OPTIMIZATION-BASED ASSISTED CALIBRATION OF TRAFFIC SIMULATION MODELS

David K. Hale[1], Constantinos Antoniou[2], Mark Brackstone[3], Dimitra Michalaka[4], Ana T. Moreno[5], and Kavita Parikh[6]

[1]University of Florida
PO Box 116585
Gainesville, FL 32611-6585
david@ce.ufl.edu

[2]National Technical University of Athens
School of Rural and Surveying Engineering
Laboratory of Transportation Engineering
antoniou@central.ntua.gr

[3]IOMI Consulting
Southampton
Hants, UK, SO40 7FF
mark.brackstone@iomi.eu

[4]The Citadel
171 Moultrie Street
Charleston, SC 29409
Dimitra.Michalaka@citadel.edu

[5]Universitat Politecnica de Valencia
Camino de Vera, S/N, Edificio 4ª
46071 Valencia, Spain
anmoch@cam.upv.es

[6]RS&H, Inc.
10748 Deerwood Park Blvd South
Jacksonville, FL 32256
Kavita.parikh@rsandh.com

[1] Present Address: Leidos, Inc., 11251 Roger Bacon Drive, Reston, VA, 20190, david.k.hale@leidos.com

Abstract. *Usage of traffic simulation has increased significantly in recent decades; and this high-fidelity modelling, along with moving vehicle animation, has allowed important transportation decisions to be made with better confidence. During this time, traffic engineers have typically been encouraged to embrace the process of calibration, in which steps are taken to reconcile simulated and field-observed traffic performance. According to international surveys, top experts, and conventional wisdom, existing (non-automated) methods of calibration have been difficult and/or inadequate. There has been a significant amount of research on techniques to improve calibration, but many of these projects and papers have not provided the level of flexibility and practicality typically required by real-world engineers. With this in mind, a patent-pending (US 61/859,819) architecture for software-assisted calibration was developed to maximize practicality, flexibility, and ease-of-use. This architecture is called SASCO (i.e. Sensitivity Analysis, Self-Calibration, and Optimization). The original optimization method within SASCO was based on "directed brute force" (DBF) searching; performing exhaustive evaluation of alternatives in a discrete, user-defined search space. Simultaneous Perturbation Stochastic Approximation (SPSA) has also gained favor as an efficient method for optimizing computationally expensive, "black-box" traffic simulations, and was also implemented within SASCO. This paper assesses the qualities of DBF and SPSA, so they can be applied in the right situations. Case study calibrations from synthetic and real-world networks reveal that the two optimization methods have different advantages, and in some cases should be applied in tandem. SPSA was found to be the fastest method, which is important when calibrating numerous inputs, but DBF was more reliable. Additionally DBF was better than SPSA for sensitivity analysis, and for calibrating complex inputs. Regardless of which optimization method is selected, the SASCO architecture appears to offer a new and practice-ready level of calibration efficiency.*

Keywords: Microscopic Simulation, SPSA, Assisted Calibration, Calibration, Simulation-Based Optimization, Traffic Simulation.

1 INTRODUCTION

Computer programs for traffic simulation have become more advanced in recent decades. Use of traffic simulation has increased significantly; and this high-fidelity modeling, along with moving vehicle animation, has allowed important transportation decisions to be made with better confidence. During this time, traffic engineers have typically been encouraged to embrace the process of calibration, in which steps are taken to reconcile simulated and field-observed traffic performance. One example of such encouragement can be found in the U.S. Federal Highway Administration (FHWA) guidelines for applying micro-simulation modeling software [1]. These guidelines state "the importance of calibration cannot be overemphasized"; and refer to a study [2] by Bloomberg et al., which makes the following statement: "Recent tests of six different software programs found that calibration differences of 13 percent in the predicted freeway speeds for existing conditions increased to differences of 69 percent in the forecasted freeway speeds for future conditions." Therefore, these studies and guidelines demonstrate some of the dangers of neglecting calibration.

Despite the importance of calibration, practical application has been difficult. According to international surveys, top experts, and conventional wisdom, existing (non-automated) methods of calibration have been difficult and/or inadequate. Consulting engineers and DOT personnel have expressed strong interest in making calibration faster, cheaper, easier, and requiring less expertise. Comprehensive surveys [3] revealed that 19% of simulation users do not perform any amount of calibration, and that only 55% of calibration efforts are based on guidelines that exist in the literature [1, 4, 5]. Finally, some simulation users believe that they have somewhat mastered the process of calibration; but that many years of experience are required to master this process, or that high-quality calibration is overly time-consuming.

There has been significant research to improve calibration for traffic simulation. Some of this research focuses on traffic assignment and origin-destination flows. A second area of research focuses on pattern matching of simulated versus field-measured vehicle trajectories and/or speed-flow relationships. A third area of research focuses on simulation-based optimization, in which the numeric discrepancy between simulated and field-measured results becomes an objective function to be minimized. Many of these research projects and papers have not provided the level of flexibility and practicality that are typically required by real-world engineers. In the papers by Lee and Ozbay [6], Lee et al. [7], and Menneni et al. [8], the authors present substantial literature reviews for both manual and automated calibration techniques. The authors then emphasize that, despite the extensive efforts, existing calibration procedures continue to require excessive time and expertise.

With this in mind, a patent-pending (US 61/859,819) architecture for software-assisted calibration was developed, within the simulation-based optimization (SO) family of methods. The new architecture uses a database of input parameters, to pre-define a narrow set of trial values to be used during optimization. The architecture also allows engineers to prioritize input and output parameters, and specify a tolerable computer run time, prior to initiating the SO-based calibration process. The "directed brute force" (DBF) search process is believed to be a key element in making the architecture flexible and practical, for real-world use. These same features were later extended to provide an enhanced platform for sensitivity analysis, and optimization. The acronym term "SASCO" (Sensitivity Analysis, Self-Calibration, and Optimization) was adopted. Thus, SASCO provides a database-centric framework to support any one of these three analysis types.

In recent years, Simultaneous Perturbation Stochastic Approximation (SPSA) has gained favor as an efficient method for optimizing computationally expensive, "black-box" traffic simulations. For example in 2007, Balakrishna et al. [9] selected SPSA "for its proven performance and computational properties in large-scale problems". In 2007-2008, Ma et al. [10] and Lee's dissertation [11] both demonstrated the effectiveness of calibrating PARAMICS traffic simulations with SPSA. Also in 2008, the Transportation Research Board posted a research needs statement [12] for calibration of simulation models, saying "recent research indicates that the SPSA algorithm can solve very large noisy problems in a computationally attractive fashion." More recently in 2013, Paz et al. [13] demonstrated the effectiveness of calibrating CORSIM simulations with SPSA. Given this track record, SPSA appears to be an attractive option for assisted calibration of computationally expensive simulations, and a possible alternative to the DBF optimization originally implemented within SASCO.

As such, this paper will assess the qualities of DBF and SPSA, so they can be applied in the right situations. The scope is limited to calibration (without validation) of average output values (not distributions); with data collection from one real-world corridor, using one underlying simulator (FRESIM, which is part of CORSIM). Follow-up studies will hopefully perform similar assessments with more optimization methods, validation following calibration, calibration of output distributions, on more real-world corridors, using more simulators.

2 SASCO ARCHITECTURE

As stated earlier, SASCO can be classified as belonging to the simulation-based optimization (SO) [14] family of methods. Under SO, numerous input combinations are simulated for the purpose of minimizing or maximizing an objective function. When the goal of SO is calibrating a simulation model, minimizing the difference between simulated and field-measured outputs is sometimes the objective function. Another important aspect of SO is the searching method. Intelligent searching methods are designed to obtain the best possible solutions in the shortest amount of time, or using the smallest number of trials. Intelligent searching methods would not be needed if computers were fast enough to perform "brute force" searching, to simulate every combination of inputs. But given the speed of modern computers, brute force optimization is not practical for computationally expensive traffic simulations.

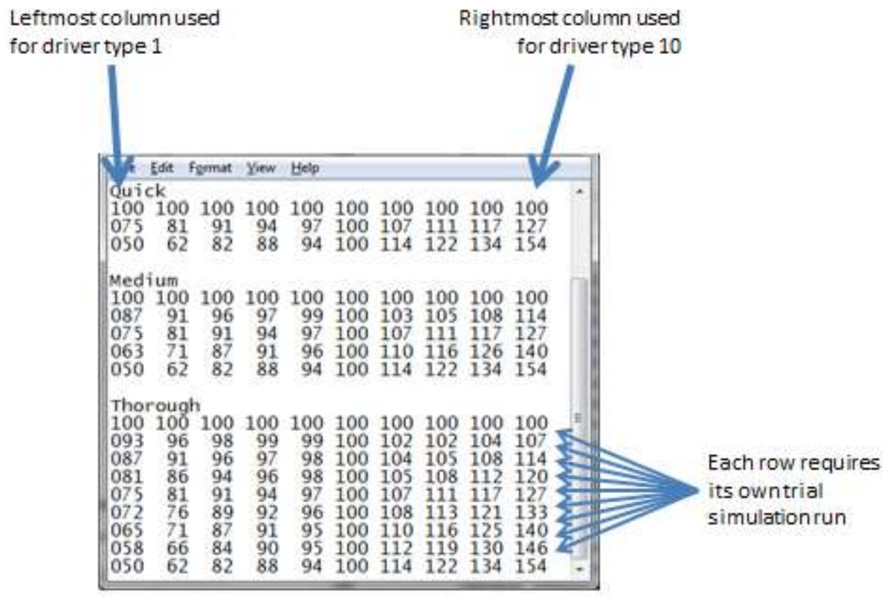
Several of the papers cited by [6, 7, 8] investigated SO-based calibration via genetic algorithms (GA) [15]. GA is a well-known, intelligent heuristic searching method. The heuristic methods are able to continuously adapt their search in response to intermediate trial results. Although GA appears to have gained commercial success in multiple industries, it has not gained commercial popularity for calibration of traffic simulations. This is likely due to the fact that GA frequently requires thousands of trials to locate an acceptable solution, and the traffic simulations cannot process thousands of trial runs in a reasonable time frame. Other heuristic methods requiring a relatively large [16, 17] number of trials, such as downhill simplex or simulated annealing, also do not seem suitable for computationally expensive simulation models. On the other hand, faster heuristics such as hill-climbing and the greedy algorithm are known to produce unsatisfactory [18, 19] solutions.

