IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 6, 2020, accepted May 25, 2020, date of publication June 1, 2020, date of current version June 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999081

A Multi-Objective Approach for the Calibration of Microscopic Traffic Flow Simulation Models

CARLOS COBOS[1], (Senior Member, IEEE), ALEXANDER PAZ[2], (Member, IEEE), JULIO LUNA[1], CRISTIAN ERAZO[1], AND MARTHA MENDOZA[1]

[1]Information Technology Research Group (GTI), Universidad del Cauca, Popayán 190001, Colombia
[2]School of Civil and Environmental Engineering, Queensland University of Technology, Brisbane, QLD 4000, Australia

Corresponding author: Alexander Paz (alexander.paz@qut.edu.au)

This work was supported in part by the University of Cauca and the Queensland University of Technology.

ABSTRACT The calibration of traffic-flow simulation models continues to be a significant problem without a generalized, comprehensive, and low-cost solution. Existing calibration approaches either have not explicitly addressed the multi-objective characteristics of the problem or determining their hyperparameters requires significant effort. In addition, statistical evaluation of alternative solution algorithms is not performed to ensure dominance and stability. This study proposes an adaptation and advanced implementation of the Multi-Objective Global-Best Harmony Search (MOGBHS) algorithm for calibrating microscopic traffic-flow simulation models. The adapted MOGBHS provides five key capabilities for solving the proposed problem including 1) consideration of multiple objectives, 2) easily extendable to memetic versions, 3) simultaneous consideration of continuous and discrete variables, 4) efficient ordering of no dominated solutions, 5) relatively easy tuning of hyperparameters, and 6) easily parallelization to maximize exploration and exploitation without increasing computing time. Three traffic flow models of different dimensionality and complexity were used to test the performance of seventeen metaheuristics for solving the calibration problem. The efficiency and effectiveness of the algorithms were tested based on convergence, minimization of errors, calibration criterion, and two statistical nonparametric tests. The proposed approach dominated all alternative algorithms in all cases and provided the most stable and diverse solutions.

INDEX TERMS Multi-objective optimization, calibration, traffic flow simulation, MOGBHS, NSGA-II.

I. INTRODUCTION

One of the most challenging tasks for the development and use of microscopic traffic flow simulation models is the calibration of their parameters [1], [2]. Considering the highly dimensional search space, the randomness present in the real-world systems been represented, and the intricate interactions among parameters, metaheuristics are often used for calibration involving: 1) criterion for the evaluation of one or multiple objective functions, 2) a vector of supply and demand parameters that must be calibrated, and 3) an algorithm that finds a solution to minimize, or maximize, the objective function(s) [4], [5]. Recognizing the challenges associated with the calibration of models representing real-word systems, Hale *et al.* [3] proposed an architecture for software assisted calibration to enable flexibility, practicality, and ease-of-use. The proposed architecture is general, but it was tested using two simple traffic models and two search approaches.

Recently, memetic algorithms [6] combining exploration and exploitation while considering the characteristics of the specific problem have showed significant advantages over their counterparts. Paz *et al.* [2] combined a Genetic Algorithm (GA) and Simulating Annealing (SA) to obtain GASA, a memetic approach that provided superior results relative to Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall [7]. Cobos *et al.* [8] proposed a mono-objective algorithm based on Solis and Wets Local Search Chaining (MA-SW-Chains) [10] for the calibration of highly dimensional traffic flow models. The proposed approach provided better results in terms of convergence compared to GASA and SPSA. However, the MA-SW-Chains approach required significantly more running time. Amirjamshidi and Roorda [9] proposed a pseudo multi-objective approach for the calibration of speeds, counts, and standard deviation of acceleration. They combined all objectives into a single function

The associate editor coordinating the review of this manuscript and approving it for publication was Ioannis Schizas.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

using weights. Hence, there is loss of information and no guaranty of minimizing (optimizing) all objectives simultaneously. Ciuffo and Punzo [5] proposed and evaluated an approach for verifying the robustness of a calibration algorithm. Synthetic data was used to evaluate five mono-objective optimization algorithms which are popular in the literature and practice for the calibration of traffic flow simulation models.

The algorithms listed above adopted a naive strategy using weights to combine multiple objectives within a single function. Most real-world problems require or will benefit from a multi-objective strategy where a solution includes a set, Pareto Front, rather than a single solution [11]. The Pareto Front only includes solutions that cannot be improved without making worse off at least one objective. Having available and visualizable multiple optimal solutions helps to get a better understanding of the trade-offs to select an adequate alternative. Preferences and considerations beyond those covered by the optimization framework such as political and/or social aspects can be used to create criteria to break ties [12].

Several recent studies have recommended multi-objective optimization for future research regarding the calibration of simulation-based traffic flow models [13], [14]. For example, Sparnaaij *et al.* [14] proposed a multi-objective approach for the calibration of pedestrian simulation models. Their results illustrated the advantages of multi- relative to mono-objective optimization. However, the study was limited to pedestrian flows and the use of a grid search which is very inefficient to solve large-scale optimization problems such as the one involving the calibration of microscopic traffic flow simulation models. Similarly, Cascan *et al.* [15] used a GA for the calibration of traffic flow models considering both flow and safety characteristics. However, their study was limited to a small number of calibration parameters and a single metaheuristic, GA.

According to several authors [11], [16], [17], the most popular multi-objective optimization algorithms are NSGA-II [18] and SPEA-2 [19], which are relatively easy to implement [20], [21]. Cobos *et al.* [22] proposed a multi-objective memetic algorithm based on NSGA-II and SA, NSGA-II-SA, for the calibration of microscopic traffic flow simulation models. NSGA-II provided exploration while SA performed exploitation. Two relatively small microscopic traffic flow simulation models were calibrated and compared using NSGA-II-SA, GASA and SPSA. The results illustrated the advantages of NSGA-II-SA over GASA and SPSA in terms of convergence and running time for small problems. However, other multi-objective metaheuristics were not considered, and the hyperparameters of the algorithms were not properly tuned. Karimi *et al.* [23] achieved great results using a multi-objective stochastic optimization algorithm for the calibration of a microsimulation traffic flow model. However, they compared the performance of their proposed algorithm with the results obtained using a mono-objective approach. Similarly, their experiments only included one traffic flow model and the objective functions did not consider traffic speed which is a key performance measure.

According to the No Free Lunch Theorems for Optimization [5], [24], the only way to determine the best algorithm to solve an optimization problem is through evaluation and comparison. Alternative optimization algorithms to be evaluated are chosen based on their characteristics and those of the problem context. Previous studies comparing various multi-objective algorithms for the calibration of various flow models have not shown yet a dominant algorithm [25], [26]. Different algorithms present various levels of complexity and implementation challenges including selection and tuning of their hyperparameters. Del Ser *et al.* [27] provided a detailed review of various relevant advanced optimization algorithms.

The current literature does not provide a systematic and comprehensive analysis of available and relevant optimization algorithms for the calibration of highly dimensional traffic flow simulation models. A systematic and comprehensive analysis requires explicit consideration of multiple goodness of fit, multiple traffic flow systems of various degrees of complexity, and a statistical analysis of performance by the optimization algorithms. Similarly, details for effective implementations are lacking in the literature. Following these limitations and the No Free Lunch Theorems for Optimization [5], [24] this study adapts, implements, and evaluates various mono- and multi-objective metaheuristics for the calibration of large microscopic traffic flow simulation models. The primary objective is to determine a calibration strategy, including an algorithm and implementation details, that is likely to be effective while explicitly considering multiple objectives. Considering the literature and the characteristics of the calibration problem, the performance of the proposed approach, Multi-objective Global-Best Harmony Search (MOGBHS), is compared with various competing alternative algorithms to illustrate its potential and advantages. In addition, three memetic versions of MOGBHS are implemented and evaluated. The proposed implementations include: 1) parallel threading for the simultaneous generation of multiple solutions to maximize exploration and exploitation without increasing computing time, 2) parallel evaluation of the objective functions to save resources, and 3) three strategies of exploitation, for the memetic versions of the algorithms, including Hill Climbing (HC), SA, and Iterate Local Search (ILS), which are applied to a percentage of the Pareto Front.

GBHS combines Harmony Search (HS) and Particle Swarm Optimization (PSO) to provide better results than HS [28] for solving problems with continuous and discrete variables [27]–[29]. It requires few hyperparameters which facilitates its implementation. Previously, HS has been successfully applied to many real-world optimization problems with better results relative to other evolutionary algorithms in terms of solution quality [29]–[31].

MOGBHS is an extension of the Global-Best Harmony Search (GBHS) which provides great results while considering multiple objectives simultaneously. For example, it has provided great results for the generation of schedules for an integrated transit system [32]. MOGBHS showed superior performance relative to NSGA-II. MOGBHS also provided superior results for the simultaneous generation of optimal routes and schedules [33] when compared to NSGA-II and MOEA/D. MOGBHS was also compared to NSGA-II, MOEA/D, MSOPS and SPEA2 using 21 multi-objective test problems (12 with restrictions and 9 without restrictions) taken from the multi-objective competition during the 2009 IEEE-CEC conference [34]. In addition, MOGBHS provided superior Inverted Generational Distance with 95% significance using 10,000 and 20,000 evaluations of the objective (fitness) functions for the nonparametric tests of Friedman and Wilcoxon [35].

The remaining of this paper is organized as follows. Section II provides the proposed solution approach, including details about the objective functions, the calibration criterion, the solution algorithm, and the threading implementation. Section III presents the traffic flow models used for testing, and provides results and analyses including convergence and nonparametric tests (Friedman and Wilcoxon). Finally, Section IV presents conclusions and future work.

II. THE PROPOSED SOLUTION

A. MATHEMATICAL PROGRAM

The most common traffic flow characteristics use for calibration and validation are errors including the difference between actual and simulated volume and speed [36]. Similar to previous studies [2], [8], this research used well established standards for the calibration of microscopic traffic flow simulation models [37].

The objective functions in this study are provided by (1) and (2); they respectively involve the minimization of the normalize root mean square (NRMS) difference between simulated and actual volumes and speeds. Previous studies [2], [38] have combined these objectives into a single function using weights as illustrated in (3). A single objective function enables the use of mono-objective optimization algorithms as well as to break ties along the Pareto Front. However, this is a naive or simplistic approach which does not take advantage of multi-objective optimization. The fundamental traffic flow relationships show that the same flow (volume) can be associated with two distinct speeds [39].

$$\boldsymbol{Min.}\ (\boldsymbol{vol.\ error})$$
$$= \boldsymbol{Min.} \left(\frac{1}{\sqrt{N}} \sum_{t=1}^{T} \sqrt{\sum_{i=1}^{N} \left(\frac{V_{i,t} - \tilde{V}_{(\theta)i,t}}{V_{i,t}} \right)^{2}} \right) \qquad (1)$$

where $V_{i,t}$ and $\tilde{V}_{(\theta)i,t}$ are respectively the actual and simulated volume counts for link $\boldsymbol{i}$ at time $\boldsymbol{t}$, $\boldsymbol{\theta}$ is the vector of supply and demand parameters that need to be calibrated, $\boldsymbol{N}$ is the total number of links with data, and $\boldsymbol{T}$ is number of considered time periods $\boldsymbol{t}$.

