

Calibration of Microscopic Traffic Flow Simulation Models Using a Memetic Algorithm with Solis and Wets Local Search Chaining (MA-SW-Chains)

Carlos Cobos¹(✉), Carlos Daza¹, Cristhian Martínez¹,
Martha Mendoza¹, Carlos Gaviria², Cristian Arteaga²,
and Alexander Paz²

¹ Universidad del Cauca, Popayán, Colombia
{ccobos, carlosdazar, cristhianm,
mmendoza}@unicauca.edu.co

² University of Nevada, Las Vegas, USA
{cgaviria, carteaga, apaz}@unlv.edu

Abstract. Traffic models require calibration to provide an adequate representation of the actual field conditions. This study presents the adaptation of a memetic algorithm (MA-SW-Chains) based on Solis and Wets local search chains, for the calibration of microscopic traffic flow simulation models. The effectiveness of the proposed MA-SW-Chains approach was tested using two vehicular traffic flow models (McTrans and Reno). The results were superior compared to two state-of-the-art approaches found in the literature: (i) a single-objective genetic algorithm that uses simulated annealing (GASA), and (ii) a stochastic approximation simultaneous perturbation algorithm (SPSA). The comparison was based on tuning time, runtime and the quality of the calibration, measured by the GEH statistic (which calculates the difference between the counts of real and simulated links).

Keywords: Calibration · Local search chaining · Solis and wets · Traffic flow simulation · Single-objective optimization · Memetic algorithm

1 Introduction

The last decade has observed an increase in the use of microscopic traffic flow simulation for a wide range of purposes. Calibration of the simulation models is critical to ensure validity and significance of the analysis. Calibration requires field measurements to be compared to the corresponding simulation outputs [1].

Microscopic traffic flow simulation models include many parameters that are adjusted to capture the required behavior. General principles for calibration of a microscopic traffic flow simulation models are summarized as follows: (1) pre-calibration, (2) selection of parameters to calibrate and the corresponding range, (3) selection of goodness of fit measures, (4) defining of a calibration procedure, and (5) execution of calibration, validation and analysis of results [2]. Optimization is recommended and typically used in step

4 (calibration procedure) for searching the numeric value to assign to each of the selected parameters to minimize the difference between simulation and field measurements.

Studies in the past have proposed a variety of methods and algorithms for conducting the search described [3]. The single sequence algorithm [4] and methodologies of stochastic approximation have been used for the simultaneous calibration of the model parameters [5, 6]. The two techniques used today include genetic algorithms (GA) [7, 8] and the simultaneous perturbation stochastic approximation algorithm (SPSA) [9, 10]. These techniques do not require information about the characteristics of the microscopic traffic flow simulation model [11]. That is, information about the methods used by the traffic flow simulation model to move vehicles and represent traffic flow dynamics is not required. Hence, these techniques assumed to work well with various types of traffic flow models.

Previous studies indicate that SPSA and GA provide similar results [10, 12]. However, SPSA requires less time and computational resources than GA. A memetic algorithm [13], GASA combines genetic algorithms with simulated annealing, which provides adequate results for low and medium complexity networks. A comparison with SPSA showed promising results [13].

There is no approach in the literature for the calibration of microscopic traffic flow models using memetic algorithms based on local search chains, as proposed in this paper. The first memetic algorithm based on local search chain approach was introduced in 2010 [14] and 2015 [15]. The algorithm showed great performance solving complex problems (continuous).

One of the interpretations of the No-Free-Lunch Theorem for optimization is that it is unknown which meta-heuristic can solve a problem in the best possible manner [16]. Thus, this paper investigates the performance of the MA-SW-Chains algorithm [14] compared to two alternative state-of-the-art algorithms found in the recent literature [11, 13].

The rest of the article is organized as follows: Sect. 2 defines the problem and explains how the MA-SW-Chains algorithm was adapted to solve the proposed calibration problem. Section 3 describes the experiments and results obtained with the MA-SW-Chains algorithm. Section 4 presents the comparison between the MA-SW-Chains, GASA, and SPSA algorithms. Finally, Sect. 5 provides conclusions and directions for future work.