When developing a new methodology for SO-based calibration, it was believed the end-user needed complete control over the run time, number of trials, and the trial values. Given the wide variety of computer run times for various simulators and traffic networks, a "one size fits all" searching method seemed impractical. Moreover, different jurisdictions have different standards and tolerances for calibration. Finally, different analysis types (small corridor, large network, academic research) require much different amounts of calibration. Although the run time for powerful heuristics can be arbitrarily controlled, for example with small annealing schedules or numbers of generations, this would likely produce unacceptable calibration results. Giving the end-user total control over the run time, number of trials, and the trial values themselves seemed the best solution. This led to development of a database-centric architecture; designed for directed brute force optimization, and described in this section.

2.1 Input Parameter Database

The concept of fast food ordering was considered as an easy method to specify preferences. In fast food restaurants it is common to make entrees, side dishes, and beverages available in sizes small, medium, and large. For example a customer may order a large sandwich, small fries, and a medium soda. They know in advance the price they will pay, the exact meal portions, and the order can be made quickly. Similarly it is possible to define the amount of calibration for each input parameter. For example, suppose someone wished to calibrate the percentage of cooperative drivers. Impacts of queue spillback are sometimes more severe without driver cooperation. Although different products would handle cooperation in different ways, by simulating three candidate values (0%, 50%, 100%) it would be possible to determine which one produces the best match between simulated and field-measured outputs. Other end-users might prefer to search at 25% increments (0%, 25%, 50%, 75%, 100%), or at 10% increments (0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%).

For many inputs it wouldn't be effective to specify increment percentages. Regarding queue discharge headways, the default value (2.0 seconds) produces a saturation flow of 1800 vehicles, but values between 1.4 and 2.6 are needed for various conditions. If the product supports values between 1.4 and 9.9, using increment percentages would be counterproductive; because values above 2.6 are almost never relevant, and are only needed in extreme circumstances. In this case the input parameter database could define Quick (1.8, 2.0, 2.2), Medium (1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3), and Thorough (1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6) levels of searching. Another benefit of the database concept is the ability to calibrate inter-dependent input parameters. For example, one product allows calibration of free-flow speed multipliers for 10 driver "aggressiveness" levels. Instead of calibrating only one value, all ten values must be calibrated simultaneously (summation of values must be 1000%). Therefore, the database can define trial values as shown in Figure 1. The database can also be made to support localized calibration of specific links and nodes. Another benefit of the database concept is the ability to supply default databases, but then allow end-users to customize that database when needed. Thus the database could define appropriate trial values for every input parameter relevant to the calibration effort.


Leftmost column used for driver type 1
Rightmost column used for driver type 10
Edit
Format
View
Help
Each row requires its own trial simulation run

```
Quick
100 100 100 100 100 100 100 100 100 100
075  81  91  94  97 100 107 111 117 127
050  62  82  88  94 100 114 122 134 154

Medium
100 100 100 100 100 100 100 100 100 100
087  91  96  97  99 100 103 105 108 114
075  81  91  94  97 100 107 111 117 127
063  71  87  91  96 100 110 116 126 140
050  62  82  88  94 100 114 122 134 154

Thorough
100 100 100 100 100 100 100 100 100 100
093  96  98  99  99 100 102 102 104 107
087  91  96  97  98 100 104 105 108 114
081  86  94  96  98 100 105 108 112 120
075  81  91  94  97 100 107 111 117 127
072  76  89  92  96 100 108 113 121 133
065  71  87  91  95 100 110 116 125 140
058  66  84  90  95 100 112 119 130 146
050  62  82  88  94 100 114 122 134 154
```

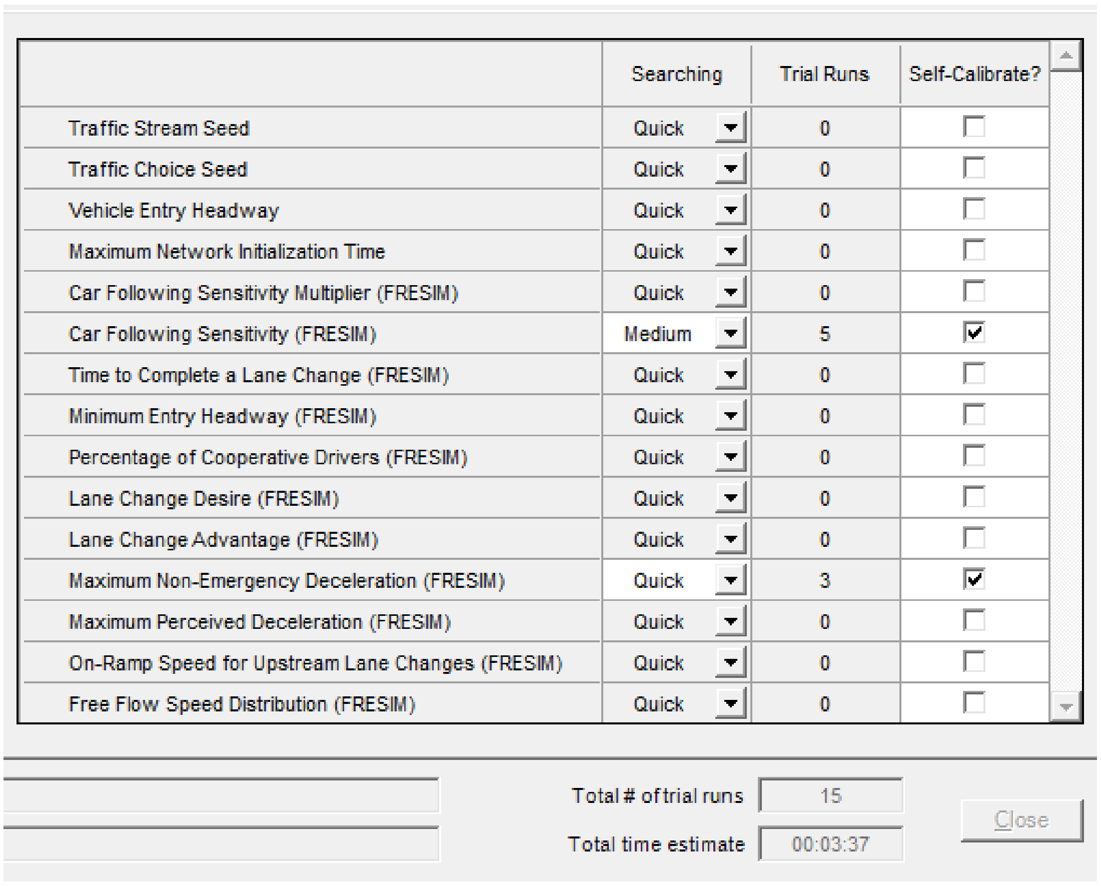
Figure 1: An example of "free-flow speed multiplier" trial values, within the input parameter database.

2.2 Input Data User-Interface

Figure 2 illustrates an example user-interface (UI) software screen design, for selection of inputs to be calibrated. This UI screen loads the input parameter database at runtime; thus databases can be customized or updated at any time, without needing new software. The end-user need only put a check mark next to each input parameter they'd like to calibrate, and then select a calibration thoroughness level (Quick, Medium, or Thorough) for each input selected.

The input data UI can provide valuable run time estimates, at the bottom of the screen. Advance knowledge of computer run times is expected to be a welcome feature for usability and practicality. In the concept of fast food menus discussed earlier, if a customer decided the total price was uncomfortably high, they could still change their order prior to purchase. In the case of directed brute force optimization, if an end-user chose Quick searching (3 trials) on one input parameter, Medium searching (5 trials) on a second, and Thorough searching (10 trials) on a third, run time from an initial simulation could be multiplied by 150 (i.e., 3*5*10), to produce a reasonable estimate. If the end-user could afford a longer run time, they might choose more thorough searching on some parameters, or add to the list of parameters to be calibrated. If the estimated run time were uncomfortably high, they might reduce the amount of searching on some parameters, or perhaps omit certain parameters from the optimization.

In the case of Simultaneous Perturbation Stochastic Approximation (SPSA), run time estimates could reflect the max number of iterations, and the number of simulations per iteration. Moreover, by allowing the end-user to adjust range limits (continuous SPSA) or trial values (discrete SPSA) for any input, SASCO's input data UI could augment the efficiency of SPSA.



	Searching	Trial Runs	Self-Calibrate?
Traffic Stream Seed	Quick	0	☐
Traffic Choice Seed	Quick	0	☐
Vehicle Entry Headway	Quick	0	☐
Maximum Network Initialization Time	Quick	0	☐
Car Following Sensitivity Multiplier (FRESIM)	Quick	0	☐
Car Following Sensitivity (FRESIM)	Medium	5	✓
Time to Complete a Lane Change (FRESIM)	Quick	0	☐
Minimum Entry Headway (FRESIM)	Quick	0	☐
Percentage of Cooperative Drivers (FRESIM)	Quick	0	☐
Lane Change Desire (FRESIM)	Quick	0	☐
Lane Change Advantage (FRESIM)	Quick	0	☐
Maximum Non-Emergency Deceleration (FRESIM)	Quick	3	✓
Maximum Perceived Deceleration (FRESIM)	Quick	0	☐
On-Ramp Speed for Upstream Lane Changes (FRESIM)	Quick	0	☐
Free Flow Speed Distribution (FRESIM)	Quick	0	☐