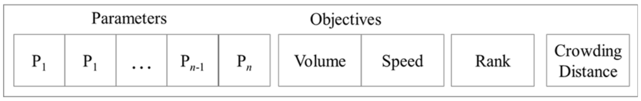


FIGURE 1. Representation of a harmony.

$$\boldsymbol{Min.}\ (\boldsymbol{speed\ error})$$
$$= \boldsymbol{Min.} \left(\frac{1}{\sqrt{N}} \sum_{t=1}^{T} \sqrt{\sum_{i=1}^{N} \left(\frac{S_{i,t} - \tilde{S}_{(\theta)i,t}}{S_{i,t}} \right)^{2}} \right) \qquad (2)$$

where $S_{i,t}$ and $\tilde{S}_{(\theta)i,t}$ are respectively the actual and simulated speeds measured on $\boldsymbol{i}$ at $\boldsymbol{t}$, and $\boldsymbol{\theta}$, $\boldsymbol{N}$ and $\boldsymbol{T}$ are as defined before.

$$\boldsymbol{Min.}\ (\boldsymbol{NRMS})$$
$$= \boldsymbol{Min.}\ (\boldsymbol{W} * \boldsymbol{vol.\ eror} + (\boldsymbol{1} - \boldsymbol{W}) * \boldsymbol{speed\ error}) \qquad (3)$$

where $\boldsymbol{W}$ is a weight use to assign relative value to errors for volumes and speeds. The vector of calibration parameters in the objective functions (1) and (2) is subject to lower and upper bounds as denoted by the following inequalities (4). These bounds are required to ensure that the parameters are within values that are consistent with the real-world.

$$\boldsymbol{Lower\ bound} \leq \boldsymbol{\theta} \leq \boldsymbol{Upper\ bound} \qquad (4)$$

B. SOLUTION ALGORITHM

An adaptation of the MOGBHS algorithm is proposed to solve the above calibration problem because: 1) multiple objectives need to be considered, 2) MOGBHS is easy to extend to achieve a memetic version of the algorithm, 3) MOGBHS enables simultaneous consideration of continuous and discrete variables, and 4) MOGBHS enables fast ordering of no dominated solutions and Crowding distance to create an effective Pareto Front [32], [33]. For each iteration, the standard MOGBHS creates a single harmony or offspring combining several solutions and/or random characteristics. Finally, the Pareto Front in the harmony memory provides a set of optimal solutions.

1) SOLUTION REPRESENTATION

A solution, or harmony, to the optimization problem generated by MOGBHS includes a vector of calibration parameters, the corresponding values of the objective functions, the rank (Pareto Front number), and the Crowding Distance in its Pareto Front rank. A representation of a harmony is provided in Fig. 1.

A solution is better than other based on its dominance. That is, a solution is better than (dominates) other if at least one objective holds a better value while the other objectives are equal. Fig. 2 illustrates the evaluation of dominance between solutions X and S given the objectives of minimizing errors for volume (Volume) and speed (Speed). The method, Dominate, in Fig. 2 compares the current harmony S with a

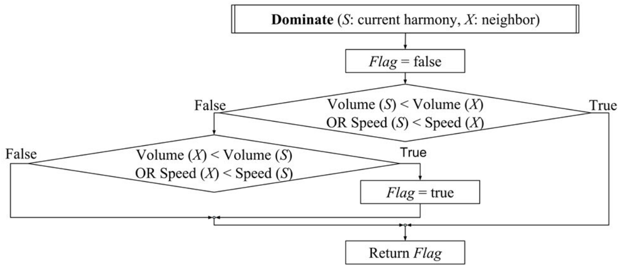

Dominate (S: current harmony, X: neighbor)
Flag = false
Volume (S) < Volume (X) OR Speed (S) < Speed (X)
False
True
Volume (X) < Volume (S) OR Speed (X) < Speed (S)
False
True
Flag = true
Return Flag

FIGURE 2. Dominance between X and S (Minimization of Volume and Speed).

potential new solution X and assigns "true" if X dominates S, otherwise it assigns "false".

2) ALGORITHMIC STEPS

Fig. 3 provides a flow chart for MOGBHS [32] applied to the proposed calibration problem. Additional details about MOGBHS including a pseudo-code were provided by Ruano-Daza *et al.* [33]. Initially, the flow chart lists the required inputs for GBHS including: (1) harmony memory size (HMS), (2) harmony memory consideration rate (HMCR), (3) minimum pitch adjustment rate (PAR_{min}), (4) maximum pitch adjustment rate (PAR_{max}), (5) maximum number of improvisations (NI), (6) number of solutions generated in parallel using threads (T), (7) dimension of the solution vector (n), (8) the local search algorithm (local), (9) probability of applying local search (r), and (10) probability of selecting a harmony (p) for local optimization. If T is set to one, the algorithm generates only one solution within the loop of improvisations. The number of calibration parameters n in each harmony is illustrated in Fig. 1. The local search could use HC, SA, or ILS to exploit the Pareto Front. If r is set to zero (0), no local search is performed, and the algorithm becomes a standard MOGBHS.

The first step of the proposed algorithm is to initialize the harmony memory generating HMS random solutions using T threads with parallel execution of simulation models and evaluation of objectives. These solutions are generated without violating constraints (4). The second step is to order the solutions using their dominance. This produces a set of Pareto Front Ranks with rank equal to zero for those solutions that are not dominated and higher values otherwise. Next, the algorithm calculates the Crowding distance within each rank [18], [33]. When the algorithm is ordering the solutions, if two solutions have the same rank, the one with the largest Crowding distance is ordered first (is better) because it is far to the others with the same rank.

Step three starts an iterative process to repeat steps four to eleven NI times. Each iteration includes the calculation of the pitch adjustment rate (PAR) used for creating the vector of parameters for a new harmony. PAR is calculated in step four using (5)

$$PAR = PAR_{min} + \frac{(PAR_{max} - PAR_{min}) * i}{NI} \tag{5}$$

where $i \in \{1, 2, \ldots, NI\}$ is the number of current iteration or improvisation.

Step five starts a loop to generate T solutions (harmonies) in parallel. In step six, the rules of GBHS are used to generate a new harmony using hyperparameters HMCR and PAR. HMCR is used to determine whether the d element of the new harmony comes from the harmony memory or is generated randomly (uniformly distributed between the lower and upper bound of the current parameter vector d). If it comes from memory, PAR decides about getting this element from either the best or a random solution in the harmony memory from the same place of the current parameter vector d.

In step seven, the fitness values (objectives) are evaluated. This requires running the microscopic traffic flow simulation model using the corresponding parameters. The evaluation of the traffic flow simulation models is the most intensive part of the solution algorithm. It is also the most memory intensive task because several models with their appropriate parameters are required at the same time, one for each thread. Hence, to maximize performance the analyst must carefully setup $\boldsymbol{T}$ according to the available resources and size of the traffic flow models. To avoid unnecessary re-evaluation of the same solutions in case that they are generated again in future interactions, they are stored in a hash structure along with the corresponding values of the objective functions. The new harmony is added to the harmony memory in step eight.

In step 9, the harmony memory is ordered as explained before. In step 10, the worse T solutions with the highest Ranks and the smallest Crowding distances at each rank are deleted from harmony memory to reduce its size and select only the best HMS solutions.

Step 11 involves deciding about performing local search and executing the corresponding action. If a uniformly random number generated between zero and one is less than r, local search is performed for a portion of solutions in the Pareto Front; otherwise, no local search is performed. This local search provides memetic capabilities to the proposed algorithm. Hyper parameter r should not be too close to one to avoid over exploitation which could affect the exploration. Only a percentage p of the solutions in the Pareto Front are affected by local search as can be seen in Fig. 4 where $|PF|$ means the total number of harmonies in the Front. Using the local search algorithm selected by the analyst, only a percentage MP (mutation probability hyper parameter) of elements in the vector of parameters of each solution are mutated to avoid leaving the local region instead of performing exploration. The Local Search method executes a maximum of T threads in parallel. If the number of harmonies selected from the Pareto Front to do local optimization is greater than T, the solutions are queued and processed as the threads are released. When all threads executing local search complete the LocalSearch method, the algorithm continues with the next iteration of MOGBHS.

When the algorithm completes the loop started in step 3 (Fig. 3), it returns harmonies (solutions) in harmony memory sorted by their Rank values (first the lowest) and Crowding

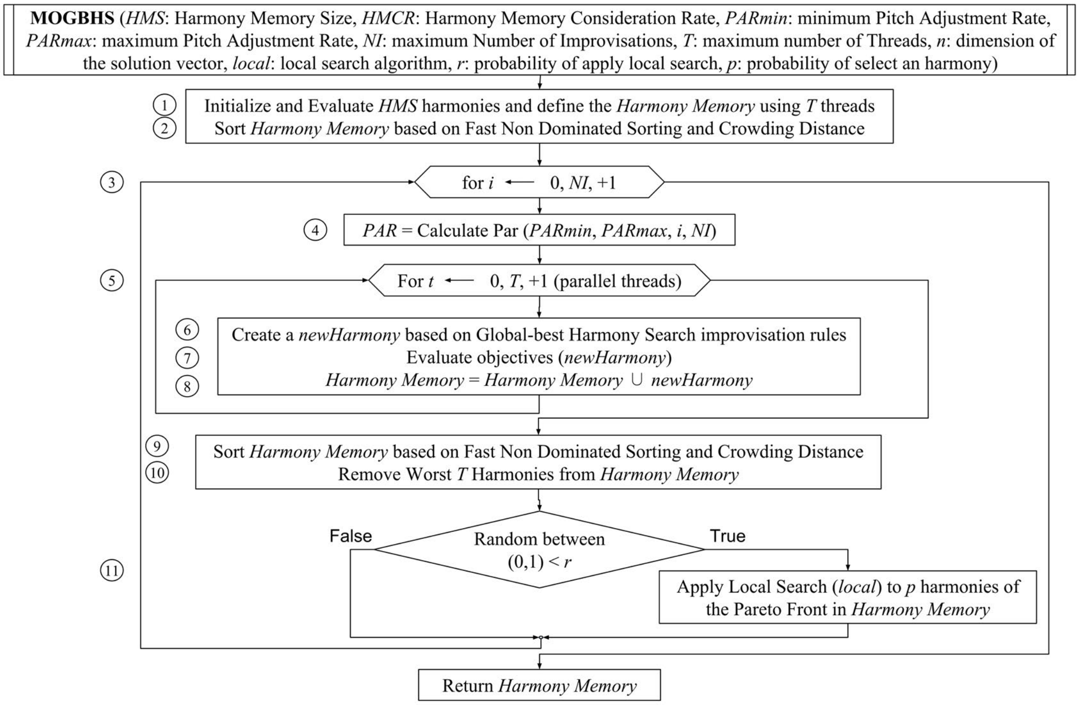

MOGBHS (HMS: Harmony Memory Size, HMCR: Harmony Memory Consideration Rate, PARmin: minimum Pitch Adjustment Rate, PARmax: maximum Pitch Adjustment Rate, NI: maximum Number of Improvisations, T: maximum number of Threads, n: dimension of the solution vector, local: local search algorithm, r: probability of apply local search, p: probability of select an harmony)
1
2
Initialize and Evaluate HMS harmonies and define the Harmony Memory using T threads
Sort Harmony Memory based on Fast Non Dominated Sorting and Crowding Distance
3
for i ← 0, NI, +1
4
PAR = Calculate Par (PARmin, PARmax, i, NI)
5
For t ← 0, T, +1 (parallel threads)
6
7
8
Create a newHarmony based on Global-best Harmony Search improvisation rules
Evaluate objectives (newHarmony)
Harmony Memory = Harmony Memory ∪ newHarmony
9
10
Sort Harmony Memory based on Fast Non Dominated Sorting and Crowding Distance
Remove Worst T Harmonies from Harmony Memory
11
False
Random between (0,1) < r
True
Apply Local Search (local) to p harmonies of the Pareto Front in Harmony Memory
Return Harmony Memory

FIGURE 3. Memetic MOGBHS algorithm using T parallel solutions.