2 Proposal

2.1 Formal Definition of the Problem

This study used the general principles described previously and in detail [2] for the calibration of microscopic traffic flow models. The objective function to minimize is the Root Mean Squared Normalized Error (RMSNE) [10] as shown in Eq. (1). Minimize RMSNE implies the minimization of the difference between actual and simulated volumes (vehicle counts) and actual and simulated vehicle speeds.

$$RMSNE = \frac{1}{\sqrt{N}} * \sum_{t=1}^T \left(W * \sqrt{\sum_{i=1}^N \left(\frac{V_{i,t} - \tilde{V}_{(\theta)i,t}}{V_{i,t}} \right)^2} + (1 - W) * \sqrt{\sum_{i=1}^N \left(\frac{S_{i,t} - \tilde{S}_{(\theta)i,t}}{S_{i,t}} \right)^2} \right) \quad (1)$$

where $V_{i,t}$ is the actual count for link i at time t and $\tilde{V}_{(\theta)i,t}$ is the simulated count for link i at time t . $S_{i,t}$ is the actual speed for link i at time t and $\tilde{S}_{(\theta)i,t}$ is the simulated speed for link i at time t . W is a weight used to assign value to the volume and the speed, N is the total number of links in the model, and T is the total number of time periods t .

Calibration criteria: The Federal Highway Administration (FHWA) guidelines for the calibration of CORSIM models were used in this study. The difference between the count for real and simulated links ought to be less than 5 for all links. The statistical GEH [2] was used, which should be less than 5 in at least 85 % of the links. The formula used to calculate the GEH statistic is provided by Eq. (2).

$$GEH = \sum_{i=1}^N \sqrt{\frac{2(V_i - \tilde{V}_{(\theta)i})^2}{V_i + \tilde{V}_{(\theta)i}}} \quad (2)$$

where V_i and $\tilde{V}_{(\theta)i}$ are, respectively, the actual and simulated count for link i .

2.2 Adaptation of the MA-SW-Chains Algorithm

In this paper, each agent (solution or individual) represents a parameter vector containing a solution for the optimization problem. In addition to a measure of fitness (efficacy), a bias vector indicates the direction of the local search. A standard deviation value (ρ) is used in the Gaussian Distribution within the Solis and Wets algorithm [17] to generate an offset vector, which helps to change the direction of the bias vector. The representation of an agent is illustrated in Fig. 1.

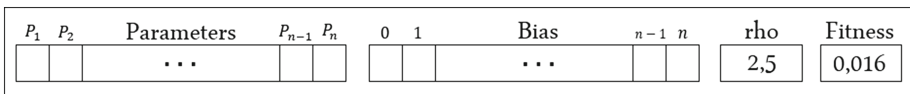


Fig. 1. Representation of an agent in the MA-SW-Chains algorithm.

Users (traffic engineers) need to select parameters that are calibrated for a specific model. These selected parameters (or record types) are part of the agent (see [11] for more details). These parameters take integer values based on restrictions defined in the CORSIM user manual [18].

The MA-SW-Chains algorithm uses a steady state GA to explore the solution search space. In this steady state GA, an (or several) offspring is created in every generation and competes with the agents in the population [19] to be a part of the new

population. The best agents are used to generate new populations through the genetic operators of the algorithm. The best agent of the population in each generation is selected for the process of exploitation using the Solis and Wets algorithm to refine the search to find better solutions.

The differentiating approach of MA-SW-Chains is the chaining of local searches, assigning a local search intensity (I_{str}) to each agent and exploiting the most promising agents more intensely. To adapt the local search intensity, local search can be applied several times to the same agent and their final parameters stored. These are used again when the agent is selected to be optimized locally, thereby creating the local search chain [14]. The stopping criterion for MA-SW-Chains is the maximum number of generations (a parameter of the algorithm), or the maximum execution time (an alternative parameter).