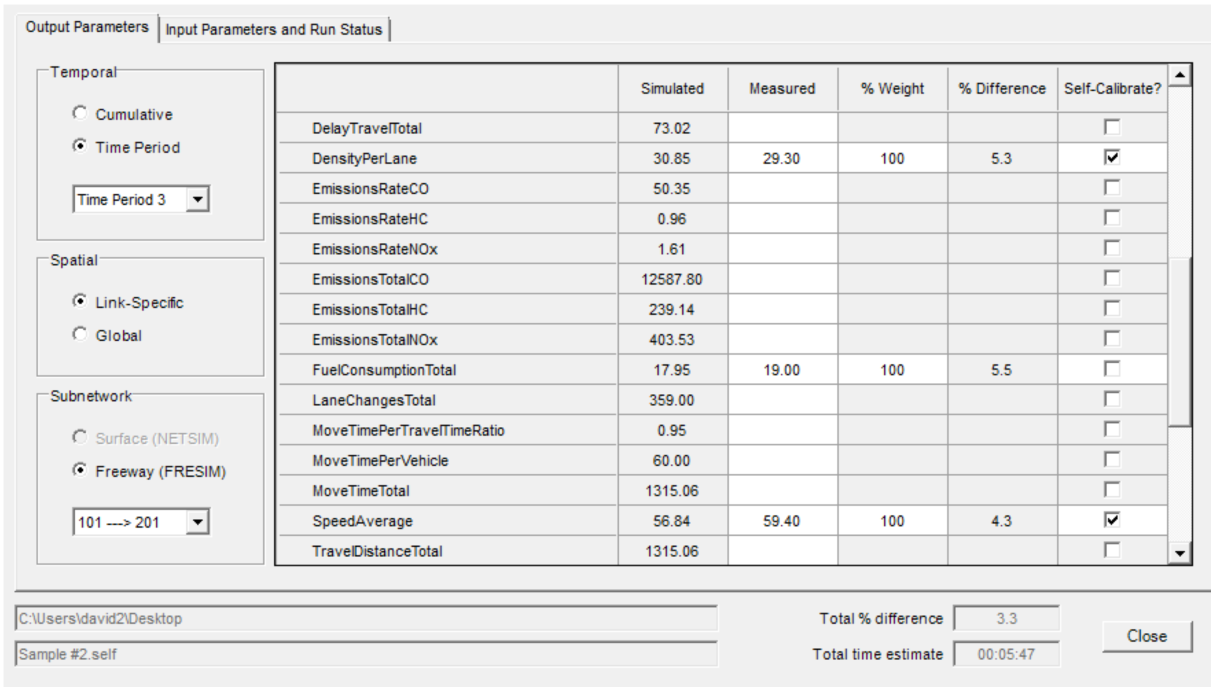
Total # of trial runs: 15
Total time estimate: 00:03:37

Figure 2: An example of user-interface screen design, for selecting calibration input parameters.

2.3 Output Data User-Interface

According to comprehensive literature reviews [6, 7, 8], prior methods offer a limited set of output parameters for calibration. By contrast, SASCO architectures allow any simulation output value, or distribution [20] of values, to be used for calibration. The software must also accept data entry for the ground truth output values. These ground truth values might be measured in the field, or estimated from the office, or obtained from a separate traffic analysis tool. Figure 3 illustrates an example screen design for selection of output parameters. Similar to the input data UI, the end-user can simply put a check mark next to each output they'd like calibration to be based on. Priority weighting defaults to 100 for each output, and can be left alone if all outputs are deemed equally important. The total percent difference (between simulated and field-measured outputs) appears at the bottom. This total percent difference responds to data entry in real-time, and is sensitive to the priority weightings.

Because the full set of outputs is not simultaneously viewable on screen, software controls (e.g., scrollbars, combo boxes, and radio buttons) must allow browsing amongst all outputs. The database-centric, fast-food ordering concept allows calibration to be based on cumulative outputs, time period-specific outputs, global outputs, link-specific outputs, surveillance detector outputs, surface street outputs, freeway outputs, or any combination of the above outputs. The end-user can provide ground truth values for one output parameter, or ten, or one hundred, or for thousands of outputs, depending on their needs. Input and output calibration settings entered by the user are continuously saved into a separate file (e.g., "Filename.self"); for future reference, and to prevent any need for re-typing. The need for a transparent approach to calibration has been cited by top experts, and the "self" file (or similar) provides this transparency. The UI design shows it is not necessary to load all outputs simultaneously. Although there may be millions of values, it is only necessary to load those outputs consistent with the active choices on screen. Most traffic analysis tools support map-based data entry, via right-clicking on links and nodes; so for large networks with many links, the software should be capable of jumping to the UI in Figure 3, and automatically switching to the chosen link.



Temporal: Cumulative [✓ Time Period] — Time Period 3

Spatial: [✓ Link-Specific] Global

Subnetwork: Surface (NETSIM) [✓ Freeway (FRESIM)] — 101 ---> 201

	Simulated	Measured	% Weight	% Difference	Self-Calibrate?
DelayTravelTotal	73.02				☐
DensityPerLane	30.85	29.30	100	5.3	☑
EmissionsRateCO	50.35				☐
EmissionsRateHC	0.96				☐
EmissionsRateNOx	1.61				☐
EmissionsTotalCO	12587.80				☐
EmissionsTotalHC	239.14				☐
EmissionsTotalNOx	403.53				☐
FuelConsumptionTotal	17.95	19.00	100	5.5	☐
LaneChangesTotal	359.00				☐
MoveTimePerTravelTimeRatio	0.95				☐
MoveTimePerVehicle	60.00				☐
MoveTimeTotal	1315.06				☐
SpeedAverage	56.84	59.40	100	4.3	☑
TravelDistanceTotal	1315.06				☐

C:\Users\david2\Desktop

Sample #2.self

Total % difference: 3.3

Total time estimate: 00:05:47

Figure 3: An example of user-interface software screen design, for selecting calibration output parameters.

2.4 Optimization Algorithm

After the end-user chooses inputs and outputs, any optimization method could be applied. Powerful heuristics (e.g. GA, simulated annealing) require an excessive and unpredictable number of trials, and are not suited to computationally expensive simulations. Simpler heuristics (e.g. hill-climbing, greedy algorithms) tend to produce poor [18, 19] solutions. By contrast, SASCO was specifically designed to calibrate expensive simulations. End-users would pro-actively customize trial values and run times for directed brute force (DBF) searching.

The only difference between "directed" and "regular" brute force is that DBF is restricted to fewer trial values. In the earlier example, Quick searching (3 trials) on one input parameter, Medium (5 trials) on a second, and Thorough (10 trials) on a third, would lead to 3*5*10=150 possible solutions. The number of trials for Quick, Medium, and Thorough would be flexible. The database would offer intelligent defaults, but also allow customization. Although SASCO was developed with DBF in mind, Simultaneous Perturbation Stochastic Approximation (SPSA) has gained favor for efficient optimization of complex simulations, as discussed in Section 1. For the discrete and continuous forms of SPSA, the database can specify trial values and range limits for each input, respectively. Given that no optimization method outperforms all others under all conditions [21], it is important for SASCO to support multiple methods. Rather than trying to determine the "best" method for all conditions, this paper will assess the strengths and weaknesses of DBF and SPSA.

Figure 4 illustrates the overall SASCO architecture. When the run is launched, the optimization algorithm (e.g., DBF or SPSA) proceeds to minimize the objective function, by testing acceptable input values from the input data UI. Throughout the run objective function values become lower and lower, thus indicating a better-calibrated model.

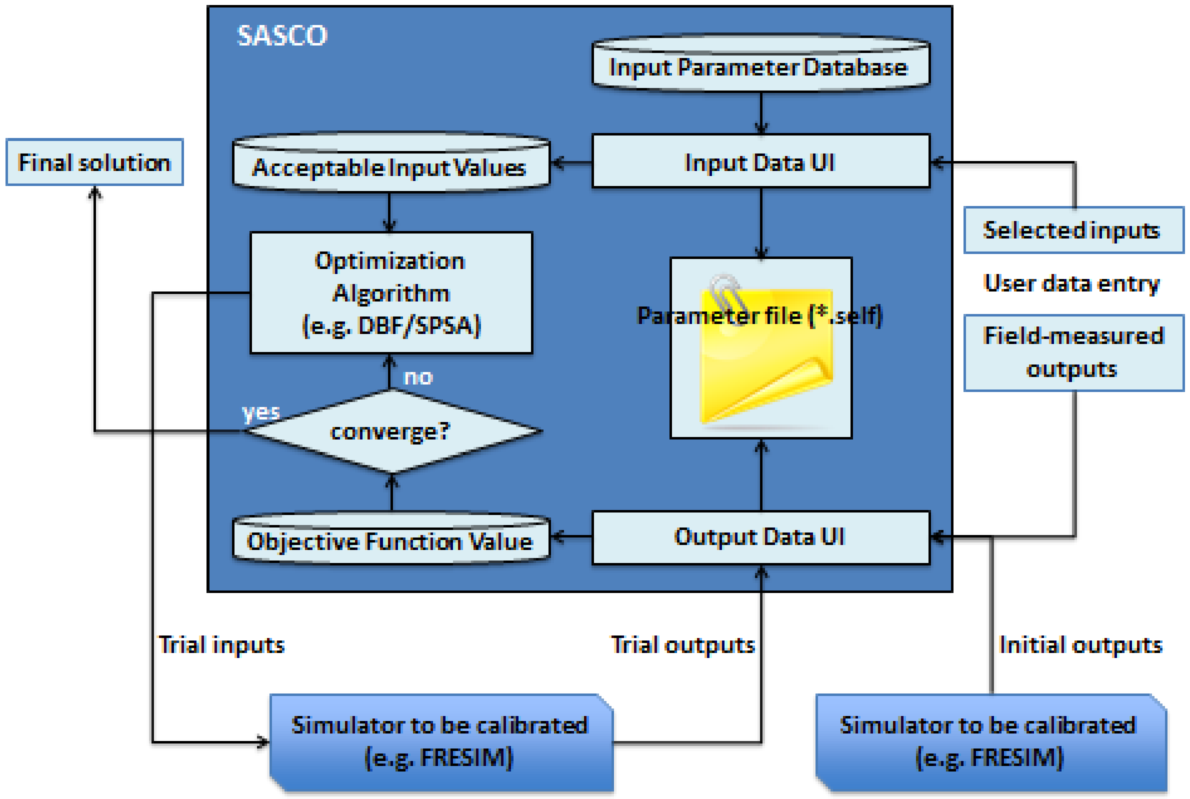


Figure 4: Overall SASCO architecture.