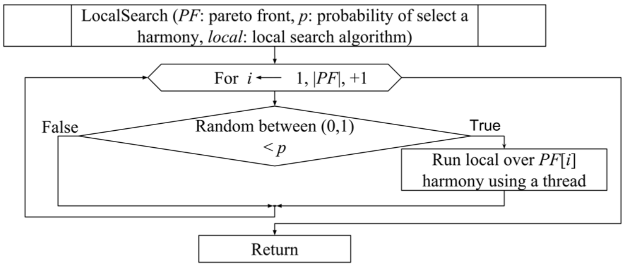

LocalSearch (PF: pareto front, p: probability of select a harmony, local: local search algorithm)
For i ← 1, |PF|, +1
False
Random between (0,1) < p
True
Run local over PF[i] harmony using a thread
Return

FIGURE 4. Local search.

distance (first the highest within each Rank). The Pareto Front (Rank = 0) easily can be showed to the analyst in a 2D dispersion graph using the x axis as the error for volume and y axis as the error for speed. Examples of this kind of graphs are shown latter as part of the experiments and results.

III. EXPERIMENTS AND RESULTS

Experiments using models from three different vehicular networks were completed to evaluate the performance of the proposed algorithm and our implementation. These models were coded using CORSIM [37]. No specific information about the characteristics of the simulation environment was used for the development of the proposed approach. Hence, the proposed approach is likely to provide similar results using alternative traffic simulation software. Similarly, the same number of evaluations of the of the objective function was used in all experiments to enable a fair comparison among algorithms.

A. CALIBRATION CRITERION

A common calibration criterion from the literature was used in this study [2], [8], [22], [38]: the absolute difference between actual and simulated link counts and speeds should be less than 15% for at least 85% of the links. In addition, the GEH [40] is required to be less than 5 for at least 85% of the links. The GEH is calculated using (6).

$$GEH = \sqrt{\frac{2(V_i - \tilde{V}_{(\theta)i})^2}{V_i + \tilde{V}_{(\theta)i}}} \tag{6}$$

where V_i and $\tilde{V}_{(\theta)i}$ are respectively the actual and simulated volume counts for link i, and θ is defined as before.

Microscopic traffic flow simulation models are not that sensitive to small changes in their parameters. However, significant changes to the model can preclude exploitation and create a random search. In our experiments, at most, 50% of calibration parameters can be adjusted (mutated) in each iteration of the proposed optimization algorithm. This is accomplished using the *MP* hyper parameter in method LocalSearch, described above. Details about the parameters that can be calibrated are provided in the user's guide of the simulation software [37].

B. TEST MODELS

Given that the performance of any metaheuristic is likely to be highly dependent of the characteristics of the specific problem under study, three simulation models with different levels of dimensionality and complexity were used to evaluate the

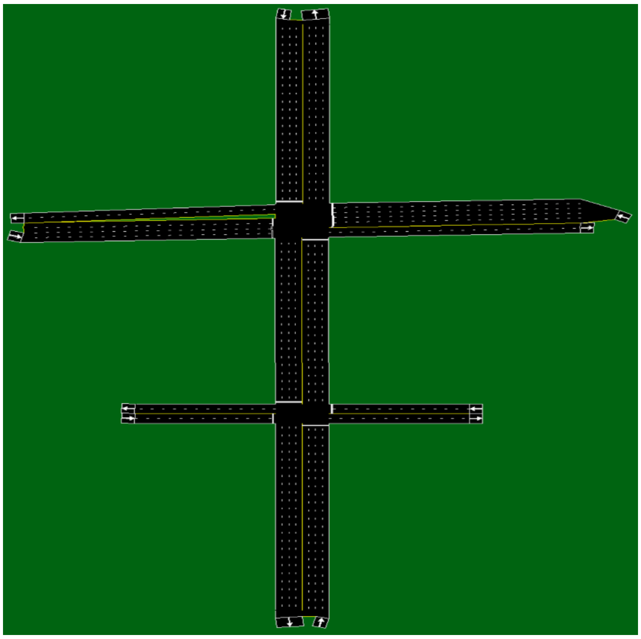


FIGURE 5. McTrans network.

proposed solution approach. These three models have very different characteristics as described below. Hence, the proposed evaluation approach is likely to be fair and sufficient to demonstrate the potential of the MOGBHS algorithm and our implementation framework for the calibration of microscopic traffic flow simulation models.

1) LOW DIMENSIONALITY MODEL: McTRANS NETWORK

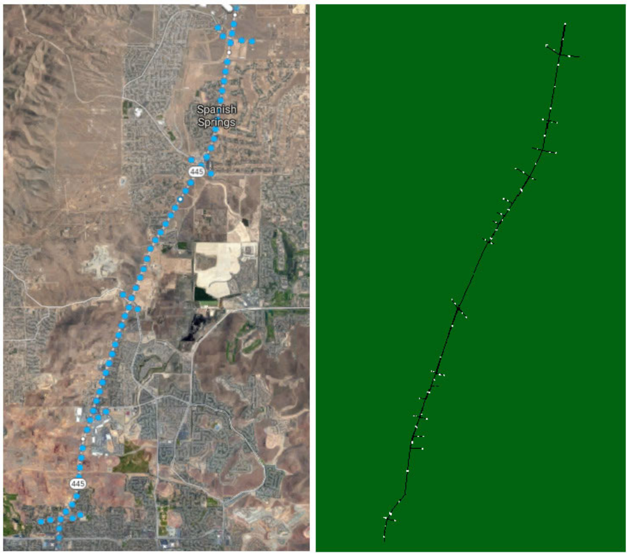
The low dimensionality model is a hypothetical network with all parameters set to default values. This model was developed by McTrans Center which is a provider of transportation analysis tools including the microscopic traffic flow simulator CORSIM [37]. The model includes 20 arterial links and two interceptions. Fig. 5 illustrates this model. The baseline corresponds to the scenario obtained with all parameters set to default values which are unknown by the proposed optimization framework. After the baseline is calculated, all parameters are perturbed randomly to represent an uncalibrated model. The initial NRMS was 0,291 with *W* 0.7.

2) MEDIUM DIMENSIONALITY MODEL: RENO NETWORK

The medium dimensionality model corresponds to the Pyramid network in Reno, Nevada. This network includes 126 arterial links. Actual volumes and speeds were obtained for 45 links using traffic sensors. The initial NRMS was 0,222 with W 0.7. Fig. 6 illustrates the Pyramid network. Field data for this model was provided by the Nevada Department of Transportation (NDOT).

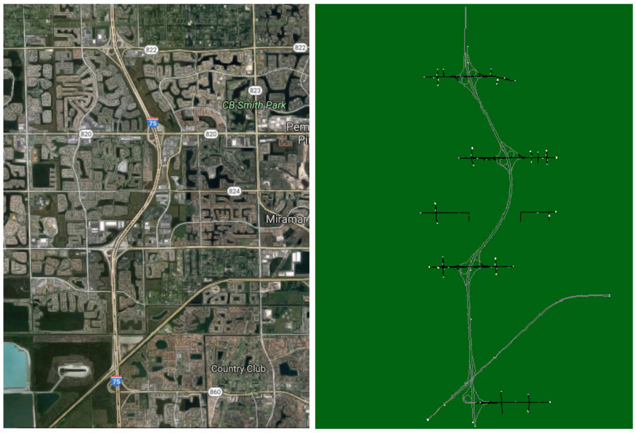
3) HIGH DIMENSIONALITY MODEL: I-75 NETWORK

The high dimensionality model corresponds to a portion of the interstate 75 (I-75) in Miami, Florida. This network includes 703 links. Sensor data was obtained for 346 locations. The initial NRMS was 0,332 with W 0.7. Fig. 7 illustrates the I-75 network used in this study.



(a) (b)

FIGURE 6. Google (a) and TRAFVU (b) view of the Pyramid network in Reno, Nevada.



(a) (b)

FIGURE 7. Google (a) and TRAFVU (b) view of the I-75 freeway, Miami, Florida.

C. ALTERNATIVE ALGORITHMS

The performance of the proposed solution algorithm is compared with the following alternatives which were used recently and successfully to solve the same calibration problem [2], [8], [22], [38]:

- Mono-objective algorithms,
- Single-state multi-objective algorithms, also used as local search strategies in memetic algorithms, and
- Memetic multi-objective algorithms.

All algorithms were implemented to take advantage of parallel computing using multi-threading. This enabled a direct comparison with the proposed solution algorithm as described above.

1) MONO-OBJECTIVE ALGORITHMS

SPSA and GASA were the two mono-objective algorithms used for comparison purposes. SPSA was proposed by [2] for the calibration of microscopic traffic flow simulation models. However, only models of low and medium dimensionality were used to evaluate the performance of the algorithm. In addition, extensive sensitivity analyses were required to fine-tune the hyperparameters used by SPSA. GASA is a memetic algorithm that uses SA in each iteration to obtain, in parallel, *n* local solutions.

2) SINGLE-STATE MULTI-OBJECTIVE ALGORITHMS

Three single-state multi-objective algorithms including HC, SA and ILS were also used to calibrate the three test traffic flow models. HC and SA generate neighbor solutions using a perturbation on a portion of the vector of traffic flow model parameters. Once an element of this vector is selected for update, there is a 50% (MP hyper parameter) chance of a positive or negative perturbation. ILS builds a set of optimal solutions by perturbing the current minimum and applying local search. The perturbation must be sufficient to lead the search to a different basin to reach an alternative local optimum. The local search in this case was performed using HC.

The optimization hyperparameters for SA were obtained from [2] and determined using sensitivity analyses. The optimization hyperparameters for ILS and HC were obtained using covering arrays. Section D provides details about the tuning process used to determine hyperparameters for the proposed algorithm. A similar process was executed for ILS and HC. The following were the corresponding hyperparameters:

- SA: temperature = 15, cooling rate = 0.95, percentage of perturbation = 30%, and acceptance constant = 0.35.
- HC: radius = 0.9 and percentage of perturbation = 10%.
- ILS: percentage of perturbation = 10%, probability of acceptance = 0.4, and acceptance constant = 0.35.

These algorithms were used as local search strategy in the memetic versions of MOGBHS and the other two multi-objective algorithms, NSGA-II and SPEA-2.

3) MEMETIC MULTI-OBJECTIVE ALGORITHMS

Similar to a GA, NSGA-II generates multiple solutions based on characteristics of an existing set of alternatives (population of solutions). For a population of size *N*, N/2 pairs of solutions are combined to generate N new offspring. Mutation are applied to these offspring to generate new solutions with different characteristics. All solutions are ordered in a Pareto Front using dominance and Crowding distance. In each iteration of the search process, the worse N solutions are eliminated. The remaining N solutions are considered as the new population. The above process is repeated until convergence or the maximum number of iterations is reached. The methods used here for fast ordering of no dominated solutions were the ones recommended by [18].The hyperparameters for NSGA-II were obtained using covering arrays [41] including: population size = 60; number of generations (iterations) = 180; percentage of mutation equal to 10% for low dimensionality model and 5% for the medium and high dimensionality models.

The SPEA-2 algorithm used in this research is based on the proposal by [42], [43]. SPEA-2 generates in each iteration a population of possible solutions. New solutions are generated in each iteration based on the best current solutions and the method of tournament. Mutation and reproduction methods were as in NSGA-II. As was the case for NSGA-II, the hyperparameters for SPEA-2 were obtained using covering arrays [41], including: population size = 60; size for the external population = 15; number of generations = 180; percentage of mutation equal to 10% for low dimensionality model and 5% for the medium and high dimensionality models.