The MA-SW-Chains algorithm for supporting the calibration process is defined by the following steps:

- Step 1:** Generate the initial population randomly and apply restrictions to the agents to avoid non-permitted values. Each parameter to be calibrated has a previously defined range and type (in this case, integer). Fitness is calculated for all agents of the population based on RMSNE according to Eq. (1). The size of the population is a parameter previously defined by the user
- Step 2:** Generate new agents (the number of agents is determined by the algorithm parameters set by the user) using a tournament selection genetic operator, uniform crossover, and multigene mutation. These new agents comply with the restrictions to avoid non-permitted values
- Step 3:** Calculate fitness for new agents based on RMSNE and add to the population
- Step 4:** Given that the lowest value of RMSNE is sought, agents with high fitness values are removed from the population. As many as required agents are removed to maintain the population size defined by the user
- Step 5:** Select the best agent in the population (this agent has the lowest fitness measure because a minimum value of RMSNE is sought)
- Step 6:** If local search has been applied previously to this best agent, then the bias vector and standard deviation (ρ) of the local search are initialized using the bias vector and standard deviation (ρ) values of the agent selected
- Step 7:** If local search has not been applied previously to this best agent, the parameters of local search are initialized using default values, with the bias vector equal to the zero vector and the standard deviation (ρ) equal to half of the distance from the nearest neighbor to the best agent in the population [20]
- Step 8:** Apply the local search algorithm, Solis and Wets, to the best agent of the population with an intensity of I_{str} , which is the number of iterations the local search will run and is defined previously by the user

- Step 9:** On completion of the local search, the bias vector and standard deviation (ρ) of the local search should be set at the bias vector and standard deviation (ρ) of the agent to which the local search was applied. These parameters will be used in future local searches if the agent is selected again as the best of the population
- Step 10:** If the stop condition is not satisfied, then back to step 2

The pseudocode of the original MA-SW-Chains algorithm can be reviewed in [14]. Local searches are performed with the Solis and Wets algorithm, which is a random Hill-Climber with adaptive step size. The Solis and Wets algorithm works while the local search iterations are less than or equal to the parameter I_{str} . The pseudocode of the Solis and Wets algorithm is found in [20] and was adapted for the calibration process using the following steps:

- Step 1:** Initialize the offset vector (same size as the bias vector) with values from the Gaussian distribution, whose parameters are zero for both mean and standard deviation (ρ) according to the data sent from MA-SW-Chains
- Step 2:** Create a new agent (Agent 1), adding each element of the vector parameters of the current best agent to each element of the bias vector and each element of the offset vector. Values are repaired to this new agent so that it meets restrictions of range and data type. The fitness measure is then calculated for this new agent, based on RMSNE
- Step 3:** If Agent 1 has a lower fitness measure than the best current agent (the best solution to the problem), Agent 1 replaces the best agent. Each element of the bias vector is updated using the equation: $(0.2 * \text{bias vector} + 0.4 * (\text{offset vector} + \text{bias vector}))$, the number of successes is increased by one and the number of failures initialized at zero. The constants in the previous equation were taken from Molina et al. [20]
- Step 4:** If Agent 1 is not better than the best current agent, then a new agent (Agent 2) is created by subtracting each element of the parameter vector of the current best agent with each element of the bias vector and each element of the offset vector. This new agent is given its values to comply with the restrictions of range and type. The fitness measure is then calculated for this new agent using RMSNE
- Step 5:** If Agent 2 has a better (lower) fitness measurement than the current best agent, then the agent is replaced by Agent 2, each element of the bias vector is updated using the equation: $(\text{bias vector} - 0.4 * (\text{offset vector} + \text{bias vector}))$, the number of successes is increased by one and the number of failures is reset to zero
- Step 6:** If Agent 2 is not better than the current best agent, then the number of failures is increased by one and the number of successes is reset to zero
- Step 7:** If the number of successes is greater than two, then the value of the standard deviation (ρ) is duplicated and the number of successes reset to zero

- Step 8:** If not, i.e. the number of failures is greater than one, then the value of the standard deviation (ρ) is reduced to half and the number of failures is reset to zero
- Step 9:** If the value of the standard deviation (ρ) is less than one, then the value of the standard deviation is made to be equal to one (this is because there are calibration parameters that vary within a small range of integer values)
- Step 10:** If the stop condition is not met, go to step 1 taking the new value of the standard deviation (ρ) modified in steps 7, 8 or 9

Finally, the execution process of the calibration is performed with its respective validation and analysis of results. The results of this step are presented in the following sections.

3 Execution of Calibration and Results

The experimentation was performed using two CORSIM models previously studied [13]. The first model is considered low complexity due to the number of links that it contains (20). It is a hypothetical network provided by McTrans. The second experiment was performed using a real network, which corresponds to the Pyramid Highway in Reno, Nevada, United States. This model is considered medium complexity because it contains 126 links.

