibration quality, and shorten calibration time. Nevertheless, calibrating a micro simulation, particularly when the network is large, is no easy task and cannot be automated. Good engineering judgment, sound knowledge of the traffic characteristics of the study network, and adequate, reliable input data are indispensable in a good calibration work.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

The main objective of TO 5308 is to develop a systematic calibration procedure and the supporting tools to assist and streamline the calibration of microscopic traffic simulation models. We began the work by having numerous discussions with experienced microscopic simulation users, and conducting an extensive literature review of the micro simulation applications and their calibration efforts. The results are documented in Part I of this report series. In that report, a wide range of transportation applications of micro simulation is reviewed, paying particularly attention to their calibration efforts. The review revealed that many studies paid little attention to model calibration, or at least not documented it if a detailed calibration effort was carried out. And among the studies that performed model calibration, the calibration procedures employed are quite ad hoc, often relying on trial-and-error. The report identified three calibration issues to be addressed in future work: the lack of a systematic calibration procedure, the lack of automated calibration tools, and the lack of reliable O-D trip demands. In that report, four important aspects of model development, namely project planning, network coding, data collection and processing, and traffic demand estimation were examined and guidelines for project planning, network coding, and data collection and preparation, synthesized from various sources including our own calibration experiences, were provided .

We then developed a systematic calibration framework that decouples the calibration process into five components: project scoping and error checking, global parameter calibration, local parameter calibration, departure/route (D-R) choice parameter calibration, and global calibration. In implementing this framework, we developed GA-based calibration tools for local and global parameters, as well as D-R choice parameters, and integrated the developed tools into one user-friendly graphical interface. This calibration framework, together with the developed tools, was tested on synthetic and real networks. It is shown by working examples that the developed tools can help achieve satisfactory calibration results with much less human intervention as found in the commonly practiced trial-and-error calibration procedure. These results are documented in Part II of this report series.

In the second phase of this research project, we continued our development of calibration tools using various heuristic optimization techniques. In Chapter 2 of this report, we implemented another promising optimization algorithm- simultaneous perturbation stochastic approximation, to calibrate driving behavior model parameters and compared it with others. Our comparison indicates that this method can generally obtain an acceptable set of parameters in much less time than the GA algorithm. Although SPSA speeds up the calibration process considerably compared with the GA method, our experiments indicate that GA is more capable of reaching finer solutions. We thus recommend that when the number of parameters to be calibrated is not large, and the network concerned is small, GA should be used. When either the network or the number of parameters to be calibrated is large, SPSA should be used. This is a trade-off between solution accuracy and solution speed.

In Chapter 3, we developed an O-D demand estimation tool based on the logit path flow estimator (LPFE). Demand estimation is a vital step in the calibration of a micro simulation and various micro simulation packages, such as Paramics, offer their own O-D estimation tools. In the case of Paramics, it was found that the estimation of O-D demand was often slow if one started with a poor initial seed O-D matrix. The implemented O-D estimation tool makes use of the results from a previous PATH project (TO 5502: Development of a Path Flow Estimator for Deriving Steady-State and Time-Dependent Origin-Destination Trip Tables), the time-dependent logit path flow estimator, to produce a good seed matrix quickly. In order to achieve this, a network conversion tool is developed to convert Paramics's detailed network settings to those of LPFE and vice versa. It was shown that with the seed O-D provided by LPFE, Paramics' O-D Estimator can usually find a good O-D demand pattern (one that produces the closest match to observed traffic counts and/or travel times) much more quickly.

In Chapter 4, the effort to enhance the traffic control functions of Paramics through API development is described. Based on previous efforts (e.g., Liu, Chu& Recker, 2001), an actuated signal control plugin was developed to enhance the actuated signal control capabilities of the Paramics software. These include adding a volume-density control function, detector placement and output diagnoses, and a conversion of SEPAC coordination data to a format acceptable by Caltrans C8's logic. We also provide a detailed, step by step procedure to code, diagnose and optimize arterial traffic signals in Paramics. These can all be used together to ease the effort of calibrating traffic signal operations and improve simulation performance.

Chapter 5 describes a summary statistics tool developed to help monitor, diagnose, and report on the calibration of local and global driving behavior parameters. This tool produces the fundamental diagram from simulated data for selected locations, so that one can judge if the capacity and shape of the fundamental diagram of any selected location is reproduced by the simulation. A good match indicates an satisfactory calibration of local and/or global driving behavior parameters. It also generates a convergence curve from which one can judge if the calibration is progressing well and when to terminate it. Besides these visual aids, the summary statistics tool also allows users to save intermediate and final calibration results in a Microsoft Excel file for later analysis.