Local search capabilities were included in NSGA-II and SPEA-2 using HC, SA and ILC with probability *r* (20%) to be applied at the end of each iteration and affecting only a percentage *p* (50%) of the solutions in the Pareto Front.

TABLE 1. Vocabulary used with covering arrays.

HMS	HMCR	PAR_{min}	PAR_{max}	NI (Improvisations)
40	0.95	0.1	0.9	450
10	0.97	0.05	0.95	550
20	0.9	0	1	500
60	0.85	0.15	0.85	400
80	0.8	0.2	0.8	360

D. HYPERPARAMETERS FOR THE PROPOSED ALGORITHM

Covering arrays were used to determine the best hyperparameters for the proposed implementation of the MOGBHS algorithm. Previous studies have showed the effectiveness of using this approach to determine this type of hyperparameters [41]. We used a covering array with strength 2 and vocabulary size of 5. Values for the vocabulary were obtained through sensitivity analyses using recommendations from the literature [28], [33]. Table 1 lists the vocabulary used during the hyperparameters tuning process. The first row provides the values recommended in the literature for GBHS. The following rows provide variations that showed adequate results during preliminary testing. There are other strategies for hyper parameter tuning including the Meta Evolutionary Algorithm (Meta-EA) [44]. However, covering arrays has showed to be a feasible alternative because they reduce the effort (execution time) and increase the probability to find adequate combinations of hyperparameters; that is, maximum coverage with the minimum effort.

Table 2 provides the resulting covering array. The number of rows (33) denotes the required number of cases. Each case was executed 30 times for the low dimensionality traffic

TABLE 2. Covering Arrays for tuning hyperparameters.

	HMS	HMCR	PAR_{min}	PAR_{max}	NI (Improvisations)
1	20	0.97	0.1	0.9	400
2	40	0.9	0.1	0.95	360
3	10	0.8	0.1	1	550
4	60	0.95	0.1	0.85	360
5	60	0.85	0.1	0.8	450
6	80	0.85	0.05	0.9	360
7	10	0.97	0.05	0.95	400
8	20	0.9	0.05	1	500
9	40	0.97	0.05	0.85	450
10	60	0.8	0.05	0.8	400
11	10	0.95	0	0.9	450
12	20	0.95	0	0.95	550
13	60	0.97	0	1	360
14	80	0.9	0	0.85	400
15	40	0.85	0	0.8	360
16	40	0.8	0.15	0.9	400
17	60	0.85	0.15	0.95	500
18	80	0.95	0.15	1	450
19	20	0.85	0.15	0.85	550
20	10	0.9	0.15	0.8	360
21	60	0.9	0.2	0.9	550
22	80	0.8	0.2	0.95	450
23	40	0.95	0.2	1	400
24	10	0.85	0.2	0.85	500
25	20	0.97	0.2	0.8	360
26	80	0.97	0.1	0.8	500
27	80	0.95	0.05	0.8	550
28	40	0.97	0.15	0.9	550
29	20	0.8	0	0.85	500
30	40	0.85	0.1	1	400
31	40	0.95	0.1	0.9	500
32	20	0.9	0.1	0.9	450
33	40	0.8	0.1	0.9	360

flow model. The hyperparameters obtained were HMS = 60, HMCR = 0.97, PAR_{min} = 0, PAR_{max} = 1, and number of improvisations NI = 360. The probability and percentage of exploitation were 10% (r) and 50% (p) respectively.

E. HARDWARE

The experiments were performed using Windows Server 2008 R2 64 bits, Intel Xeon CPU X7560 2.26 GHz with 64 cores, and 256 Gigabytes DDR3 of Ram. The calibration software was implemented using Java Virtual Machine V1.7.60. Considering the characteristics of available hardware and the size of the high dimensionality test model, 15 solutions were obtained in parallel in all cases.

F. RESULTS AND ANALYSIS

1) LOW DIMENSIONALITY MODEL

TABLE 3. Results for the calibration of the low dimensionality model (best in bold).

Algorithm	NRMS with W 0.7				Average searching time (mm:ss)
	Best	Worse	Average over 30 repetitions	Standard deviation	
HC	0.0152	0.0684	0.0257	0.011851450	12:52
SA	0.0168	0.0611	0.0239	0.007659980	14:09
ILS	0.0150	0.0767	0.0239	0.011230950	14:00
NSGA-II	0.0137	0.0311	0.0232	0.003979053	14:09
NSGA-II-HC	0.0126	0.0340	0.0219	0.005008199	12:34
NSGA-II-SA	0.0134	0.0312	0.0223	0.003884468	12:33
NSGA-II-ILS	0.0126	0.0302	0.0217	0.004202199	14:25
MOGBHS	**0.0088**	**0.0175**	**0.0147**	0.001697645	13:24
MOGBHS-HC	0.0107	0.0207	0.0155	0.002364544	13:25
MOGBHS-SA	0.0115	0.0213	0.0174	0.002086068	13:24
MOGBHS-ILS	0.0102	0.0188	0.0158	**0.001513986**	13:28
SPEA-2	0.0163	0.0337	0.0247	0.004368220	13:14
SPEA-2-HC	0.0116	0.0324	0.0201	0.005081893	13:01
SPEA-2-SA	0.0154	0.0343	0.0225	0.003933253	13:13
SPEA-2-ILS	0.0134	0.0288	0.0187	0.003381781	13:11
GASA	0.0119	0.0354	0.0236	0.005046561	11:49
SPSA	0.0248	0.2163	0.0821	0.040124355	11:30

Table 3 shows the results obtained using the low dimensionality traffic flow model, McTrans Network. The MOGBHS provided superior results in terms of "Best NRMS", "Worse NRMS", and "Average NRMS." That is, the standard MOGBHS algorithm provided the best results without the need for additional local search. However, the proposed memetic MOGBHS-ILS provided similar results with slightly less standard deviation which denotes more stable solutions. These results were followed by those obtained using multi-objective algorithms. It was observed that the incorporation of local search into the NSGA-II and SPEA-2 improved the quality of the results and running time for NSGA-II. As expected, single-state multi-objective and mono-objective algorithms provided the worse results including larger standard deviations compared to all other algorithms.

SPSA provided the smallest running time. However, its performance was the worst. These results were interesting given that SPSA has received significant attention by the transportation engineering community. The results from this research should encourage others to consider MOGBHS for solving other types of relevant optimization problems. Despite the difference in performance illustrated in Table 3, all algorithms were able to meet the calibration criterion.

All algorithms were analyzed using the Friedman and Wilcoxon nonparametric tests [45] using their NRMS. Table 4 shows the ranks obtained under the Friedman test. The lower the ranking, the better the statistical performance for the corresponding algorithm. Consistent with the previous results, the MOGBHS outperformed all other algorithms followed by its memetic versions.

Similarly, the SPSA algorithm provided the worse performance. These results also showed that adding local search to some of the multi-objective algorithms improved their performance. However, this was not the case for

TABLE 4. Results for the Friedman test using the low dimensionality model (best in bold).

Algorithm	Ranking	Algorithm	Ranking
MOGBHS	**2.2667**	ILS	10.3
MOGBHS-ILS	3.3	NSGA-II-SA	10.3
MOGBHS-HC	3.3667	SPEA-2-SA	10.7667
MOGBHS-SA	5.5333	SA	11.1667
SPEA-2-ILS	6.7333	GASA	11.2
SPEA-2-HC	8.1	NSGA-II	11.6
NSGA-II-ILS	9.4333	SPEA-2	12.2333
NSGA-II-HC	9.6333	SPSA	16.8
HC	10.2667		

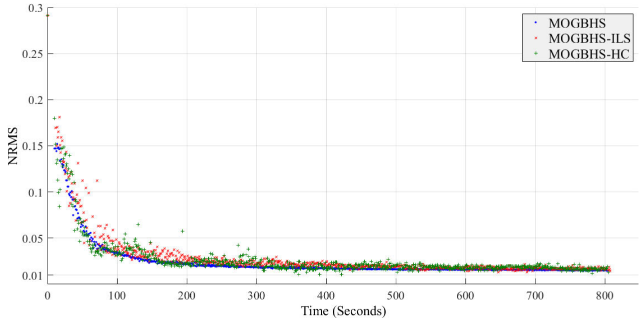


FIGURE 8. NRMS versus searching time by the top three algorithms for the calibration of the low dimensionality model.

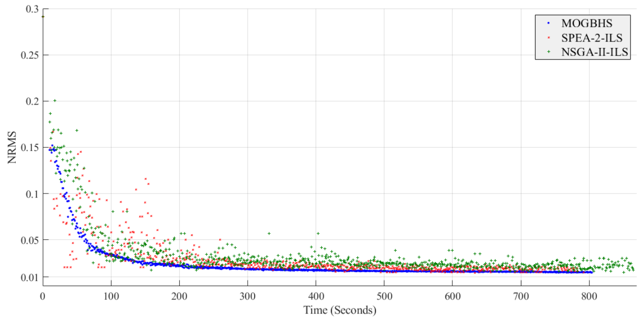


FIGURE 9. NRMS versus searching time using MOGBHS, SPEA-2-ILS and NSGA-II-ILS for the calibration of the low dimensional model.

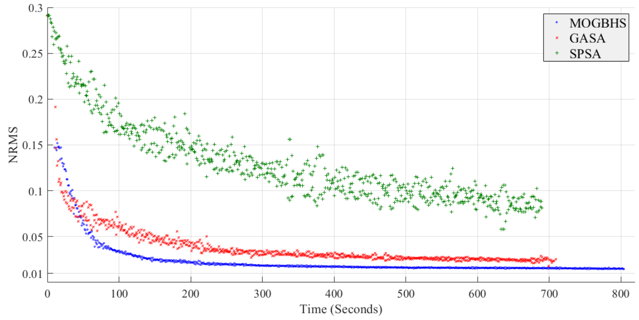


FIGURE 10. NRMS versus searching time using MOGBHS, GASA and SPSA for the calibration of the low dimensionality model.

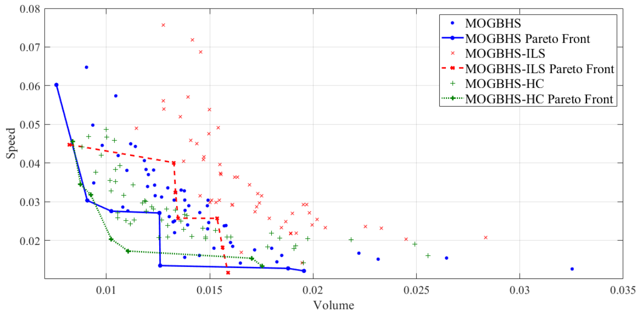


FIGURE 11. Pareto Fronts using the top three algorithms for the calibration of the low dimensionality model.

the MOGBHS which performed better than its memetic versions.

Table 5 shows the dominance of the algorithms based on their average performance. A solid circle denotes the dominance of the algorithm of that row with respect to the corresponding column. The hollow circle denotes the dominance of the algorithm associated with that column with respect to the corresponding row. The results above and below the diagonal have a significance level of 0.9 and 0.95, respectively. MOGBHS dominated all other algorithms but MOGBHS-HC. Consistent with the previous results, SPSA was dominated by all other algorithms. GASA only dominated SPSA.