2.5 Practical Considerations

Efficiency: Origin-destination (O-D) calibration requires optimizing a significant number of unique values. When O-D flows don't need to be optimized, calibration efforts sometimes focus on a much smaller number of high-priority inputs. This is because low-priority inputs tend to 1) have little impact on results, and/or 2) change outputs inappropriately. Indeed, calibration of start-up lost times is often discouraged, because clients are confident in the default value. SASCO was originally developed with DBF optimization in mind, and originally intended to address calibrations focusing on a relatively small number of inputs. However SASCO's ability to import calibrated values enhances DBF's ability to optimize more inputs more quickly. For example, suppose the end-user wished to calibrate four inputs, each having five trial values. Simultaneous optimization would require 5*5*5*5=625 simulations. If desired the end-user could save time by optimizing the two most important inputs, importing those calibrated values, and then optimizing the least important inputs. This "sequential" optimization would require only (5*5)+(5*5)=50 simulations. Another option would be "iterative sequential" where an additional 50 simulations are performed, using optimized values from the first 50 simulations as a starting point. Sequential optimization requires fewer trial simulations than simultaneous optimization, but does not evaluate as many candidate solutions. The important point is that SASCO allows the flexibility of choosing between simultaneous or sequential optimization, and allows the flexibility of choosing between algorithms such as DBF or SPSA.

Stochasticity: Many products contain random number seed (RNS) data entry, to analyze stochastic effects. Changing the RNS can influence driver "aggressiveness", driver decisions, headways between vehicles, etc. When inexperienced engineers perform only one simulation, they might misinterpret those outputs as being typical, average results. For unstable traffic conditions, numerous simulations (with different RNS) are needed for experienced engineers to be confident in their results. This presents a dilemma for automated, SO-based calibration. If 10 simulations (with different RNS) were needed for each combination of calibration inputs, this would drastically inflate computer run times. The SASCO architecture provides some assistance in addressing uncertainty and randomness. At the top of Figure 2 shown earlier, RNS inputs (Traffic Stream Seed, Traffic Choice Seed, etc.) are available for selection, similar to the calibration inputs. These RNS inputs sit atop the list to encourage end-users to analyze randomness. The architecture allows simultaneous optimization of RNS and calibration inputs; but run times would be high, and results difficult to interpret. To manage randomness when run times are high, "pre- and post-" analysis is one option. Prior to the "standard" calibration run, a pre-calibration run could be performed to "calibrate" and import RNS producing the most "average" results. These RNS could then remain in effect during the standard calibration run. After importing the optimized inputs, a post-calibration run could be performed, to determine whether the solution was stable for different RNS. Pre-calibration stochastic analysis would increase odds of a stable final solution; but if unstable, the "average" RNS could be imported at this time. The sequential technique of "pre- and post-" stochastic analysis is not a perfect strategy because randomness is ignored during the standard run. Another option would be performing multiple optimizations but manually changing RNS before each. For example, after five optimizations one could assess variance of the five final objective function values, and then import calibrated settings that produced the median final objective function value. The important point is that SASCO allows end-users to control how much stochastic analysis is performed; whether that involves sequential optimizations, or whether that involves reduced levels (Quick/Medium/Thorough) of RNS replication.

3 SYNTHETIC NETWORK CASE STUDY

Test calibrations were performed to assess benefits of DBF and SPSA, when applied from within SASCO, with FRESIM as the simulator. The terms "calibration" and "optimization" will be used interchangeably because every optimization achieves calibration. The first case study involved a synthetic network. Figure 5 illustrates a simple freeway facility with three through lanes on the mainline, plus three pairs of on-ramps and off-ramps. Locations "A" and "B" exhibit the largest discrepancy in results, as discussed later. Both downstream weaving sections are approximately 0.8 kilometers wide, whereas all other ramp spacings are approximately 1.6 km. Mainline free-flow speeds are set to 97 km/h, and ramp free-flow speeds are set to 64 km/h. The model contains five 15-minute time periods, with 5% trucks throughout the network. Table 1 illustrates the mainline and ramp flow rates during each time period.

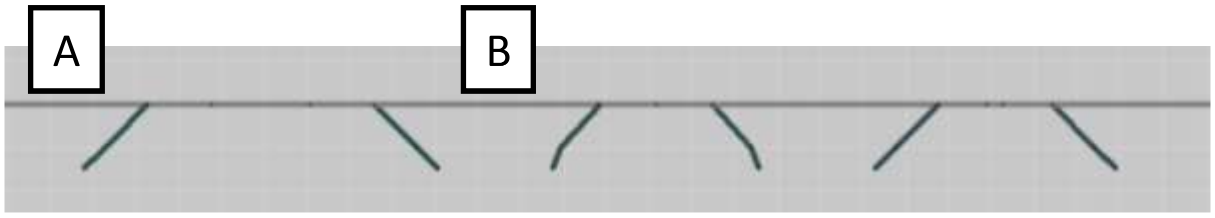


Figure 5: Synthetic network roadway geometry.

	TP1	TP2	TP3	TP4	TP5
upstream mainline	4505	4955	5225	4685	3785
upstream on-ramp	450	540	630	360	180
middle on-ramp	540	720	810	360	270
downstream on-ramp	450	540	630	450	270
upstream off-ramp	270	360	270	270	270
middle off-ramp	360	360	360	360	180
downstream off-ramp	270	270	450	270	180

Table 1: Synthetic network flow rates (in vehicles per hour) across five time periods.

Field-measured speed and density data are available on the mainline, in time period #3. Densities are between 29.3 vehicles per lane per hour on the upstream end, 31.9 vplph in the middle segments, and 37.7 vplph on the downstream end. Speeds are between 96 km/h upstream, 94 km/h in the middle, and 89 km/h downstream. Overall, simulated speeds and densities are 3.4% worse (i.e. lower speeds, higher densities) than their field-measured counterparts. Preliminary stochastic analysis indicates that, although network-wide percent differences fluctuate between 2.7% and 3.8% for individual simulations, the average percent difference over 25 runs (different random number seeds) remains 3.4%.

The next step is to select from an available list of 20 freeway calibration parameters on the default list (this number of parameters would be different for each simulation product). To choose parameters wisely, it helps to take a closer look at the segment-specific results; which in this case reveal a better match in the more-congested downstream segments (between 0 and 3 percent), than in locations "A" and "B" (between 5 and 6 percent). Ideally, detailed field observations would explain the reason(s) for these discrepancies. One possible reason for these results is that, in the real world, drivers are actually driving far above the posted speed limit in lengthy sections A and B, indicating that free-flow speeds have not been entered properly within the model. Alternatively, car-following and lane-changing issues could be the problem. Without proper field observations it might be necessary to provide multiple calibration options to the client, and inform them that insufficient information exists to know which option is best. If localized calibration were performed on segments A and B, 15 global parameters could be eliminated from consideration. Given that these long segments are not likely experiencing significant influence from downstream ramps, three ramp-specific input parameters could be eliminated from consideration. The two remaining segment-specific parameters are car-following multipliers and desired free-flow speeds.

At the Thorough level, simultaneous DBF calibration of these two parameters would require 77 trial simulation runs (18 minutes). Although 18 minutes would be acceptable in many cases, the same parameters could be calibrated with only 35 runs (8 minutes) at the Medium level. When the calibration is performed in this manner, percent difference drops from 3.4% to 1.8%. On segments A and B, car-following multipliers were optimized to 120%, whereas desired free-flow speeds were optimized to 104 km/h. After importing the optimized input values, a subsequent stochastic analysis (25 trial runs with different random number seeds) reveals an average percent difference of 1.9%, implying that the 1.8% result is fairly stable. Moreover segments A and B, whose discrepancies between simulated and field-measured results had been in the range of 5-6%, are now showing discrepancies of 0-2%. Percent difference results for alternative calibration strategies are summarized below:

- Original settings (3.4% overall, 5.5% on critical segments A and B)
- Car-following calibration on segments A and B (3.0% overall, 4.0% on A and B)
- Car-following calibration on all segments (2.6% overall, 4.3% on A and B)
- Multivariate calibration on all segments (2.5% overall, 4.0% on A and B)
- Multivariate calibration on segments A and B (1.8% overall, 1.0% on A and B)

Multivariate calibration on segments A and B improved the objective function from 3.4% to 1.8%. This calibration was performed using directed brute force (DBF) searching; in which car-following multipliers (five trial values) and desired free-flow speeds (seven trial values) were optimized simultaneously. Table 2 illustrates all 35 parameter value combinations, and shows the optimum solution (1.8%) was achieved at 104 km/h and 120%.

		Car-Following Multiplier (%)				
		60	80	100	120	140
Desired FFS (km/h)	64	19.9	20.1	20.1	20.6	22.2
	72	14.6	15.2	14.9	15.5	16.5
	80	10.7	10.4	10.6	11.3	12.5
	88	6.3	7.2	6.6	7.7	7.8
	96	2.5	2.9	3.4	3.9	4.6
	104	3.3	2.3	2.1	1.8	2.2
	112	5.3	5.2	4.9	3.9	2.6

Table 2: Discrete trial values used during synthetic network case study.

The DBF search method tends to benefit from a highly reduced search space, which the end-user can control through SASCO. One disadvantage of reducing the search space too much is that numerous trial values are skipped over. Table 2 implies the optimum range of desired FFS is in the range of 96 to 112 km/h, and that car-following sensitivity multiplier should fall between 100 and 140 percent. Table 3 illustrates a portion of these 451 parameter value combinations, and shows the optimum solution (1.2%) was achieved at 103 km/h and 105% (or 108%). Objective function values such as 1.2% were based on a single simulation, so the discontinuity of tabular results reflects and includes some amount of stochastic noise.

	Car-Following Sensitivity Multiplier (%)													
Desired Free-Flow Speed (km/h)	100	101	102	103	104	105	106	107	108	109	110	111	112	113
99	3.4	3.4	3.4	3.5	3.5	3.6	3.9	3.4	3.8	4.0	3.8	3.8	3.7	4.0
100	3.3	2.6	2.9	3.2	2.6	2.8	2.9	3.3	3.5	2.9	2.9	2.9	3.1	3.0
101	2.5	1.8	2.4	2.3	2.5	2.6	2.1	2.4	2.3	2.6	2.2	2.2	2.0	2.4
102	1.7	1.8	1.6	1.7	1.4	1.5	1.5	1.6	1.8	1.7	1.9	1.9	2.1	1.8
103	2.0	2.1	1.7	1.7	1.4	1.2	1.5	1.5	1.2	2.1	1.5	1.5	1.4	1.7
104	2.1	2.4	2.2	2.1	2.1	2.1	2.1	2.5	1.9	1.8	2.4	2.4	1.8	2.2
105	2.9	2.6	3.0	2.8	2.8	2.9	2.6	3.2	2.9	2.7	2.5	2.5	2.5	2.6
106	3.7	2.9	3.4	3.6	3.4	3.2	3.0	3.5	3.2	3.2	3.0	3.0	3.2	3.1
107	3.9	4.0	3.8	3.6	4.0	3.6	3.7	3.6	3.8	3.4	3.5	3.5	3.9	3.6
108	4.8	4.3	4.5	4.3	4.5	4.2	4.2	4.2	4.3	4.3	4.5	4.5	4.1	4.3
109	4.9	4.8	4.7	4.8	4.7	4.4	4.5	4.5	4.9	4.6	4.8	4.8	4.6	4.5

Table 3: Continuous trial values used during synthetic network case study.