To perform the experiments, the MA-SW-Chains algorithm was implemented and included in the software tool, “Calibration Tool” [13], which provides four alternative algorithms: a memetic algorithm, which combines a genetic with simulated annealing (GASA), a memetic algorithm, which combines a genetic with Tabu search (GATS), a genetic algorithm without a local optimizer (GA), and the SPSA. In this research, the results obtained by the MA-SW-Chains algorithm were compared with the results from GASA and SPSA, which provided best results in the past.

3.1 Algorithm Parameters

Parameters used in MA-SW-Chains for the experimentation were as follows: Initial population 4, Percent of selection 60, Percent of crossover 75 and percent of mutation 7. The local search intensity parameter (I_{str}) of Solis and Wets was set to 30.

3.2 Experiments

The two models used in the experiments, volume and speed were considered simultaneously for the calibration. The Reno model included actual field measurements taken on 45 links of the model. For the field data of the McTrans model, default values were used. For each of the models, 50 independent runs of the algorithms were performed and the results averaged.

As a result, 50 independent runs on the McTrans model (Fig. 2 (a)) with an average of 0.094 was achieved in the objective function (RMSNE) with a standard deviation of 0.0545, which indicates an adequate quality in the calibration of the model. However, the behavior of the algorithm in the Reno model is much better (Fig. 2 (b)), with an average of 0.0955 was obtained in the objective function with a standard deviation of 0.0144. This indicates a greater stability in the behavior of the algorithm, obtaining in 78 % of the 50 runs a RMSNE < 0.10, which is promising for this model (this model is more complex).

The best RMSNE obtained in each model after running all experiments was 0.0163 for McTrans and 0.0813 for the Reno network. In Figs. 3 to 5 are the results obtained in the best experiment of each model.

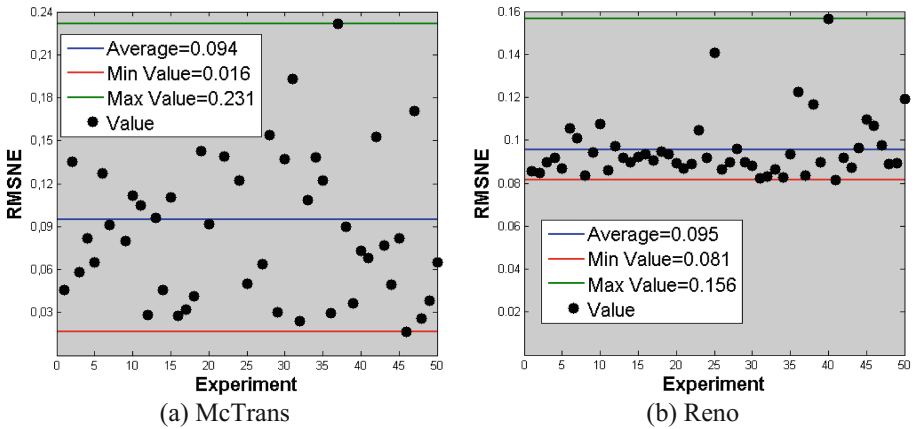


Fig. 2. The objective function for the two CORSIM models.

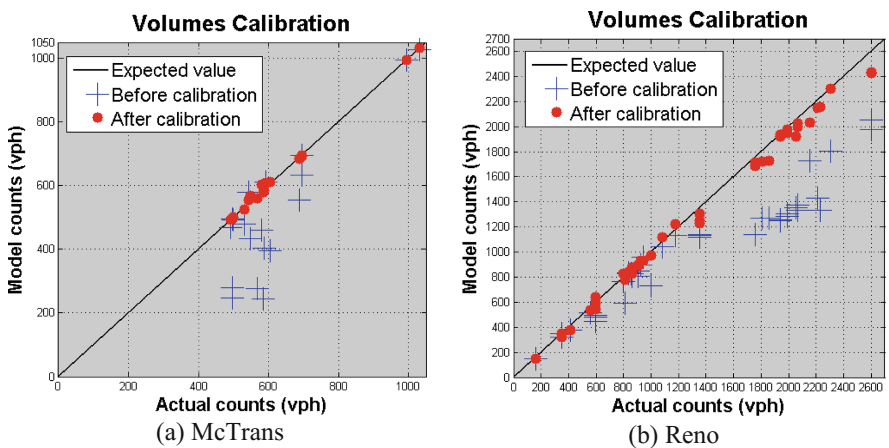


Fig. 3. Volume data before (blue) and after (red) calibration in the two models. (Color figure online)