Fig. 8 illustrates the convergence curve for the three top performing algorithms. In these experiments, W was 0.7. The performance of these algorithms was similar. However, MOGBHS provided less dispersion or more stability. Fig. 9 shows the convergence for MOGBHS as well as SPEA-2-ILS and NSGA-II-ILS. MOGBHS showed a less dispersion. Similarly, for some early iterations SPEA-2-ILS found NRMS close to the best found in short time (less than 100 seconds) because of its exploitation capabilities. However, there was significant dispersion.

Fig. 10 shows the convergence for MOGBHS, GASA and SPSA. GASA found better results early in the process (fist 40 seconds) because it included exploitation around the best solutions. However, MOGBHS converged faster (after 55 seconds) to even better solution and provided less dispersion with smaller average NRMS. SPSA did not converge fast and provided solutions with significant dispersion. Fig. 11 shows the Pareto Fronts using lines for the top three algorithms. As expected, the solutions along the Pareto Front obtained using MOGBHS and its variations were distributed uniformly. All other solutions tended to converge towards the front.

Fig. 12 shows the Pareto Fronts for MOGBHS, SPEA-2-ILS and NSGA-II-ILS. SPEA-2-ILS and NSGA-II-ILS found the same solution multiple times. That is, these algorithms generated solutions that were relatively the same in terms of the performance of the traffic flow model. In contrast, MOGBHS did not accept similar solutions in terms of the performance of the traffic flow model. In this

TABLE 5. Results for the Wilcoxon test using the low dimensionality model.

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
HC	(1)	-							○	○	○	○		○		○		●
SA	(2)		-						○	○	○	○		○		○		●
ILS	(3)			-					○	○	○	○				○		●
NSGA-II	(4)				-			○	○	○	○	○		○		○		●
NSGA-II-HC	(5)					-			○	○	○	○	●			○		●
NSGA-II-SA	(6)						-		○	○	○	○	●			○		●
NSGA-II-ILS	(7)							-	○	○	○	○	●			○		●
MOGBHS	(8)	●	●	●	●	●	●	●	-		●	●	●	●	●	●	●	●
MOGBHS-HC	(9)	●	●	●	●	●	●	●		-	●		●	●	●	●	●	●
MOGBHS-SA	(10)	●	●	●	●	●	●	●	○	○	-	○	●	●	●			●
MOGBHS-ILS	(11)	●	●	●	●	●	●	●	○		●	-	●	●	●	●	●	●
SPEA-2	(12)					○	○	○	○	○	○	○	-	○	○	○		●
SPEA-2-HC	(13)	●	●		●				○	○	○	○	●	-	●		●	●
SPEA-2-SA	(14)								○	○	○	○		○	-	○		●
SPEA-2-ILS	(15)	●	●	●	●	●	●	●	○	○		○	●		●	-	●	●
GASA	(16)								○	○	○	○		○		○	-	●
SPSA	(17)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-

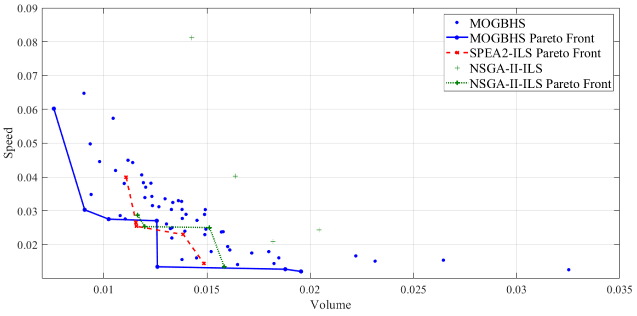


FIGURE 12. Pareto Fronts using MOGBHS, SPEA-2-ILS and NSGA-II-ILS for the calibration of the low dimensional model.

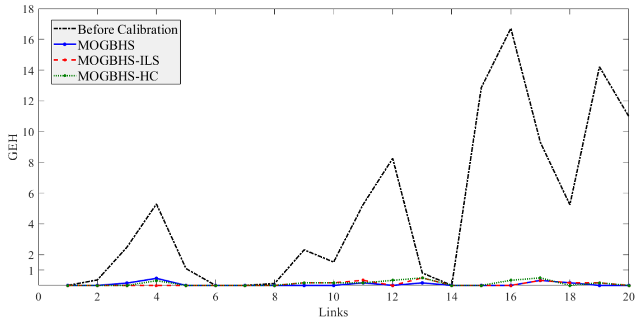


FIGURE 13. GEH before and after calibration using the top three algorithms and the low dimensionality model.

case, all solutions found by SPEA-2-ILS were part of the Pareto Front.

Fig. 13 provides the GEH before and after calibration using the top three algorithms. The GEH before calibration, denoted by the black pointed line without marker, was less than 5 for 50% of the links. After calibration using any of the top three algorithms, the GEH was less than 5 for all links.

2) MEDIUM DIMENSIONALITY MODEL

Table 6 shows the results obtained using all algorithms for the calibration of the medium dimensionality traffic flow model, Reno Network. Running times were similar other tan for SPEA-2, GASA, and SPSA. The algorithms involving NSGA-II and SPEA-2 provided similar trends to the ones obtained using the low dimensionality model. That is, their memetic versions provided better NRMS results in this case as well. The memetic versions of SPEA-2 converged much faster than the base algorithm. Similarly, MOGBHS provided the best average results followed by its memetic versions. However, MOGBHS-ILS provided the overall best results while MOGBHS-SA provided the smallest standard deviation. All results obtained using MOGBHS and its memetic versions were relatively similar. The memetic versions of NSGA-II and MOGBHS that provided the best results were those involving ILS for local search. The memetic versions of NSGA-II involving ILS and SA provided similar results. As in the previous set of experiments using the low dimensionality model, SPSA provided the worse overall results. However, all algorithms were able to find parameters that enabled to meet the required calibration criterion. All algorithms, other than SA, were able to produce GEH less than 5 for all links in the network.

Table 7 provides the rankings based on the Friedman test. Similar results were obtained for this set of experiments compared to the ones using the low dimensionality model. MOGBHS provided the best performance followed by its memetic versions. SPSA presented the worse performance.

The memetic versions of NSGA-II and SPEA-2 with the best performance were those using ILS for local search, followed by those including SA. In this experiment, GASA won several places in the ranking and was better than all algorithms except MOGBHS and its variations.

Table 8 shows the dominance for all algorithms using Wilcoxon test. MOGBHS was the only one that dominates with a 95% level of significance all other algorithms except MOGBHS-SA and MOGBHS-ILS. It was not possible to stablish a relationship of dominance

TABLE 6. Results for the calibration of the medium dimensional model (best in bold).

Algorithm	NRMS with W 0.7				Best GEH	Average searching time (mm:ss)
	Best	Worse	Average over 30 repetitions	Standard deviation		
HC	0.0754	0.1239	0.0977	0.011790708	100%	51:50
SA	0.0894	0.1142	0.1016	0.006022747	95%	52:20
ILS	0.0745	0.1281	0.0944	0.013749052	100%	50:55
NSGA-II	0.0756	0.1011	0.0938	0.005664098	100%	48:40
NSGA-II-HC	0.0754	0.1179	0.0957	0.009251681	100%	51:18
NSGA-II-SA	0.0808	0.1033	0.0923	0.004210363	100%	51:37
NSGA-II-ILS	0.0792	0.1054	0.0917	0.006710928	100%	51:35
MOGBHS	0.0700	**0.0873**	**0.0785**	0.003885580	100%	47:50
MOGBHS-HC	0.0719	0.0943	0.0819	0.005231409	100%	48:57
MOGBHS-SA	0.0743	0.0889	0.0804	**0.003097676**	100%	48:33
MOGBHS-ILS	**0.0675**	0.0917	0.0803	0.005166690	100%	48:43
SPEA-2	0.0864	0.1116	0.0965	0.007005492	100%	62:04
SPEA-2-HC	0.0759	0.1208	0.0917	0.010539557	100%	51:21
SPEA-2-SA	0.0845	0.0990	0.0900	0.004016990	100%	50:42
SPEA-2-ILS	0.0754	0.1091	0.0900	0.007914779	100%	50:30
GASA	0.0795	0.0932	0.0862	0.003521316	100%	61:00
SPSA	0.1051	0.2196	0.1526	0.025917078	91%	40:26

TABLE 7. Results for the friedman test in the medium dimensional model (best in bold).

Algorithm	Ranking	Algorithm	Ranking
MOGBHS	**2.2**	SPEA-2-HC	10
MOGBHS-SA	3.0667	NSGA-II	10.3333
MOGBHS-ILS	3.6667	ILS	10.9333
MOGBHS-HC	4.6	NSGA-II-HC	11
GASA	6.6	SPEA-2	11.9333
SPEA-2-ILS	8.2	HC	12.3333
SPEA-2-SA	8.4	SA	13.9333
NSGA-II-ILS	9.2667	SPSA	16.8667
NSGA-II-SA	9.6667		

between MOGBHS and MOGBHS-SA. However, MOGBHS dominated MOGBHS-ILS with a 90% of significance. Similar to the previous results, SPSA was dominated by all other algorithms. GASA dominated all single-state multi-objective algorithms and all algorithms involving NSGA-II and SPEA-2.

Fig. 14 shows the convergence obtained using the top three best performance algorithms. Consistent with previous results, MOGBHS found better and less spread solutions compare to its memetic versions, MOGBHS-ILS, and MOGBHS-SA. As the running time increased the magnitude of the spread decreased for the memetic algorithms, but it was never as small as the dispersion obtained using MOGBHS without any additional local search capability.

Fig. 15 shows the convergence for MOGBHS and the memetic algorithms SPEA-2-ILS and NSGA-II-ILS. As before, MOGBHS found better and less spread solutions. Initially (before 700 seconds), SPEA-2-ILS and NSGA-II-ILS found better solutions due to their local search capabilities. However, after 1500 seconds, the dominance of MOGBHS was clear. Even with longer running times, SPEA-2-ILS and NSGA-II-ILS did not get better solutions than MOGBHS.

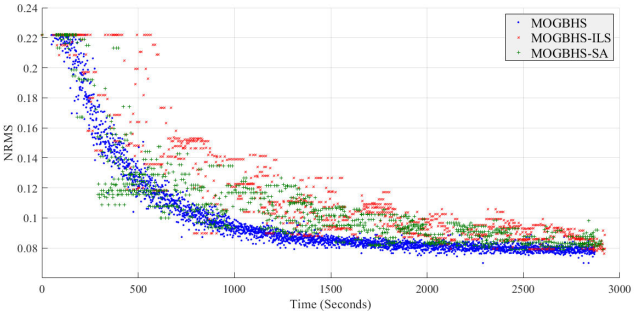


FIGURE 14. NRMS versus running time using the top three algorithms for the calibration of the medium dimensionality model.

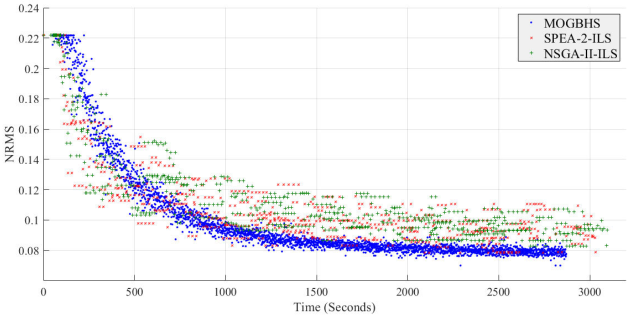


FIGURE 15. NRMS versus search time using MOGBHS, SPEA-2-ILS and NSGA-II-ILS for the calibration of the medium dimensionality model.