The DBF and Simultaneous Perturbation Stochastic Approximation (SPSA) search methods can be applied to either the narrow ("discrete") search space having 35 total options, or to the broader and more continuous search space having 451 total options. In this case the discrete DBF optimization would require 8 minutes on a typical computer, whereas the continuous DBF optimization would require approximately 103 minutes. Unfortunately the end-user would not know in advance whether 95 additional minutes of optimization would yield significant additional benefit. As discussed earlier, SPSA has a track record of efficiently optimizing complex problems, requiring fewer trials than other methods. By implementing SPSA within the framework of SASCO, it was possible to compare of DBF and SPSA efficiency.

Table 4 contains SPSA calculations for desired free-flow speed. Similar calculations exist for car-following sensitivity, but for brevity are not shown here. The underlying simulator (FRESIM) assumes English units (intermediate Table 4 values could not be converted to SI units without corrupting algorithm calculations). For the discrete case, "xplus" and "xminus" trial values were rounded to the nearest 5 mi/h (8 km/h) prior to simulation. For the continuous case trial values were rounded to the nearest 1 mi/h (1.6 km/h), because FRESIM required integer data entry for this input parameter. Alpha and gamma are values recommended [22] by SPSA experts. Parameters "a" and "c" are user input values, which affect optimization efficiency. Intermediate values "ak" and "ck" are based on "a" and "c", but grow smaller with each iteration. Delta is a Bernoulli +1 or -1 distribution, with a 50/50 probability for each outcome. The objective of SPSA is to optimize theta. Xplus and xminus are trial values on either side of theta. Yplus and yminus are objective function values generated by the simulator. Grade is the gradient generated by yplus and yminus, but also affected by "ck".

SPSA typically performs two simulations per iteration, such that Table 4 illustrates a similar number of simulations (i.e., 34) as were performed during the discrete DBF optimization (Table 2). With a 58 mi/h (93 km/h) starting point, SPSA converged on 67 mi/h (108 km/h) after only 14 simulation runs, which seems remarkably efficient. However, the effectiveness of SPSA is also known to be dependent on its starting point, and its internal optimization parameters [23]. Moreover, SPSA is known to sometimes converge on local optimum solutions instead of global optimum solutions [24]. Table 5 illustrates the effectiveness of discrete and continuous SPSA optimization under several sets of internal optimization parameters. These results imply that continuous SPSA might be more efficient than discrete SPSA, but that only certain internal optimization parameters can produce a desirable result. In addition the global optimum solution (1.2%), which was revealed by Table 3, was not located.

k	ck	ak	a	c	alpha	gamma	theta	delta	xplus	yplus	xminus	yminus	grade	theta
1	0.20	0.20	0.2	0.2	0.6	0.1	58.0	-1	46.4	13.6	69.6	3.9	-24.3	62.9
2	0.19	0.13	0.2	0.2	0.6	0.1	62.9	-1	52.0	9.9	73.7	4.9	-13.4	64.6
3	0.18	0.10	0.2	0.2	0.6	0.1	64.6	-1	54.2	8.5	75.0	5.3	-8.9	65.5
4	0.17	0.09	0.2	0.2	0.6	0.1	65.5	1	75.6	3.9	55.4	6.2	-6.6	66.1
5	0.17	0.08	0.2	0.2	0.6	0.1	66.1	-1	56.2	5.7	76.0	3.9	-5.3	66.5
6	0.17	0.07	0.2	0.2	0.6	0.1	66.5	-1	56.8	6.5	76.2	4.9	-4.8	66.8
7	0.16	0.06	0.2	0.2	0.6	0.1	66.8	1	76.4	4.9	57.3	5.7	-2.4	67.0
8	0.16	0.06	0.2	0.2	0.6	0.1	67.0	1	76.4	4.9	57.6	5.2	-0.9	67.0
9	0.16	0.05	0.2	0.2	0.6	0.1	67.0	1	76.4	3.8	57.7	4.3	-1.6	67.1
10	0.16	0.05	0.2	0.2	0.6	0.1	67.1	1	76.3	5.1	57.9	5.2	-0.3	67.1
11	0.16	0.05	0.2	0.2	0.6	0.1	67.1	1	76.3	5.1	58.0	5.1	0.0	67.1
12	0.16	0.04	0.2	0.2	0.6	0.1	67.1	-1	58.1	4.3	76.2	4.6	1.0	67.1
13	0.15	0.04	0.2	0.2	0.6	0.1	67.1	-1	58.1	5.1	76.1	5.1	0.0	67.1
14	0.15	0.04	0.2	0.2	0.6	0.1	67.1	-1	58.2	4.3	76.0	4.6	1.0	67.1
15	0.15	0.04	0.2	0.2	0.6	0.1	67.1	1	75.9	4.6	58.2	4.3	1.0	67.0
16	0.15	0.04	0.2	0.2	0.6	0.1	67.0	1	75.8	4.6	58.2	4.3	1.0	67.0
17	0.15	0.04	0.2	0.2	0.6	0.1	67.0	-1	58.2	5.1	75.7	5.1	0.0	67.0

Table 4: Sample SPSA calculations for the continuous case.

param_a	0.05	0.1	0.2	0.4	0.05	0.05
param_c	0.2	0.2	0.2	0.2	0.1	0.3
discrete	2.6	2.9	2.1	4.5	1.7	3.2
continuous	3.5	1.6	2.6	3.5	1.8	1.7

Table 5: SPSA optimization results for the discrete and continuous cases.

In summary, synthetic network comparison results revealed similar effectiveness between DBF and SPSA optimization. Both methods were able to achieve objective function values in the range of 1.6 to 1.8% within a relatively low number of trial simulation runs. Unlike SPSA, DBF optimization would be guaranteed to locate the global optimum solution upon requesting a sufficiently high number of trial simulation runs. SPSA appears to reliably find a local optimum solution within the fewest number of trials, but only if the internal parameters are set properly. It is unknown whether SPSA could have located the global optimum solution by using other sets of internal parameters, or with overwhelming numbers of trial simulations. Based on the quick convergence shown by Table 4, it seems unlikely that an overwhelming number of trials would allow SPSA to escape the local optimum solution; but some research [6, 25] indicates there are special and/or customized ways SPSA can be implemented, to facilitate escaping local optima.

4 REAL-WORLD CASE STUDY

Following the synthetic network tests, more calibrations were performed with real-world data, to further assess DBF and SPSA. Figure 6 illustrates two freeway interchanges along I-95 (near Jacksonville, FL), with three mainline lanes. Mainline and ramp free-flow speeds were 113 and 40 km/h. The model contained twelve 15-minute time periods, with 7% trucks. Vehicle speeds were collected on twelve days during the peak hour (periods 5 through 8). Locations "C", "D", and "E" exhibited the largest discrepancy between real speeds (averaged over twelve days), and simulated speeds, in km/h. Table 6 illustrates the peak-hour flow rates.

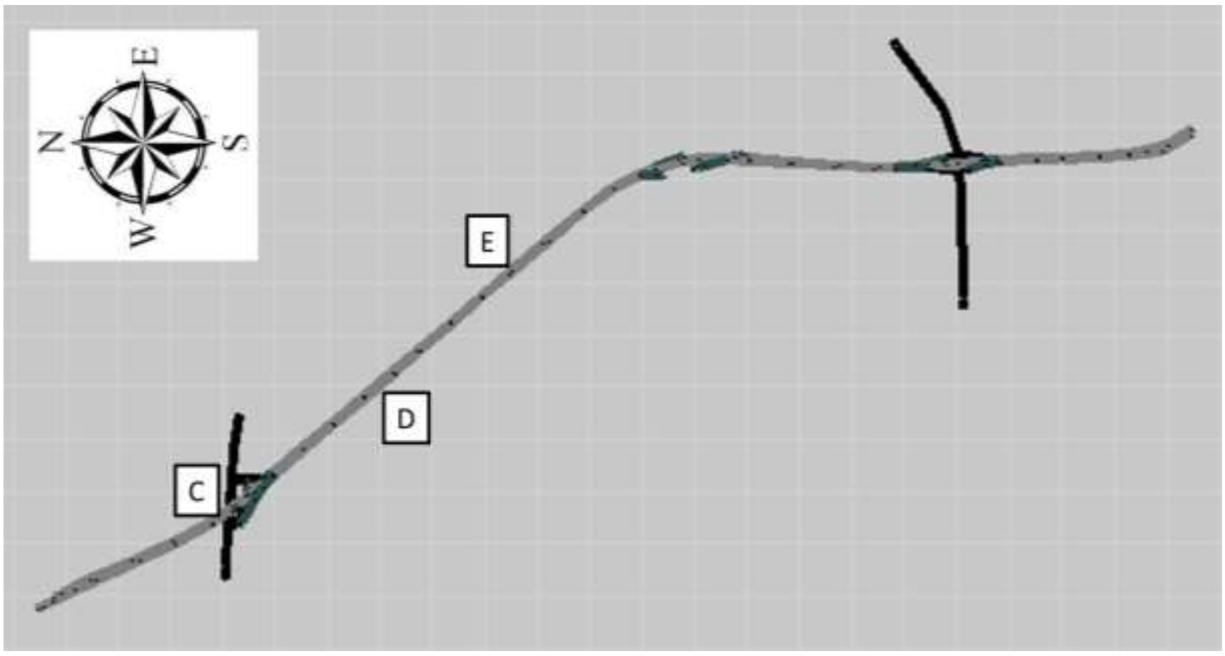


Figure 6: I-95 network roadway geometry.

	TP5	TP6	TP7	TP8
SB mainline	2644	3240	2780	3032
SB off-ramp at "C"	1144	1456	1024	1084
NB on-ramp at "C"	1208	992	1124	1252
NB mainline	3800	3488	3468	3484

Table 6: I-95 network flow rates (in vehicles per hour) across four time periods.