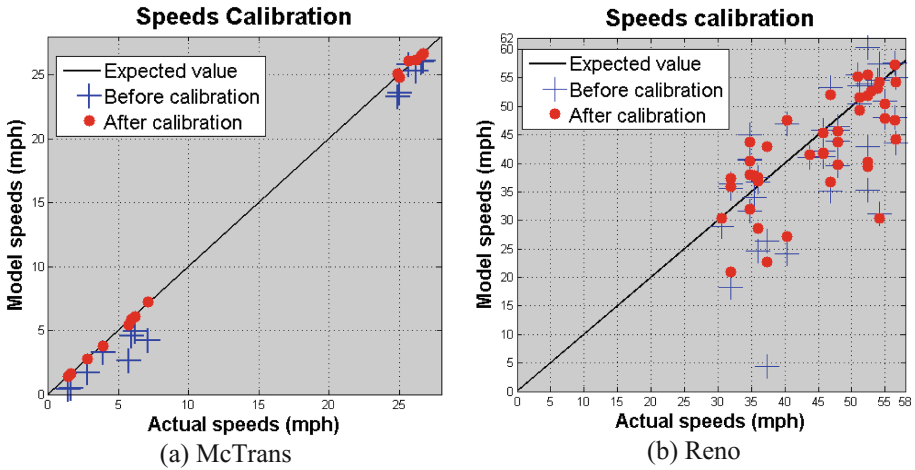


Fig. 4. Speed data before (blue) and after (red) calibration for the two models. (Color figure online)

Figure 3 (a & b) shows the volume (vehicles per hour, vph) obtained by the simulation before calibration (blue crosses) far from the actual data (which would place them on the diagonal line) for the two models. After performing the calibration process, the volume (red dots) approach the diagonal for the two models, which means that the simulated volume largely coincides with the actual.

The same occurs with the speed data (miles per hour, mph) in the two models Fig. 4 (a & b). However, the improvements in this case are not as significant as in the Reno model. This is because in the objective function (RMSNE) more weight was assigned to volume (70 %), compare to speed (30 %). Although the weight parameter can be modified, in these experiments greater weight was assigned to volume because there was a greater confidence in the volume data (real data).

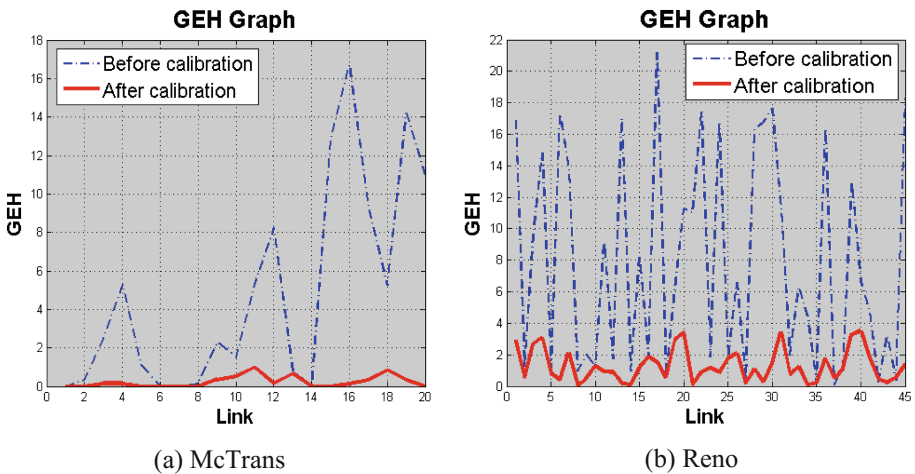


Fig. 5. GEH statistic for the two models. (Color figure online)

Figure 5 (a & b) shows the results of statistical GEH, in which the dotted blue line indicates the value before the calibration process and clearly shows many links above 5, indicating that the model is not yet calibrated. The thick red line indicates its value after calibration. Following calibration, a marked improvement was obtained in both models, in which 100 % of the links had a GEH < 5; thus, fulfilling the defined calibration criteria.