Fig. 16 shows the convergence for MOGBHS, GASA and SPSA. As before, GASA provided better solutions before the first 1000 seconds. However, MOGBHS provided the best convergence and less spread. As in the previous set of experiments, SPSA provided the worse convergence and its results were overly spread.

Fig. 17 shows the Pareto Front for the best three algorithms tested in this set of experiments. The Front is uniformly distributed, and other solutions tend to converge towards

TABLE 8. Results for the Wilcoxon test in the medium dimensionality model.

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
HC	(1)	-			○		○	○	○	○	○	○			○	○	○	●
SA	(2)		-		○	○	○	○	○	○	○	○		○	○	○	○	●
ILS	(3)			-					○	○	○	○			○	○	○	●
NSGA-II	(4)		●		-				○	○	○	○			○		○	●
NSGA-II-HC	(5)		●			-			○	○	○	○			○		○	●
NSGA-II-SA	(6)	●	●				-		○	○	○	○					○	●
NSGA-II-ILS	(7)		●					-	○	○	○	○	●				○	●
MOGBHS	(8)	●	●	●	●	●	●	●	-	●		●	●	●	●	●	●	●
MOGBHS-HC	(9)	●	●	●	●	●	●	●	○	-			●	●	●	●	●	●
MOGBHS-SA	(10)	●	●	●	●	●	●	●		○	-	○	●	●	●	●	●	●
MOGBHS-ILS	(11)	●	●	●	●	●	●	●			●	-	●	●	●	●	●	●
SPEA-2	(12)								○	○	○	○	-	○	○	○	○	●
SPEA-2-HC	(13)		●						○	○	○	○		-			○	●
SPEA-2-SA	(14)	●	●			●			○	○	○	○	●		-		○	●
SPEA-2-ILS	(15)	●	●						○		○	○	●			-		●
GASA	(16)	●	●	●	●	●	●	●	○	○	○	○	●	●	●		-	●
SPSA	(17)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-

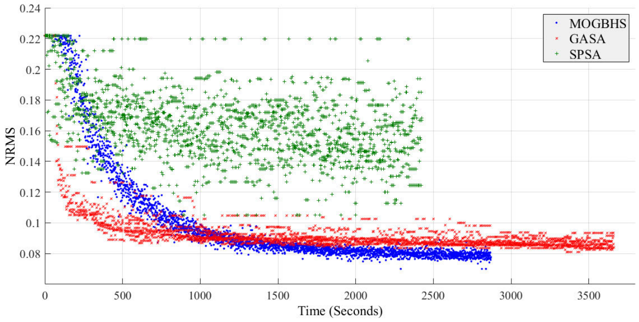


FIGURE 16. NRMS versus search time using MOGBHS, GASA, and SPSA for the calibration of the medium dimensionality model.

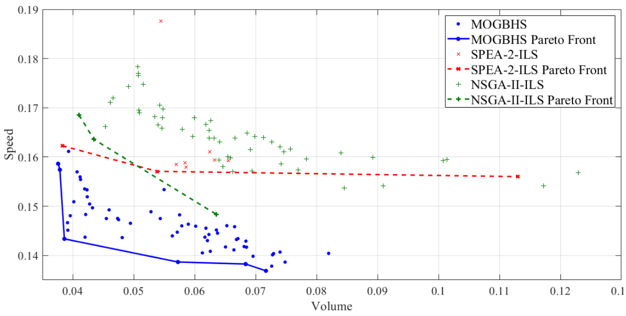


FIGURE 18. Pareto Fronts using MOGBHS, SPEA-2-ILS and NSGA-II-ILS for the calibration of the medium dimensionality model.

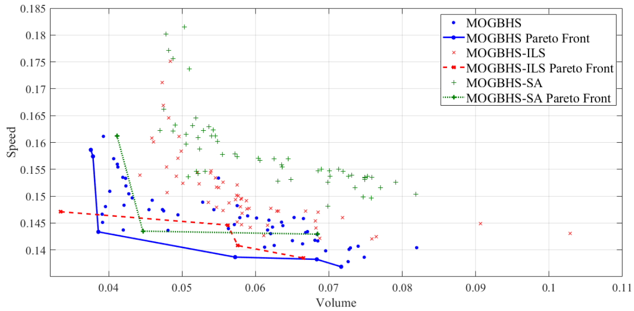


FIGURE 17. Pareto Fronts using the best three algorithms for calibration of the medium dimensionality model.

the Front. In general, MOGBHS performed better followed by MOGBHS-ILS.

Fig. 18 comparers the Pareto Fronts for MOGBHS, SPEA-2-ILS and NSGA-II-ILS. For SPEA-2-ILS and NSGA-II-ILS the number of similar solutions was much less compared with the previous set of experiments. This is probably because the search space in this case was much larger than in the previous set of experiments. In general, MOGBHS performed better followed by SPEA-2-ILS. In addition, the Pareto Front for MOGBHS included more solutions.

Fig. 19 shows the GEH statistics before and after calibration using the best three optimization algorithms. The GEH was calculated using the best results in each case. The black pointed line without marker provides the results before calibration with values less than 5 for 47% of links. All algorithms were able to calibrate the model with a GEH less than 5 for all links.

3) HIGH DIMENSIONALITY MODEL

Table 9 provides a summary of the results obtained using all algorithms. All search times were similar other than for the multi-objective NSGA-II, SPEA-2, and GASA. In contrast to the results from the previous sets of experiments, the memetic versions of NSGA-II and SPEA-2 using HC provided better NRMS with less search time. As with the previous sets of experiments, MOGBHS provided the best results. SPSA provided the smallest searching time and the worse performance in terms of convergence and NRMS. GASA took the longest searching time after SPEA-2. The average NRMS obtained using GASA was only outperformed by MOGBHS and its

TABLE 9. Results for the calibration of the high dimensionality traffic flow model (best in bold).

Algorithm	NRMS with W 0.7				Best GEH	Average searching time (mm:ss)
	Best	Worse	Average over 30 repetitions	Standard deviation		
HC	0.0887	0.1363	0.1080	0.014850969	93%	03:25
SA	0.1784	0.3318	0.2766	0.066797703	69%	03:24
ILS	0.0937	0.1277	0.1069	0.011912253	95%	03:35
NSGA-II	0.0804	0.1283	0.1029	0.015667031	93%	04:20
NSGA-II-HC	0.0901	0.1195	0.1018	0.010196162	95%	03:32
NSGA-II-SA	0.1065	0.1599	0.1293	0.019868558	92%	03:47
NSGA-II-ILS	0.0852	0.1513	0.1122	0.018561540	96%	03:46
MOGBHS	**0.0676**	**0.0883**	**0.0746**	0.007524712	**97%**	03:22
MOGBHS-HC	0.0739	0.1279	0.0872	0.015689967	**97%**	03:01
MOGBHS-SA	0.0809	0.1261	0.0969	0.014504441	95%	03:04
MOGBHS-ILS	0.0696	0.1182	0.0861	0.012578814	**97%**	03:13
SPEA-2	0.0795	0.1345	0.1050	0.01910512	96%	04:33
SPEA-2-HC	0.0801	0.1092	0.0957	0.010866424	96%	03:38
SPEA-2-SA	0.1167	0.1517	0.1348	0.012730534	95%	03:34
SPEA-2-ILS	0.0939	0.1191	0.1079	**0.007194136**	95%	03:51
GASA	0.0821	0.1486	0.0968	0.015549272	93%	04:33
SPSA	0.3125	0.3318	0.3305	0.004964004	42%	02:50

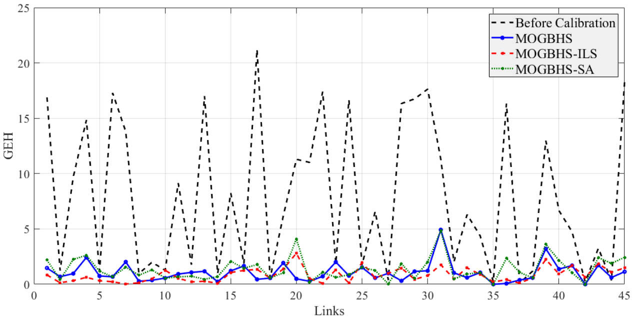


FIGURE 19. GEH before and after calibration using the best three optimization algorithms for the calibration of the high dimensionality model.

TABLE 10. Friedman test in the high dimensionality model (best in bold).

Algorithm	Ranking	Algorithm	Ranking
MOGBHS	1.6	HC	9.4
MOGBHS-ILS	3.9	ILS	9.5
MOGBHS-HC	4	SPEA-2-ILS	10.1
MOGBHS-SA	6.5	NSGA-II-ILS	10.4
GASA	6.6	NSGA-II-SA	12.8
SPEA-2-HC	7.2	SPEA-2-SA	13.7
NSGA-II-HC	7.9	SA	16.1
NSGA-II	8.1	SPSA	16.9
SPEA-2	8.3		

memetic versions. All algorithms other than SA and SPSA were able to meet the required calibration criterion. All algorithms that were able to calibrate the model achieved GEH less than 5 for at least 90% of the links.

Table 10 shows the results of the Friedman test. MOGBHS and SPSA provided the best and worse rankings, respectively. The memetic versions of NSGA-II and SPEA-2 using HC provided the best rankings compared to the standard and other memetic versions of these algorithms. GASA kept its ranking compared with the previous sets of experiments. The three sets of experiments performed in this research show that as the complexity of the traffic flow model increased, MOGBHS provided better ranking. This implies that the joint use of the harmony search, the swarm approach (taken from PSO) for local search around best solutions, and the use of Pareto domination and Crowding distance to find solutions was highly effective for solving calibration problems of high dimensionality. All results revealed that additional local search did not improve but rather negatively affect the search capabilities of MOGBHS.

Table 11 shows the dominance results obtained using the Wilcoxon test. As in the previous two sets of experiments, MOGBHS dominated all other algorithms with a 95% level of significance. Similarly, SPSA is dominated by all other algorithms. GASA dominated all algorithms other tan MOGBHS and its memetic versions. Most algorithms involving SA, were dominated by all other.

Fig. 20 shows the convergence of the three best algorithms used in this set of experiments. It shows similar trends to the ones observed in the previous sets of experiments. MOGBHS found the best solutions and they were less dispersed. MOGBHS-ILS and MOGBHS-HC produced better solutions than MOGBHS only at the beginning of the search process. Similarly, the spread of the solutions was reduced over time.

Fig. 21 shows the convergence for MOGBHS, SPEA-2-HC, and NSGA-II-HC. MOGBHS provided less disperse solutions. For early iterations (execution time less than 3000 seconds), the exploitation approaches in SPEA-2-ILS and NSGA-II-ILS helped find better solutions than MOGBHS. However, as the number of iterations increased, MOGBHS outperformed all other algorithms.

Fig. 22 shows the convergence for MOGBHS, GASA and SPSA. Initially (execution time less than 5000 seconds), GASA found better solutions. However, as before, MOGBHS

TABLE 11. Results for the Wilcoxon test in the high dimensionality model.