At location C, the discrepancy in speeds was large in both northbound and southbound (NB and SB) freeway directions (i.e. upstream of the SB off-ramp, and downstream of the NB on-ramp). Location D represents a SB segment, whereas E is a NB segment. After entering field-measured speeds for the four locations and time periods into the output data UI, SASCO showed an overall difference of 4.3% versus simulated results. Differences were actually highest (10-20%) in location C, and 0-5% elsewhere. Preliminary stochastic analysis, involving 8 simulation runs with 8 different sets of random number seeds, indicated that this percentage difference was closer to 4.5% than 4.3%. Thus, the original objective function value for this series of test optimizations was considered to be 4.5%.

4.1 Preliminary Univariate Calibration

A nearby interchange exists just north of location C, but was not included in the simulation network dataset. Because of this, and because speeds were known to be inaccurate in location C, there was reason to believe drivers exhibited more aggressive car-following behavior between those interchanges. Thus a preliminary calibration of car-following sensitivity multipliers was performed on the seven SB segments upstream of location C. SPSA optimization was able to reduce the objective function value from 4.5 to 3.5% within 13 iterations (26 simulations). DBF optimization reduced the objective function value from 4.5 to 3.0% within a similar number of simulations. The optimum car-following multiplier recommended by SPSA was 72.7%, whereas 74.0% was produced by DBF. In both cases, "post-stochastic" analysis (same as preliminary stochastic analysis, but with optimum car-following multipliers) increased the percent difference to 3.6%. Thus the end result of preliminary calibration was a car-following sensitivity multiplier of 74.0% on seven segments, and an objective function value of 3.6%. Following these runs it was decided no further localized (i.e., link-specific) calibrations were justified, and that all subsequent calibrations would involve "global" (i.e., network-wide) input parameters.

4.2 Intermediate Sensitivity Analysis

At this stage there were 16 global input parameters available for possible calibration, but simultaneous DBF calibration of 16 inputs would require overwhelming computer run times. Moreover, it was believed that simultaneous SPSA calibration of 16 input parameters might also lead to excessive run times and/or highly suboptimal solutions. As such, some intermediate sensitivity analysis (SA) runs were performed, to gauge the relative importance of the remaining 16 input parameters. By exhaustively simulating every trial value, every DBF optimization produces SA results. Thus the 16 inputs were examined one by one, with 16 runs.

This series of DBF SA runs implied that 9 out of 16 input parameters reduced the objective function value (i.e., percent difference between simulated and field-measured results), also known as "diff", by more than 0.1%. However one of these input parameters was the Desired Ramp Free-Flow Speed, whose optimum value was higher than 97 km/h. Although the extremely high ramp speeds would allow simulated mainline speeds to match field-measured mainline speeds more closely, these ramp speeds would be impractical in the real world. After discarding this input parameter there were 8 remaining, but post-stochastic analysis implied that only 5 of them actually reduced diff by more than 0.1%:

- Vehicle Entry Headway
- Time to Complete a Lane Change
- Minimum Entry Headway
- Percentage of Cooperative Drivers
- Off-Ramp Reaction Distance

Optimizing Vehicle Entry Headway with SPSA would be difficult. These headways consist of both discrete words (i.e. Normal, Erlang, Uniform), and numeric values associated with each word. The combination of DBF searching and the input parameter database is ideal for optimizing unusual inputs like this, but SPSA seems designed to optimize numbers only. Therefore the Vehicle Entry Headway was optimized by DBF SA, producing a diff value of 3.1% within 7 simulations. Post-stochastic analysis then decreased diff from 3.1 to 3.0%. Thus the end result of intermediate SA was a new Vehicle Entry Headway distribution, an objective function value of 3.0%, and four remaining input parameters to be optimized.

4.3 Final Multivariate Calibration

At this stage there were 4 global input parameters remaining for calibration. Only DBF searching was used in the first set of multivariate calibrations. Following the prior calibration of Vehicle Entry Headway, subsequent DBF SA runs implied that Minimum Entry Headway and Percentage of Cooperative Drivers would no longer reduce diff by more than 0.1%. Therefore, Time to Complete a Lane Change (TTCLC) and Off-Ramp Reaction Distance (ORRD) were simultaneously calibrated with 7*8=56 simulation runs, such that the entire optimization run required less than one hour. This optimization run located a diff value 1.7% at TTCLC of 4.5 seconds, with ORRD between 1050 and 1650 meters. Post-stochastic analysis then increased diff from 1.7 to 1.9%. Thus the end result of a typical DBF calibration would be an objective function value of 1.9%, with several input parameters optimized.

However in order to set the stage for detailed comparisons between DBF and SPSA, TTCLC and ORRD were then analyzed with 17*18=306 simulation runs, requiring approximately four hours. The resulting Table 7 below shows optimal ranges for TTCLC (between 5.5 and 7.5) and ORRD (between 1200 and 1800), but also reveals the inconsistent diff values caused by stochastic noise. According to Table 7, the quick DBF calibration based on only 56 runs did a fairly good job of locating a near-optimum solution within a reasonable time frame.

Off-Ramp Reaction Distance (m)	Time to Complete a Lane Change (sec)																	
	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.0	avg
450	4.0	3.4	3.4	2.8	2.8	2.5	4.5	2.4	2.7	2.3	2.6	2.1	2.4	2.3	2.5	2.1	2.9	2.8
600	3.8	3.3	3.1	3.3	2.3	2.5	2.0	2.1	2.0	1.8	2.5	1.9	2.5	2.4	2.3	2.3	2.0	2.5
750	3.0	2.7	2.5	2.5	2.5	2.0	2.2	2.0	2.4	1.8	2.3	2.0	2.2	2.4	2.5	2.3	2.1	2.3
900	2.9	2.3	2.4	2.0	2.3	2.0	1.9	1.9	1.8	2.1	2.4	2.0	2.1	2.2	1.7	1.9	2.0	2.1
1050	2.7	2.5	4.5	2.1	2.0	1.7	1.8	2.4	1.5	1.8	1.8	1.7	2.1	1.9	2.0	2.3	2.5	2.2
1200	3.0	2.1	2.4	2.2	1.9	1.7	1.8	1.8	1.8	2.1	1.9	1.6	1.8	2.6	1.8	1.9	2.2	2.0
1350	2.4	2.3	2.0	2.1	2.0	1.9	2.2	1.7	1.6	2.3	1.7	1.8	1.8	2.0	2.0	2.0	2.2	2.0
1500	2.7	2.0	2.0	2.7	2.1	1.9	2.6	1.9	1.6	2.0	1.9	1.9	1.9	1.8	1.6	2.0	2.0	2.0
1650	2.8	2.3	2.1	1.8	1.8	1.7	2.1	1.9	1.8	1.7	1.9	1.9	2.4	1.8	2.1	2.1	2.0	2.0
1800	2.5	2.2	2.0	2.0	2.0	2.0	2.1	2.1	1.6	2.0	1.8	1.6	2.3	2.0	2.3	1.9	2.1	2.0
1950	2.5	2.7	2.1	2.0	1.9	2.1	2.2	1.6	2.0	1.8	2.2	2.0	2.0	2.2	2.3	2.3	2.2	2.1
2100	2.6	2.0	2.2	1.8	2.0	2.1	1.9	2.2	2.0	1.6	2.3	2.2	2.0	1.7	2.1	2.2	2.4	2.1
2250	2.6	2.5	2.3	2.1	2.0	2.0	2.1	1.9	1.9	2.3	2.3	1.9	1.8	2.2	1.9	2.5	2.0	2.1
2400	2.6	2.4	2.4	2.3	2.3	2.0	2.3	2.2	2.5	2.0	2.0	2.0	2.4	2.1	1.9	2.1	2.3	2.2
2550	2.7	2.5	2.8	2.2	2.1	2.1	1.9	1.8	2.3	1.9	1.9	2.1	2.3	2.3	3.3	2.6	2.3	2.3
2700	2.9	2.9	2.3	2.1	2.3	4.7	1.9	1.9	1.8	2.3	2.2	2.2	2.4	2.4	2.5	2.2	2.2	2.4
2850	3.1	2.2	2.4	2.1	3.7	1.9	1.8	2.4	1.9	2.4	2.3	2.0	1.9	2.4	2.0	2.3	1.9	2.3
3000	2.6	2.5	2.6	2.1	2.1	2.4	2.0	1.6	1.8	2.3	2.1	2.3	3.3	2.2	2.0	2.0	2.1	2.2
avg	2.9	2.5	2.5	2.2	2.2	2.2	2.2	2.0	1.9	2.0	2.1	2.0	2.2	2.2	2.2	2.2	2.2	

Table 7: Objective function ("diff") values for the Jacksonville case study.

Figure 7 illustrates the optimized "theta" values over 100 iterations (200 simulations) of SPSA. The top two graphs were seeded with good starting points near the optimal range, and the bottom two graphs were seeded with bad starting points. Each graph compares four optimization runs. In "2P" and "2N", TTCLC and ORRD were simultaneously optimized with wide and narrow range limits, respectively. In "4P" and "4N", Minimum Entry Headway and Percentage of Cooperative Drivers were simultaneously optimized along with TTCLC and ORRD, using both wide and narrow range limits. These tests were performed because it was believed SPSA would be more efficient under narrow input parameter ranges (e.g. 4.5-7.5 instead of 2.0-10.0), and when simultaneously optimizing fewer input parameters (e.g. TTCLC and ORRD by themselves, instead of TTCLC and ORRD with two additional inputs).