Finally, in Chapter 6, we conducted a case study with a large size network, the State Route 41 network in Fresno, California, to evaluate the effectiveness of the developed calibration framework and the corresponding calibration tools. Following our five-step procedure, we first performed error-checking and detected some network settings that triggered a network wide gridlock in the network simulation. We then corrected these coding problems and performed local and global parameter calibration on selected sites, and using our own O-D estimation tool, LPFE, to obtain the O-D trip tables for the 2-hour PM peak simulation. The simulation results were comparable to the ones obtained in a previous calibration study, but we were able to do this in a much shorter time with the help of our developed tools. Our case study also revealed some critical issues of micro simulation and areas for further improvement. These include improved lane utilization models, proper treatment of blocking traffic at intersections, and elimination of software conflicts between Paramics' O-D Estimator and the traffic signal control API.

This case study indicates that the developed calibration tools can indeed ease, streamline and speed up the calibration of micro simulation, particularly when the network concerned is large. It also reveals that the calibration of a micro simulation is a complex task that involves many engineering judgment and cannot be fully automated. In a micro simulation, every modeling detail matters and each must be treated properly to ensure a good simulation outcome.

It should be noted that while our calibration tools were specifically designed for Paramics, owing to its extensive usage in California, our calibration framework is applicable to any micro simulation model.

Further work of this research includes several improvements to the developed tools. For example, the conversion between LPFE and Paramics' networks at this stage still needs quite an amount of human intervention, and a more automated process is desirable. The signal control API, while the best in its class for Paramics, can benefit from expanding its control logic to include other types of controllers than the type-170. When working with signalized intersections, lane utilization and intersection blocking can create quite serious problems for calibration, and at present a user has to check intersection by intersection to spot problems and fix them, which is very time consuming. A tool to automate or at least streamline this calibration task is highly desirable. Besides these improvements to the developed tools, emerging ATMIS features should also be incorporated into the microscopic simulation packages and their calibration tools.

REFERENCES

Bacon, V., J.R. Windover, A.D. May 1995 Investigating Intelligent Transportation Systems Strategies On the Santa Monica Freeway Corridor, California PATH Research Report UCB-ITS-PRR-95-38.

Banks, J. H. 2004. Introduction to Transportation Engineering, Second Edition, McGraw-Hill.

Bell, M.G., C.M. Shield 1995. A log-linear model for path flow estimation. In Proceedings of the Fourth International Conference on the Application of Advanced Technologies in Transportation Engineering. Capri, U.K.

Bell, M.G., S. Grosso 1998. The path flow estimator as a network observer. Traffic Engineering and Control, No 39: p. 540-549.

Ben-Akiva, M. et al. 2002. Calibration and evaluation of MITSIMLab in Stockholm. 81st TRB Annual Meeting, CD-ROM.

Ben-Akiva, M. et al. 2004. Calibration of microscopic traffic simulation models with aggregate data. 83rd TRB Annual Meeting, CD-ROM.

Bridges, M. TRANSIMS-DOT. in 82nd Transportation Research Board Annual Meeting. 2003. Washington, D.C.

California Department of Transportation (CalTrans) 1993. Model 170 Traffic Signal Control Program, Version 3.0.

Cheu, R.L., X. Jin, K.C. Ng, D. Srinivasan, Calibration of FRESIM for Singapore expressway using genetic algorithm. ASCE Journal of Transportation Engineering, 1998. 124(6): p. 526-535.

Choe, T., A. Skabardonis, P. Varaiya 2002. Freeway performance measurement system (PeMS): an operational analysis tool. 81st TRB Annual Meeting CDROM.

Chu, L. et al 2004. A calibration procedure for microscopic traffic simulation. 83th TRB Annual Meeting CD-ROM, paper No. 04-4165.

Chu, L., Liu X., McNally, M. and Recker, W. (2005) Development of the Capability-Enhanced PARAMICS Simulation Environment, California PATH Research Report UCB-ITS-PRR-2005-12.

Dowling, R., J. Holland, and A. Huang 2002. Guidelines for applying traffic simulation modeling software. Dowling Associates Inc. Oakland, CA

Eagle Traffic Control Systems, SEPAC Actuated Signal Control Software, Rev. F.32.

Fresno COG 2006. TRANSPORTATION MODELS AND FORECASTS, http://www.fresnocog.org/document.php?pid=99, accessed December, 2006.

Gardes, Y., A. D. May. 1993. Simulation of IVHS on the Santa Monica freeway corridor using the integration model. Phase 2: Preliminary ATIS and ATMS experiments. Institute of Transportation Studies, UC Berkeley: Berkeley, CA.

Gardes, Y. et al. 2001. Freeway calibration and application of the PARAMICS model. 80th TRB Annual Meeting, CD-ROM.

Gardes, Y. and L. Bloomberg. 2003. Devore/I-15 Traffic Simulation Analysis Report-Level 1 Network Investigations. Institute of Transportation Studies, University of California, Berkeley.

GIS/Trans, Transportation model - GIS data integration, case study: Southern California Association of Governments (SCAG). 1996, Irvine Corporate Center: Irvine, CA.