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
HC	(1)	-	●				●		○	○	○	○			●		○	●
SA	(2)	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
ILS	(3)		●	-			●		○	○		○			●			●
NSGA-II	(4)		●		-		●		○	○		○			●			●
NSGA-II-HC	(5)		●			-	●		○	○		○			●	●		●
NSGA-II-SA	(6)	○	●	○	○	○	-	○	○	○	○	○	○	○		○	○	●
NSGA-II-ILS	(7)		●				●	-	○	○	○	○		○	●		○	●
MOGBHS	(8)	●	●	●	●	●	●	●	-	●	●	●	●	●	●	●	●	●
MOGBHS-HC	(9)	●	●	●			●	●	○	-	●		●		●	●		●
MOGBHS-SA	(10)		●				●	●	○	○	-				●	●		●
MOGBHS-ILS	(11)	●	●	●	●	●	●	●	○		●	-	●		●	●	●	●
SPEA-2	(12)		●						○	○	○		-		●			●
SPEA-2-HC	(13)		●				●	●	○		○			-	●	●		●
SPEA-2-SA	(14)	○	●	○	○	○		○	○	○	○	○	○	○	-	○	○	●
SPEA-2-ILS	(15)		●				●		○	○	○	○		○	●	-		●
GASA	(16)		●				●		○		○				●		-	●
SPSA	(17)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	-

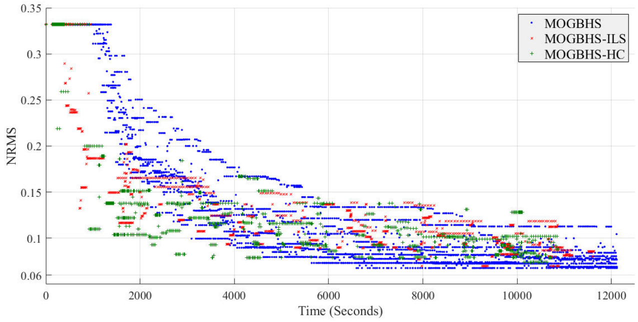


FIGURE 20. **NRMS versus search time using the best three algorithms for the calibration of the high dimensionality model.**

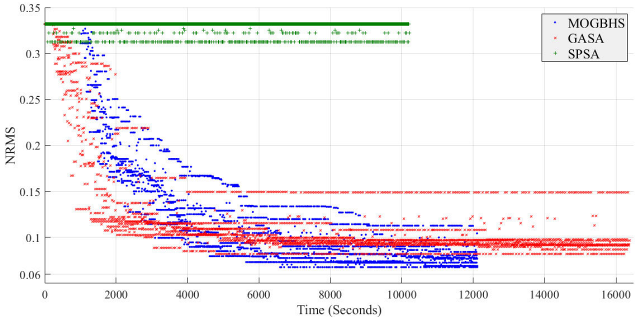


FIGURE 22. **NRMS versus search time using MOGBHS, GASA, and SPSA for the calibration of the high dimensionality model.**

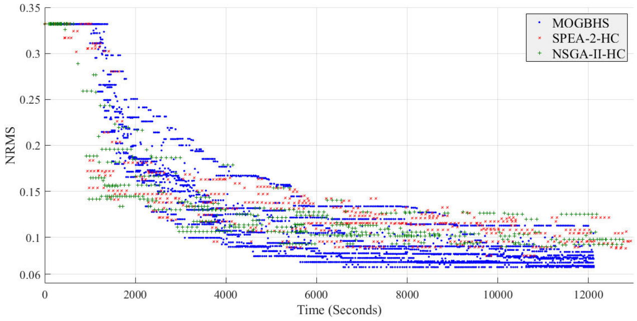


FIGURE 21. **NRMS versus search time using MOGBHS, SPEA-2-HC and NSGA-II-HC for the calibration of the high dimensionality model.**

outperformed all other algorithms with smaller value for the objective function and less dispersion of the solutions. In addition, MOGBHS converged much faster than the other algorithms. It was not possible to find hyperparameters for SPSA that can provide competitive solutions. This illustrated a major challenge for using SPSA to solve this type of highly complex optimization problems.

Fig. 23 shows the Pareto front obtained using the three best performing algorithms. As before, the solutions were more or less uniformly distributed and those that were not part of the Pareto Front tended in its direction. In this case, it was not straight forward to decide between the solutions provided by MOGBHS and MOGBHS-ILS.

Fig. 24 shows the Pareto front for MOGBHS, NSGA-II-HC, and SPEA-2-HC. MOGBHS provided the best solutions. Relative to the first experiment, the number of repeated solutions decreased significantly for SPEA-2-HC and NSGA-II-HC. This was probably because the search space was much larger for this experiment. SPEA-2-HC provided the most diverse Pareto front. However, these solutions were very far compared to the ones obtained by MOGBHS and NSGA-II-HC.

Fig. 25 provides the GEH before and after calibration for the best three algorithms. GEH was calculated using the results from the best solution in each case. The black pointed line provided the results before calibration when only 40% of the links have GEH less than 5. After calibration using MOGBHS, MOGBHS-ILS and MOGBHS-HC the GEH was less than 5 for 95-97% of the links.

G. SENSITIVITY ANALYSIS FOR MOGBHS

A sensitivity analysis regarding the hyperparameters required by MOGBHS was performed to understand their impact on

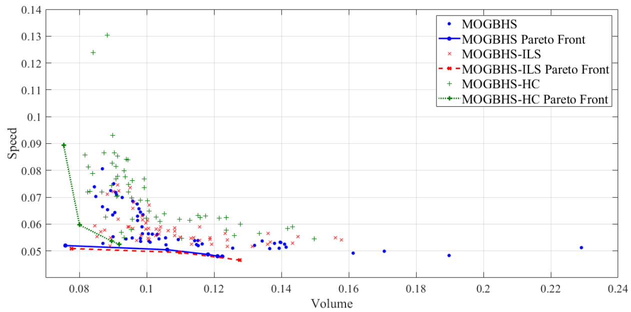


FIGURE 23. Pareto Fronts using the best three performing algorithms for the calibration of the high dimensionality model.

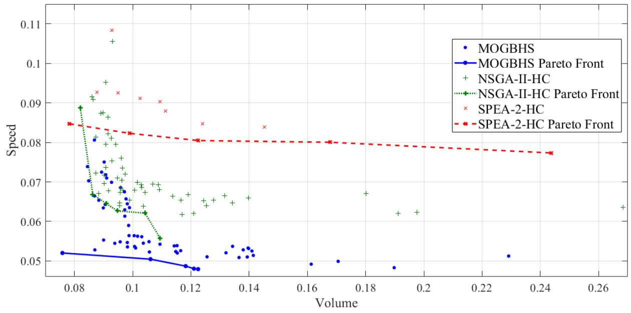


FIGURE 24. Pareto Fronts using MOGBHS, SPEA-2-HC and NSGA-II-HC for the calibration of the high dimensionality model.

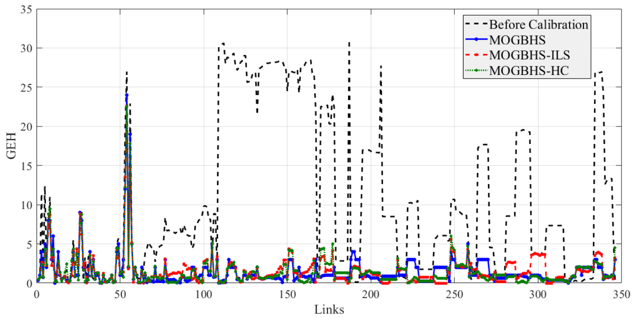


FIGURE 25. GEH using MOGBHS, MOGBHS-ILS and MOGBHS-HC before and after calibrating the high dimensionality model.

the results. The analysis involved the low complexity model as well as covering arrays with hyperparameters provided in Table 2 and 30 iterations. The results were classified using the criterion in Table 12.

A decision tree was obtained using the J48 algorithm [46] to determine that many iterations enables finding great solutions without having to fine-tune the other hyperparameters used by MOGBHS. When a small number of iterations were used, MOGBHS required that hyper parameter HMCR be less or equal to 0.8 (more exploration) to obtain acceptable NRMS. If HMCR was between 0.8 and 0.9 and the size of the harmony memory was less or equal to 20, MOGBHS was able to find acceptable solutions. If we used relatively large populations (more than 20), MOGBHS was able to obtain great solutions and the quality increased with the size of HMCR. Hyperparameters $\mathrm{PAR}_{\min}$ and $\mathrm{PAR}_{\max}$ did not affect the performance of MOGBHS to solve the proposed calibration problem.

TABLE 12. Categories based on NRMS.

Category	Range
Excellent	$NRMS <= 0.016$
Acceptable	$0.016 < NRMS <= 0.017$
Poor	$NRMS > 0.017$

IV. CONCLUSIONS AND FUTURE WORK

This research investigated the performance of multiple optimization algorithms for the calibration of microscopic traffic flow simulation models. Three simulation models of different dimensionality and complexity were used to test the performance of the algorithms. Among the tested algorithms, MOGBHS provided the best performance in all cases. The basic MOGBHS was expanded to include memetic characteristics which included three local search algorithms, HC, SA, and ILS, to explore opportunities for further improvement. Other multi-objective algorithms that were evaluated included NSGA-II and SPEA-2 which were also coupled with HC, SA, and ILS. All multi-objective algorithms were able to meet the calibration criterion with and without adding local search capabilities. The Friedman and Wilcoxon tests showed that the algorithm with the best statistical performance was MOGBHS followed by its memetic versions.

MOGBHS provided better results than its memetic versions probably because its Particle Swarm Optimization search process performs the appropriate exploitation around the current best solution. Hence, the addition of local search algorithms only helped the search process at the beginning. As required, the basic MOGBHS performs more exploitation as the number of iterations increases. The results showed that SPSA was outperformed in all cases by all other algorithms. This is probably because SPSA does not provide mechanisms to leave local minima. New solutions are generated by SPSA using a gradient based approach while modifying all components of an existing solution.

In terms of running time, overall SPSA was the fastest algorithm followed by MOGBHS. However, GASA was faster than MOGBHS for the calibration of the low dimensional traffic flow simulation model. In general, as the dimensionality increased GASA provided better convergence at the beginning of the search process, but then, MOGBHS provided better and more stable convergence achieving the best results.

A sensitivity analysis including the best algorithm, MOGBHS, was performed. The analysis revealed that, other than for the harmony memory size, using default hyperparameters provides similar results. That is, MOGBHS was not much sensitive to changes in its hyperparameters. This is an advantage because little time is required to implement and use an effective MOGBHS. Most optimization algorithms require fine-tuning hyperparameters to be able to find the best solutions for a problem.

Future research is recommended to implement the solution framework using as objective functions the cumulative of the absolute difference between simulated and actual metrics rather than NRMS. This is because metaheuristics optimization algorithms in general seem to perform better using absolute differences rather than the NRMS. Another interesting extension is the explicit consideration of more aspects of the real system such as minimization of queue lengths.

Other opportunities to improve the search process include using NSGA-II and MOGBHS with a density function involving the nearest neighbor as in SPEA-2. This is with the objective of generating a more diverse Pareto Front as is the case of SPEA-2. Another opportunity is to borrow the exploitation approach used in GASA to search the neighborhood of the best solution. Other available multi-objective algorithms that were not tested but have potential for solving the calibration problem include, among others, MOEA/D (a decomposition-based evolutionary multi-objective algorithm), OMOPSO (a multi-objective algorithm based on Particle Swarm Optimization), ε-NSGA-II, and NSGA-III.

REFERENCES

[1] D. Henclewood, W. Suh, M. O. Rodgers, R. Fujimoto, and M. P. Hunter, "A calibration procedure for increasing the accuracy of microscopic traffic simulation models," *Simulation*, vol. 93, no. 1, pp. 35–47, Jan. 2017, doi: 10.1177/0037549716673723.

[2] A. Paz, V. Molano, E. Martinez, C. Gaviria, and C. Arteaga, "Calibration of traffic flow models using a memetic algorithm," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 432–443, Jun. 2015, doi: 10.1016/j.trc.2015.03.001.