4 Comparative Analysis of Results (MA-SW-Chains Vs GASA and SPSA)

To demonstrate the benefits of MA-SW-Chains for the calibration of microscopic traffic flow simulation models, a comparison was made with the state-of-the-art algorithms GASA and SPSA [13]. Table 1 shows the results obtained by the three algorithms.

Table 1. Comparison of MA-SW-Chains, GASA and SPSA

Experiments/ Criteria	McTrans			Reno		
	MA-SW Chains	GASA	SPSA	MA-SW Chains	GASA	SPSA
Run time	9,56 min	9,53 min	9,55 min	59,45 min	43,55 min	29,30 min
RMSNE before	0.2911	0.2911	0.2911	0.2217	0.2217	0.2217
RMSNE after	0.0163	0.0192	0.0404	0.0813	0.0857	0.0957
GEH (% of links < 5) before	55 %	55 %	55 %	47 %	47 %	47 %
GEH after	100 %	100 %	100 %	100 %	93 %	100 %
Time for parameter tuning	1 h	1 h	20 h	1 h	1 h	20 h

SPSA requires less time to run but more time for parameter tuning (20 h), which is not the case with the GASA and MA-SW-Chains algorithms. Time for parameter tuning is the time required for determining appropriate parameter values for the optimization algorithm (setting up the optimization algorithm). While the results showed in Table 1 are very similar among the algorithms, the potential of the MA-SW-Chains algorithm is shown in Fig. 6 using the speed of convergence in contrast to GASA and SPSA. In the McTrans model, the MA-SW-Chains algorithm reaches RMSNE values below 0.02 faster, approximately at four minutes while GASA requires eight minutes. SPSA did not reach this value and its convergence curve varies over time. On the other hand, in the Reno model, the MA-SW-Chains algorithm converges faster to values close to 0.09, at approximately 10 min, GASA at 14 min, and SPSA does not reach these values.

The proposed algorithm obtained best solution in the objective function (RMSNE) and met the calibration criteria for 100 % of the links in the two models, indicating that the memetic algorithm based on Solis and Wet local search chains obtains better and faster solutions.

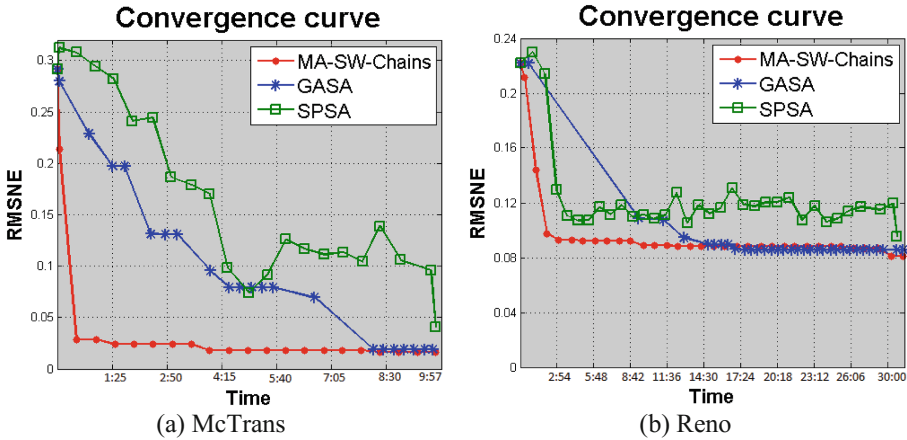


Fig. 6. Convergence curve of the objective function for the three algorithms in the two CORSIM models.

5 Conclusions and Future Work

This paper proposes the adaptation of the MA-SW-Chains algorithm, a memetic algorithm with Solis and Wet local search chains, to support the calibration of microscopic traffic flow simulation models. The use of this algorithm with the methodology proposed in [13] makes possible the calibration for any type of implementation because no information about the characteristics of traffic flow model is required. The MA-SW-Chains algorithm successfully calibrated two CORSIM vehicular traffic models, MacTrans and Reno, with the GEH statistical and RMSNE measure within the required limits.

Comparison of the MA-SW-Chains algorithm with the GASA and SPSA algorithms, previously discussed in the literature [11, 13] showed, indicated that MA-SW-Chains is an alternative that provides better results, measured by the objective function and statistical GEH.