Goldberg, D. E. 1989. Genetic Algorithms in Search Optimization and Machine learning. Addison-Wesley.

Gomes, G., Adolf D. May, Roberto Horowitz. A Microsimulation model of a congested freeway using VISSIM. in 83rd Transportation Research Board Annual Meeting. 2004. Washington, D.C.: Transportation Research Board.

Highway Agency, Design Manual for Roads and Bridges, Volume 12, Traffic Appriasal of Road Schemes, Section 2, Part I, Traffic Appraisal in Urban Areas. 1996, The Stationery Office: London.

Hoyer, R. M. Fellendorf. Parametrization of microscopic traffic flow models through image processing, 8th IFAC Symposium on Transport Proceedings, Chania, Crete, June 1997.

Intersection Development Corporation, 1994. "Multisonics Model 820A-VMS Vehicle Management System User Guide",

ISAC 1999. L. Experiences with PARAMICS at ISAC in winter 98-99.

Jeannotte, K. et al. 2004. Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis Tools. Report FHWA-HRT-04-038, Federal Highway Administration.

Kim, K. and L. R. Rilett 2004. A genetic algorithm based approach to traffic micro-simulation calibration using ITS data. 83rd TRB Annual Meeting, CD-ROM.

Lee, D.-H., Xu Yang, P. Chandrasekar. Parameter calibration for paramics using genetic algorithm. in 80th Transporation Research Board Annual Meeting CDROM. 2001. Washington D.C.: Transportation Research Board.

Lee, E.B., C.Kim. 2004. Validation of a microscopic traffic simulation study on urban freeway reconstruction project with high-traffic volume. in 84th Transportation Research Board Annual Meeting. 2004. Washington D.C.: Transportation Research Board.

Liu, H. X., L. Chu, W. Recker. 2001. PARAMICS API Development Document for Actuated Signal, Signal Coordination and Ramp Control. Working Paper UCI-ITS-TS-WP-01-03, Institute of Transportation Studies, University of California, Irvine.

Liu, H.X., L. Ding, J. X. Ban, A. Chen, P. Chootinan, 2006. A streamlined network calibration procedure for California SR41 corridor traffic simulation study. Proceedings of 85th TRB Annual Meeting, Paper No. 3050

Ma, J., H. M. Zhang and H. Dong 2006. Calibration of Departure Time and Route Choice Parameters in Micro Simulation with Macro Measurements and Genetic Algorithm. 85th TRB Annual Meeting CD-ROM, paper No. 2376.

Ma, T., B. Abdulhai 2002. Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters. Transportation Research Record 1806: 6-15.

McShane, W. R., Roess, R. P., and Prassas, E. S. 1998 "Traffic Engineering", Second Edition, Prentice Hall.

Quadstone Ltd. 2004a. Paramics V5.1 Estimator Reference Manual. Edinburgh, Scotland.

Quadstone Ltd. 2004b. Paramics V5.1-Modeller Reference Manual. Quadstone Ltd, Edinburgh, Scotland.

Quadstone Ltd. 2004c. Paramics V5.1-Modeller User Guide. Quadstone Ltd, Edinburgh, Scotland.

Sadegh, P. 1997. Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation. Automatica, 33:889–892.

Sadegh, P., J. C. Spall 1998. Optimal random perturbations for multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transcations on Automatic Control, 43(3):1480:1484.

Sheffi, Y., Urban transportation networks: equilibrium analysis with mathematical programming methods. 1985: Prentice-Hall.

Smith, M.C., Adel W. Sadek. Challenges in calibrating large-scale microscopic simulation model of diverse urban, suburban, and rural networks: practical guide. in 86th TRB Annual Meeting. 2006. Washington D.C.

Spall. J. C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transcations on Automatic Control, 37(3):332:341.

Spall, J. C. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Transcations on Aerospace and Electronic Systems, 34(3):817:823.

Transportation Research Board 2000. Highway Capacity Manual, Washington D.C.

Zhang, M.H., Y. Nie, W. Shen. 2005. Development of A Path Flow Estimator for Inferring Steady-State and Time-Dependent Origin-Destination Trip Matrices, Institute of Transportation Studies, University of California at Davis, CA.

Zhang, H. Michael, J. Ma. 2006. DEVELOPING CALIBRATION TOOLS FOR MICROSCOPIC TRAFFIC SIMULATION PART I: OVERVIEW OF CALIBRATION METHODS AND GUIDELINES ON PROJECT SCOPING AND DATA COLLECTION. University of California at Davis, PATH Research Report.

Zhang, H.M., J. Ma, H. Dong. 2006. DEVELOPING CALIBRATION TOOLS FOR MICROSCOPIC TRAFFIC SIMULATION, PART II: CALIBRATION FRAMEWORK, AND CALIBRATION OF LOCAL/GLOBAL DRIVING BEHAVIOR AND D/R CHOICE MODEL PARAMETERS. University of California at Davis, PATH Research Report.