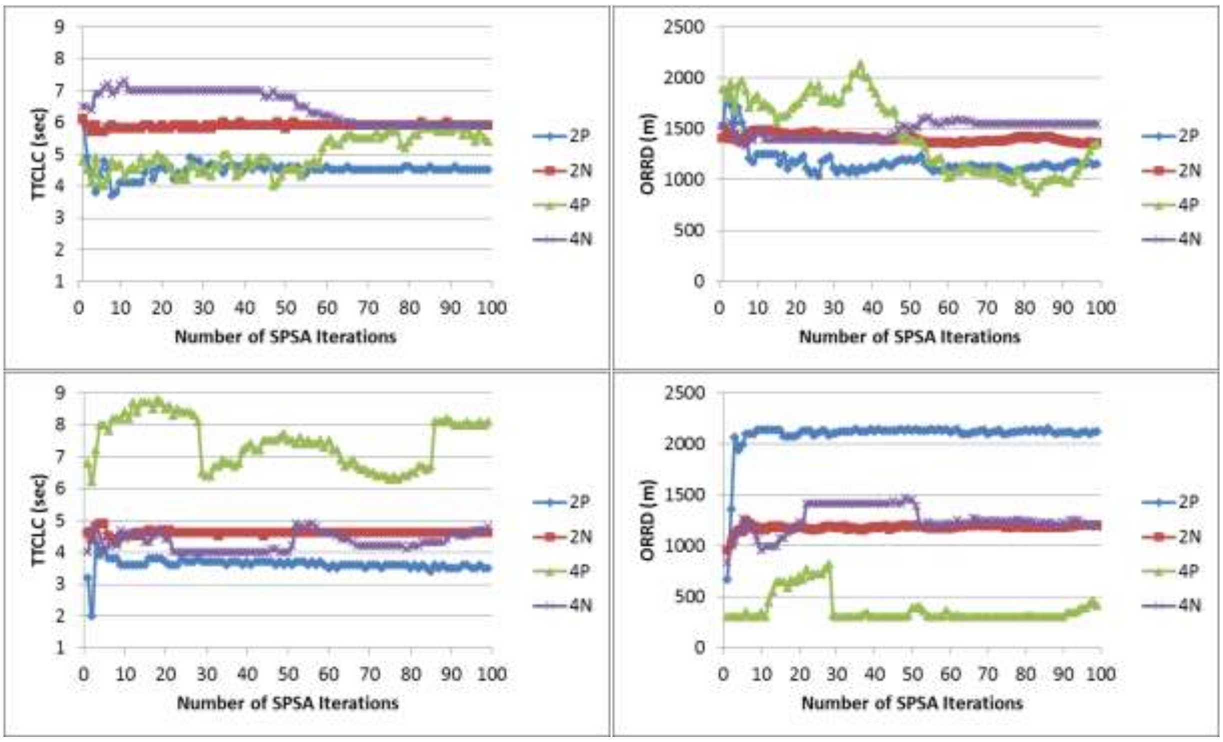


Figure 7: Optimized (theta) SPSA values for TTCLC and ORRD.

Based on Table 7 results showing optimal ranges for TTCLC (5.5-7.5) and ORRD (1200-1800), Figure 7 implies the "bad" starting points (two bottom graphs) indeed cause less efficient optimization, at least during those first 200 simulations. Regarding the comparison between optimizing 2 or 4 inputs, it appears SPSA handled them with similar efficiency, although it is noted the two "extra" parameters were known to have little impact on results. Finally the narrow ranges were usually more effective than wide ranges, implying that SASCO's range-setting features would likely augment the efficiency of SPSA.

Figure 7 results imply that bad starting points can make SPSA highly ineffective, but this is not the only way in which SPSA can be evaluated or applied. Table 4 from the synthetic network study shows the trial values ("xplus", "xminus") are quite different than "theta", especially in early iterations, and often produce very good results. Figure 8 shows that all SPSA optimization runs produced trial values resulting in diff values below 2.0%, in 30 simulations or less. This was true even when simultaneously optimizing 4 inputs, having wide ranges and bad starting points. Thus the end result of DBF or SPSA calibration was a diff reduction from 4.5% to below 2.0%. Moreover, location C diff values dropped from 10-20% to below 5%.

One caveat is that several optimizations terminated prematurely, and had to be re-run with better internal parameter values. Essentially the "a" parameter provided a step-size for theta, whereas the "c" parameter provided a step-size for trial values xplus and xminus. For simple optimizations (e.g., calibrating two inputs with narrow ranges), small step-sizes could cause premature termination. For more complex optimizations large step-sizes could cause excessive iterations, or prevent optimization completely. In the end, it appears SPSA is capable of better efficiency than DBF if 1) proper internal parameters are chosen, 2) trial values are used instead of theta values, and 3) convergence is not required. However SPSA can be inefficient when not applied properly, is less suitable for sensitivity analysis, and cannot explicitly optimize interdependent or non-numeric inputs.

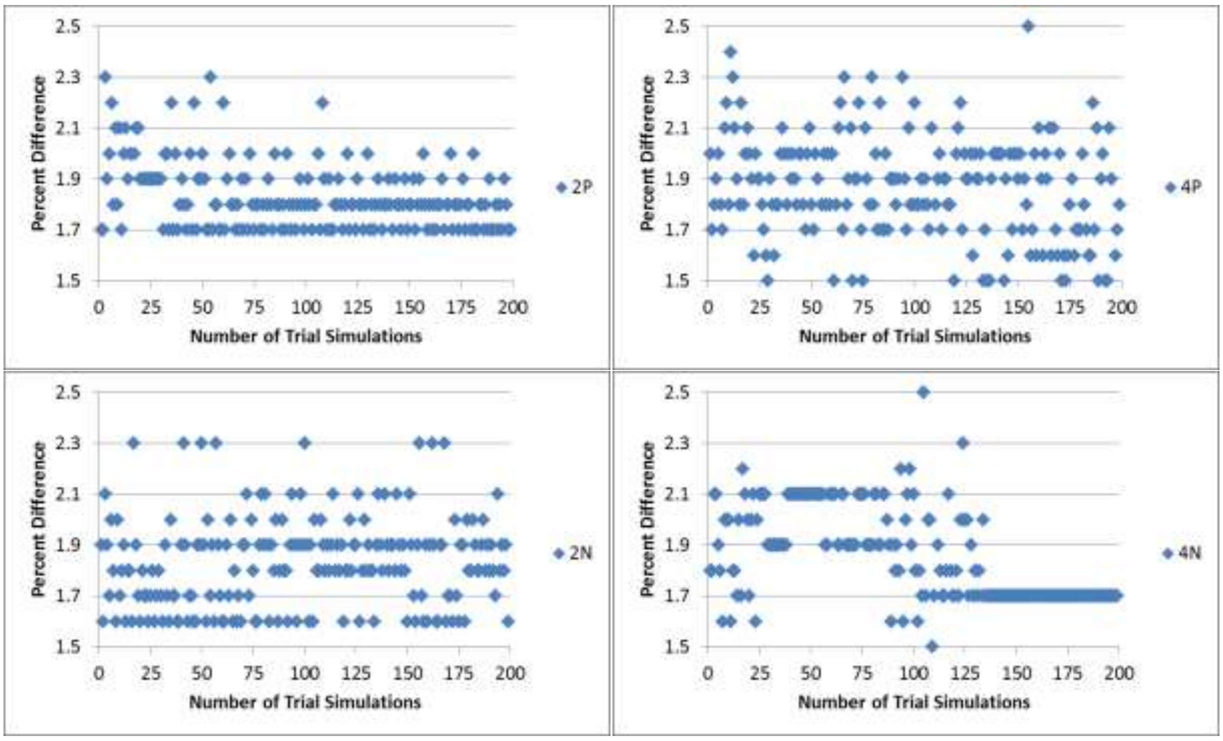


Figure 8a: Trial (xplus, xminus) SPSA objective function (diff) values (good starting points).

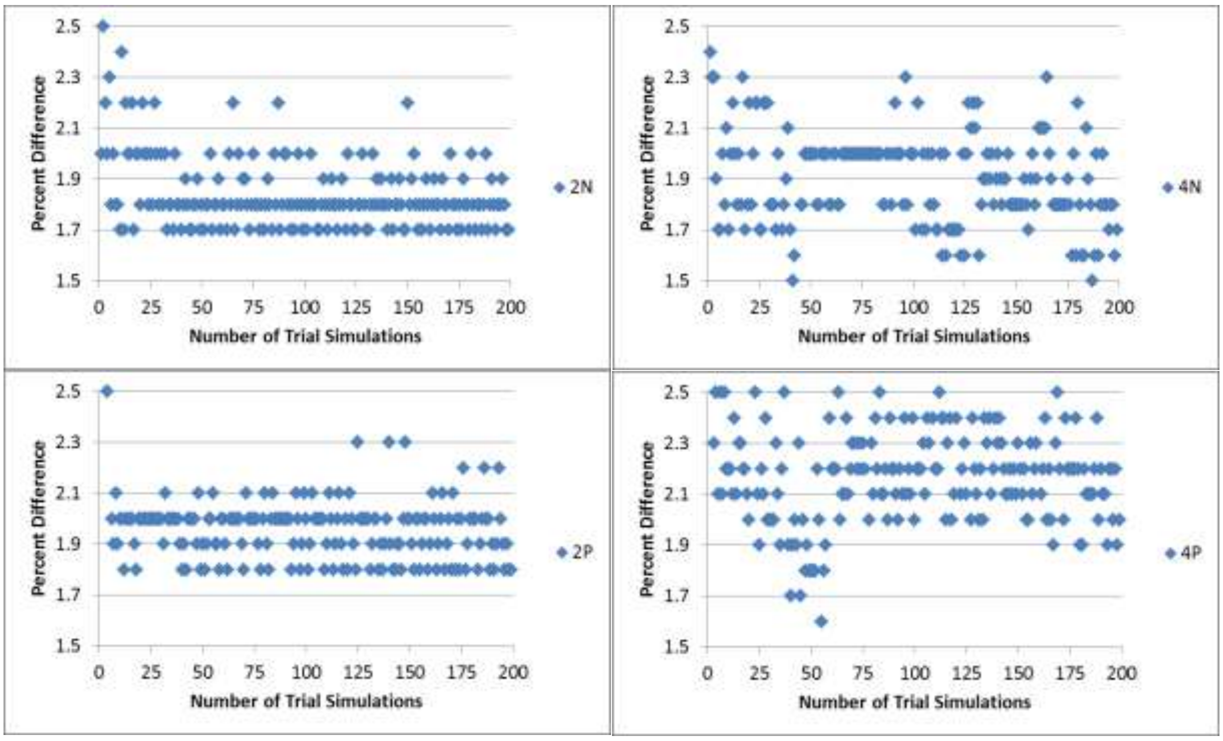


Figure 8b: Trial (xplus, xminus) SPSA objective function (diff) values (bad starting points).

4.4 Simultaneously Optimizing all Inputs with SPSA

In the I-95 Jacksonville case study just shown, input parameters were calibrated in a systematic fashion. First, preliminary univariate calibration was used to reflect increased driver aggression between closely-spaced interchanges. Then, sensitivity analysis was used to determine a small number of high-impact input parameters. Finally, multivariate calibration was used to optimize high-impact inputs. This procedure is necessary for DBF, which cannot simultaneously optimize large numbers of inputs in a reasonable time frame. However, SPSA can do it. The question is how well can it do it, and are larger optimizations effective?