As future work, the research group intends to evaluate other single-objective optimization approaches, such as the use of multiple offspring sampling (MOS) [21] and differential evolution using cooperative coevolution (DECC-G) [22]. In addition, queue lengths are to be included as an additional factor in the objective function.

Acknowledgements. The work in this research study was supported by the University of Cauca (Popayan, Colombia) and the University of Nevada Las Vegas, United States. We are grateful to Mr. Colin McLachlan for his help translating the first version of this document.

References

1. Hale, D.K., et al.: Optimization-based assisted calibration of traffic simulation models. *Transp. Res. Part C Emerg. Technol.* **55**, 100–115 (2015)
2. Hollander, Y., Liu, R.: The principles of calibrating traffic microsimulation models. *Transportation* **35**(3), 347–362 (2008)

3. Abdalhaq, B.K., Baker, M.I.A.: Using meta heuristic algorithms to improve traffic simulation. *J. Algorithms* **2**(4), 110–128 (2014)
4. Kim, K.-O., Rilett, L.: Simplex-based calibration of traffic microsimulation models with intelligent transportation systems data. *Transp. Res. Rec. J. Transp. Res. Board* **1855**, 80–89 (2003)
5. Park, B., Kamarajugadda, A.: Development and evaluation of a stochastic traffic signal optimization method. *Int. J. Sustain. Transp.* **1**(3), 193–207 (2007)
6. “Brian” Park, B., Yun, I., Ahn, K.: Stochastic optimization for sustainable traffic signal control. *Int. J. Sustain. Transp.* **3**(4), 263–284 (2009)
7. Chiappone, S., et al.: Traffic simulation models calibration using speed–density relationship: an automated procedure based on genetic algorithm. *Expert Syst. Appl.* **44**, 147–155 (2016)
8. Korcek, P., Sekanina, L., Fucik, O.: Calibration of traffic simulation models using vehicle travel times. In: Sirakoulis, Georgios Ch., Bandini, S. (eds.) *ACRI 2012. LNCS*, vol. 7495, pp. 807–816. Springer, Heidelberg (2012)
9. Lee, J.-B., Ozbay, K.: Calibration of a macroscopic traffic simulation model using enhanced simultaneous perturbation stochastic approximation methodology. In: *Transportation Research Board 87th Annual Meeting* (2008)
10. Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H., Wen, Y.: Calibration of microscopic traffic simulation models: methods and application. *Transp. Res. Rec. J. Transp. Res. Board* **1999**, 198–207 (2007). doi:[10.3141/1999-21](https://doi.org/10.3141/1999-21)
11. Paz, A., Molano, V., Gaviria, C.: Calibration of corsim models considering all model parameters simultaneously. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE (2012)
12. Yuan, J., Ng, S.H., Tsui, K.L.: Calibration of stochastic computer models using stochastic approximation methods. *IEEE Trans. Autom. Sci. Eng.* **10**(1), 171–186 (2013)
13. Paz, A., et al.: Calibration of traffic flow models using a memetic algorithm. *Transp. Res. Part C Emerg. Technol.* **55**, 432–443 (2015)
14. Molina, D., Lozano, M., Herrera, F.: MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization. In: *2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE (2010)
15. Li, X., et al.: 2015 IEEE Congress on Evolutionary Computation Competition on: Large Scale Global Optimization, p. 19. RMIT University (2015)
16. Alabert, A., et al.: No-Free-Lunch theorems in the continuum. *Theor. Comput. Sci.* **600**, 98–106 (2015)
17. Solis, F.J., Wets, R.J.-B.: Minimization by random search techniques. *Math. Oper. Res.* **6**(1), 19–30 (1981)
18. McTrans, CORSIM User’s Guide (2011)
19. Altıparmak, F., et al.: A steady-state genetic algorithm for multi-product supply chain network design. *Comput. Ind. Eng.* **56**(2), 521–537 (2009)
20. Molina, D., et al.: Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. *Soft. Comput.* **15**(11), 2201–2220 (2011)
21. LaTorre, A., et al: Multiple offspring sampling in large scale global optimization. In: *2012 IEEE Congress on Evolutionary Computation* (2012)
22. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. *Inf. Sci.* **178**(15), 2985–2999 (2008)