The simulator used in the case study was FRESIM, having approximately 16 input parameters relevant to the calibration process. Three of these inputs were interdependent distributions of numbers and/or discrete words; which could not be optimized by SPSA without "encoding algorithms", which were not developed during this research. One of these inputs was Desired Ramp Free-Flow Speed, whose optimum value (113 km/h) was judged as unrealistic during the DBF calibration. Therefore, a new set of SPSA optimization experiments was performed; to calibrate the remaining 12 inputs simultaneously, instead of sequentially.

For this experiment a discrete form of SPSA was applied, as there was insufficient time to update the software for continuous SPSA optimization of 12 inputs. Discrete SPSA might be less efficient than continuous, as evidenced by results from the synthetic network study. Secondly, local calibration was not performed near location C, as car-following sensitivity was instead calibrated globally. Third, the number of simulations was doubled compared to the sequential optimizations. Finally, the internal SPSA random number (SRN) for generating "delta" (+1 or -1) values was also varied, to assess variability within the optimization process.

Results of the prior experiments (sequential optimization) indicated that trial values (e.g., from Figure 8) provide the fastest way to obtain good solutions. Previous results also showed that a "good solution" would mean reducing the original diff value (4.5%) to somewhere below 2.0%. Table 8 illustrates the number of trial values (simulations) needed before locating any solution below 2.0%. A total of 6 optimization runs (2400 simulations) were performed, based on 3*2 =6 combinations of internal ("a" and "c") parameter values and SRN values.

SRN	7783	7783	7783	7781	7781	7781
param_a	0.1	0.2	0.3	0.1	0.2	0.3
param_c	0.1	0.2	0.3	0.1	0.2	0.3
# of runs	146	106	189	30	28	n/a*

Table 8: Number of simulations before SPSA located diff below 2.0% when simultaneously optimizing 12 inputs.

Table 8 illustrates the extent to which SPSA might be affected by its internal parameter settings. In the best-case scenario, a solution below 2.0% was located after only 28 simulations, which is drastically more efficient than the DBF process. However under a different set of SRN, changing delta values caused the number of required simulations to increase from 28 to 106. Furthermore, a slightly different set of "a" and "c" values caused the number of simulations to increase from 106 to 189. Finally, one combination of internal values prevented any solution below 2.0% from being located within 400 simulations, although a reasonably* good solution (2.1%) was found within 28 simulations. Similar to the sequential results, these simultaneous optimization results suggest that SPSA has the potential for extremely fast calibration. Unfortunately this efficiency is dependent on internal parameter settings; and the impact of random delta values during this experiment, revealed by changing SRN, was disconcerting.

5 CONCLUSIONS

Because the existing (non-automated) methods of calibration have been difficult and/or inadequate, significant research has been performed in the area of software-assisted calibration techniques. However much of this research has not provided the level of flexibility and practicality typically required by real-world engineers. With this in mind, a patent-pending (US 61/859,819) architecture for assisted calibration was developed to maximize practicality, flexibility, and ease-of-use. This architecture was extended to support applications of sensitivity analysis and optimization, leading to the "SASCO" acronym.

Some of the prior art calibration techniques are ideal for origin-destination flow modeling, while others employ pattern matching of vehicle trajectories and/or speed-flow relationships. A third family of calibration method employs simulation-based optimizations (SO), which have usually relied on heuristic searching methods. However the SO-based calibration methods have not gained commercial popularity for calibration of traffic simulations, most likely because their heuristic methods tend to require an excessive and unpredictable number of trial simulations. To achieve the necessary reduction in trial simulations, the SASCO architecture allows for quick and easy reduction of the search space, by defining trial values and range limits imposed upon SO-based calibration. Once the search space has been reduced through SASCO, the directed brute force (DBF) and Simultaneous Perturbation Stochastic Approximation (SPSA) methods appear to be good candidates for optimizing the input parameters.

Testing was done to assess DBF and SPSA qualities, so they could be applied in the right situations. While SPSA performed true to its reputation by optimizing very efficiently, its effectiveness was dependent on internal ("a" and "c") parameter values, randomly-generated delta (+1 and -1) values, and starting points. Moreover, trial (xplus, xminus) values seemed much more helpful than the so-called optimum (theta) values, and SPSA usually found local instead of global optimum solutions. By contrast, DBF guarantees locating the best solution within a user-defined search space, although these search spaces sometimes fail to include global optimum solutions. DBF can also optimize advanced (interdependent numeric distributions and discrete words) inputs. Finally, DBF is ideal for sensitivity analysis, and can help determine which inputs to calibrate.

Given these complementary attributes, the overall calibration process could conceivably be more efficient with the two methods applied in tandem. For example, SPSA might be ideal for locating a local optimum solution more quickly than DBF would. Using this local optimum solution as a starting point, DBF could then perform a more detailed and reliable scan of the surrounding area, to locate global optimum solutions.

Follow-up studies should examine more optimization methods, validation following calibration, calibration of output distributions, more real-world corridors, and more simulators. Moreover it would help to systematically determine the best SPSA internal parameter values for various network conditions. Regardless of which optimization method is selected, the SASCO architecture appears to offer a new and practice-ready level of efficiency, for software-assisted calibration.

REFERENCES

[1] Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software, Publication No. FHWA-HRT-04-040, 2004.

[2] Bloomberg, L., M. Swenson, and B. Haldors, *Comparison of Simulation Models and the Highway Capacity Manual*, Preprint, Annual Meeting, TRB, Washington, DC, 2003.

[3] Brackstone, M., Montanino, M., Daamen, W., Buisson, C., Punzo, V., "Use, Calibration, and Validation of Traffic Simulation Models in Practice: Results of Web-Based Survey", Paper #12-2606, Transportation Research Board, Washington, D.C., 2012.

[4] Vaughn, B., and McGregor, A., "Guidelines for Microscopic Simulation Modeling", Highways Agency, 2007.

[5] Traffic Analysis Toolbox Volume IV: Guidelines for Applying CORSIM Microsimulation Modeling Software, Publication No. FHWA-HOP-07-079, 2007.

[6] Lee and Ozbay, "New Calibration Methodology for Microscopic Traffic Simulation Using Enhanced Simultaneous Perturbation Stochastic Optimization Approach", *Transportation Research Record: Journal of the Transportation Research Board, No. 2124,* Transportation Research Board of the National Academies, Washington, D.C., 233-240, 2009.

[7] J. Lee, B. Park, J. Won, and I. Yun, "A Simplified Procedure for Calibrating Microscopic Traffic Simulation Models", Proceedings of the 92nd TRB Annual Meeting Compendium DVD, Washington D.C., 2013.

[8] Menneni, S., Sun, C., and P. Vortisch, "Microsimulation Calibration Using Speed-Flow Relationships", *Transportation Research Record: Journal of the Transportation Research Board, No. 2088,* Transportation Research Board of the National Academies, Washington, D.C., 1-9, 2008.

[9] Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H., and Wen, Y., "Calibration of Microscopic Traffic Simulation Models: Methods and Application", *Transportation Research Record: Journal of the Transportation Research Board, No. 1999,* 198-207, 2007.

[10] Ma, J., Dong, H., and Zhang, H.M., "Calibration of Microsimulation with Heuristic Optimization Methods", *Transportation Research Record: Journal of the Transportation Research Board, No. 1999,* 208-217, 2007.

[11] TRB Research Needs Statements (http://rns.trb.org/dproject.asp?n=15590) accessed on February 28th, 2014.

[12] Lee and Ozbay, "Calibration of Traffic Simulation Models Using Simultaneous Perturbation Stochastic Approximation (SPSA) Method extended through Bayesian Sampling Methodology", Ph.D. Dissertation, Rutgers University, New Brunswick, New Jersey, 2008.

[13] Paz, A., Molano, V., and Khan, A., "Calibration of Micro-Simulation Traffic-Flow Models Considering All Parameters Simultaneously", Proceedings of the 93rd Transportation Research Board Annual Meeting, 2014.

[14] Law, A., and McGomas, M., "Simulation-Based Optimization", Proceedings of the 2000 Winter Simulation Conference, 2000.

[15] Foy, M., Benekohal, R., and D. Goldberg, "Signal Timing Determination using Genetic Algorithms", *Transportation Research Record: Journal of the Transportation Research Board, No. 1365,* Transportation Research Board of the National Academies, Washington, D.C., 1992.

[16] R. John Koshel, "Enhancement of the downhill simplex method of optimization", *Proc. SPIE* 4832, International Optical Design Conference, 270, 2002.

[17] http://www.alignstar.com/support/clients/webhelpv5/Simulated_Annealing.htm, accessed on 11/27/2013, TTG Incorporated

[18] Hale, D.K., TRANSYT-7F Users Guide. McTrans Center, University of Florida, Gainesville, FL, 2009.

[19] J. Bang-Jensen, G. Gutin and A. Yeo, "When the greedy algorithm fails", Discrete Optimization 1, 121-127, 2004.

[20] Kim, S.J., Kim, W., and L.R. Rilett, "Calibration of Microsimulation Models Using Nonparametric Statistical Techniques", *Transportation Research Record: Journal of the Transportation Research Board, No. 1935,* 111-119, 2005.

[21] Ciuffo, B. and V. Punzo, "No Free Lunch Theorems Applied to the Calibration of Traffic Simulation Models", IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 2, 553-562, April 2014.

[22] Spall, J. C., "Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization", IEEE Vol. 34, No. 3, 817-823, 1998.

[23] Cipriani E., Florian M., Mahut M. And M. Nigro, "A gradient approximation approach for adjusting temporal origin-destination matrices", Transportation Research C, 19(3), 270-282, 2011.

[24] Lu, X., Lee, J., Chen, D., Bared, J., Dailey, D., and Shladover, S., "Freeway Microsimulation Calibration: Case Study Using Aimsun and VISSIM with Detailed Field Data", Proceedings of the 93rd Transportation Research Board Annual Meeting, 2014.

[25] Gerencser, L., Vago, Z., and Hill, S., "The Magic of SPSA", PAMM (Proc. Appl. Math. Mech.) **7**, 1062501-1062502, 2007.