[3] D. K. Hale, C. Antoniou, M. Brackstone, D. Michalaka, A. T. Moreno, and K. Parikh, "Optimization-based assisted calibration of traffic simulation models," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 100–115, Jun. 2015, doi: 10.1016/j.trc.2015.01.018.

[4] M. Yu and W. D. Fan, "Calibration of microscopic traffic simulation models using metaheuristic algorithms," *Int. J. Transp. Sci. Technol.*, vol. 6, no. 1, pp. 63–77, Jun. 2017, doi: 10.1016/j.ijtst.2017.05.001.

[5] B. Ciuffo and V. Punzo, "'No free lunch' theorems applied to the calibration of traffic simulation models," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 553–562, Apr. 2014, doi: 10.1109/TITS.2013.2287720.

[6] P. Moscato and C. Cotta, "An accelerated introduction to memetic algorithms," in *Handbook Metaheuristics*, M. Gendreau J.-Y. Potvin, Eds. Cham, Switzerland: Springer, 2019, pp. 275–309.

[7] J. C. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins APL Tech. Dig.*, *Johns Hopkins APL Tech. Dig.*, vol. 19, no. 4, pp. 482–492, 1998.

[8] C. Cobos, C. Daza, C. Martínez, M. Mendoza, C. Gaviria, C. Arteaga, and A. Paz, "Calibration of microscopic traffic flow simulation models using a memetic algorithm with Solis and wets local search chaining (MA-SW-chains)," in *Advances in Artificial Intelligence—IBERAMIA*. Cham, Switzerland: Springer, 2016, pp. 365–375.

[9] G. Amirjamshidi and M. J. Roorda, "Multi-objective calibration of traffic microsimulation models," *Transp. Lett.*, vol. 11, no. 6, pp. 311–319, Jul. 2019, doi: 10.1080/19427867.2017.1343763.

[10] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Math. Oper. Res.*, vol. 6, no. 1, pp. 19–30, 1981.

[11] J. J. Durillo and A. J. Nebro, "JMetal: A java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, Oct. 2011, doi: 10.1016/j.advengsoft.2011.05.014.

[12] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013, doi: 10.1109/TSMCB.2012.2231860.

[13] V. Papathanasopoulou, I. Markou, and C. Antoniou, "Online calibration for microscopic traffic simulation and dynamic multi-step prediction of traffic speed," *Transp. Res. C, Emerg. Technol.*, vol. 68, pp. 144–159, Jul. 2016, doi: 10.1016/j.trc.2016.04.006.

[14] M. Sparnaaij, D. C. Duives, V. L. Knoop, and S. P. Hoogendoorn, "Multiobjective calibration framework for pedestrian simulation models: A study on the effect of movement base cases, metrics, and density levels," *J. Adv. Transp.*, vol. 2019, Jul. 2019, Art. no. 5874085, doi: 10.1155/2019/5874085.

[15] E. T. Cascan, J. Ivanchev, D. Eckhoff, A. Sangiovanni-Vincentelli, and A. Knoll, "Multi-objective calibration of microscopic traffic simulation for highway traffic safety," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 4548–4555.

[16] G. Dhiman, M. Soni, H. M. Pandey, A. Slowik, and H. Kaur, "A novel hybrid hypervolume indicator and reference vector adaptation strategies based evolutionary algorithm for many-objective optimization," *Eng. Comput.*, Feb. 2020, doi: 10.1007/s00366-020-00986-0.

[17] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. D. S. Coelho, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Syst. Appl.*, vol. 47, pp. 106–119, Apr. 2016, doi: 10.1016/j.eswa.2015.10.039.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[19] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2," in *Parallel Problem Solving From Nature—PPSN VIII*. Berlin, Germany, 2004, pp. 742–751.

[20] S. Dutta and K. N. Das, "A survey on Pareto-based EAs to solve multi-objective optimization problems," in *Soft Computing for Problem Solving*. Singapore, 2019, pp. 807–820.

[21] N. Gunantara, "A review of multi-objective optimization: Methods and its applications," *Cogent Eng.*, vol. 5, no. 1, Jul. 2018, Art. no. 1502242, doi: 10.1080/23311916.2018.1502242.

[22] C. Cobos, C. Erazo, J. Luna, M. Mendoza, C. Gaviria, C. Arteaga, and A. Paz, "Multi-objective memetic algorithm based on NSGA-II and simulated annealing for calibrating CORSIM micro-simulation models of vehicular traffic flow," in *Advances in Artificial Intelligence*. Cham, Switzerland: Springer, 2016, pp. 468–476.

[23] M. Karimi, M. Miriestahbanati, H. Esmaeeli, and C. Alecsandru, "Multi-objective stochastic optimization algorithms to calibrate microsimulation models," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 4, pp. 743–752, Apr. 2019, doi: 10.1177/0361198119838260.

[24] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[25] N. V. Dung, B. Merz, A. Bárdossy, T. D. Thang, and H. Apel, "Multi-objective automatic calibration of hydrodynamic models utilizing inundation maps and gauge data," *Hydrol. Earth Syst. Sci.*, vol. 15, no. 4, pp. 1339–1354, Apr. 2011, doi: 10.5194/hess-15-1339-2011.

[26] A. J. Siade, B. Rathi, H. Prommer, D. Welter, and J. Doherty, "Using heuristic multi-objective optimization for quantifying predictive uncertainty associated with groundwater flow and reactive transport models," *J. Hydrol.*, vol. 577, Oct. 2019, Art. no. 123999, doi: 10.1016/j.jhydrol.2019.123999.

[27] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. Coello Coello, and F. Herrera, "Bio-inspired computation: Where we stand and what's next," *Swarm Evol. Comput.*, vol. 48, pp. 220–250, Aug. 2019, doi: 10.1016/j.swevo.2019.04.008.

[28] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, May 2008, doi: 10.1016/j.amc.2007.09.004.

[29] V. Hajipour, S. H. A. Rahmati, S. H. R. Pasandideh, and S. T. A. Niaki, "A multi-objective harmony search algorithm to optimize multi-server location–allocation problem in congested systems," *Comput. Ind. Eng.*, vol. 72, pp. 187–197, Jun. 2014, doi: 10.1016/j.cie.2014.03.018.

[30] S. Salcedo-Sanz, D. Manjarrés, Á. Pastor-Sánchez, J. Del Ser, J. A. Portilla-Figueras, and S. Gil-López, "One-way urban traffic reconfiguration using a multi-objective harmony search approach," *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3341–3350, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417412012845, doi: 10.1016/j.eswa.2012.12.043.

[31] S. Sivasubramani and K. S. Swarup, "Multi-objective harmony search algorithm for optimal power flow problem," *Int. J. Electr. Power Energy Syst.*, vol. 33, no. 3, pp. 745–752, Mar. 2011, doi: 10.1016/j.ijepes.2010.12.031.

[32] E. Ruano, C. Cobos, and J. Torres-Jimenez, "Transit network frequencies-setting problem solved using a new multi-objective global-best harmony search algorithm and discrete event simulation," in *Advances in Soft Computing*. Cham, Switzerland: Springer, 2017, pp. 341–352.

[33] E. Ruano-Daza, C. Cobos, J. Torres-Jimenez, M. Mendoza, and A. Paz, "A multiobjective bilevel approach based on global-best harmony search for defining optimal routes and frequencies for bus rapid transit systems," *Appl. Soft Comput.*, vol. 67, pp. 567–583, Jun. 2018, doi: 10.1016/j.asoc.2018.03.026.

[34] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. (2009). *Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition*. [Online]. Available: https://www.al-roomi.org/multimedia/CEC_Database/CEC2009/MultiObjectiveEA/CEC2009_MultiObjectiveEA_TechnicalReport.pdf

[35] C. Ordoñez, E. Ruano, C. Cobos, H. Ordoñez, and A. Ordoñez, "Comparative analysis of MOGBHS with other state-of-the-art algorithms for multi-objective optimization problems," in *Advances in Soft Computing*. Cham, Switzerland: Springer, 2018, pp. 154–170.

[36] A. Paz, V. Molano, and C. Gaviria, "Calibration of CORSIM models considering all model parameters simultaneously," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 1417–1422.

[37] *CORSIM User's Guide: McTrans. Transportation Institute*, University of Florida, Version 6.0. ed., ITT Industries, Gainesville, FL, USA, 2006.

[38] A. Paz, V. Molano, and J. Sanchez-Medina, "Holistic calibration of microscopic traffic flow models: Methodology and real world application studies," in *Engineering and Applied Sciences Optimization: Dedicated to the Memory of Professor*, M. G. Karlaftis, N. D. Lagaros, and M. Papadrakakis, Eds. Cham: Springer, 2015, pp. 33–52.

[39] F. L. Mannering and S. S. Washburn, *Principles of Highway Engineering and Traffic Analysis*, 6th ed. Hoboken, NJ, USA: Wiley, 2016.

[40] P. Holm, D. Tomich, J. Sloboden, and C. F. Lowrance, "Traffic analysis toolbox: Guidelines for applying CORSIM microsimulation modeling software," Tech. Rep., 2007.

[41] I. Izquierdo-Marquez, J. Torres-Jimenez, B. Acevedo-Juárez, and H. Avila-George, "A greedy-metaheuristic 3-stage approach to construct covering arrays," *Inf. Sci.*, vols. 460–461, pp. 172–189, Sep. 2018, doi: 10.1016/j.ins.2018.05.047.

[42] D. Hadka, *MOEA Framework—A Free and Open Source Java Framework for Multiobjective Optimization*, 2.12 ed., 2011. [Online]. Available: https://sourceforge.net/p/moeaframework/bugs/_discuss/thread/12d4a637/4a31/attachment/MOEAFramework-2.1-ManualFixed.pdf

[43] D. Hadka, P. M. Reed, and T. W. Simpson, "Diagnostic assessment of the borg MOEA for many-objective product family design problems," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–10.

[44] A. E. Eiben and S. K. Smit, "Evolutionary algorithm parameters and methods to tune them," in *Autonomous Search*, Y. Hamadi, E. Monfroy, F. Saubion, Eds. Berlin, Germany: Springer, 2012, pp. 15–36.

[45] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010, doi: 10.1016/j.ins.2009.12.010.

[46] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, Nov. 2009, doi: 10.1145/1656274.1656278.

CARLOS COBOS (Senior Member, IEEE) received the B.S. degree in systems engineering and the M.S. degree in informatics from the Industrial University of Santander, Bucaramanga, Colombia, in 1995 and 2004, respectively, and the Ph.D. degree in systems engineering and computation from the National University of Colombia, Bogota, in 2014. From 1995 to 1999, he was a Lecturer with the Industrial University of Santander. Since 1999, he has been a Titular Professor with the Computer Science Department, University of Cauca, Colombia. He has coauthored three books, 16 JCR articles, and more than 50 SJR and Scielo articles. He has advised two Ph.D. and 18 master's graduates. His research interests include data mining, computational intelligence, deep learning, machine learning, natural language processing, information retrieval, and text mining. He is classified as a Senior Researcher in the National System of Researchers, Colombia.

ALEXANDER PAZ (Member, IEEE) is currently the Transport Main Roads Chair with the Queensland University of Technology. Before joining QUT, he was an Associate Professor of civil engineering and the Director of the Transportation Research Center, University of Nevada. He is also a Chartered Professional Engineer in Australia and a Professional Engineer Licensed in the State of Nevada. He has a strong background in transport engineering, transport planning, and traffic safety. He has significant experience developing methods, algorithms and software tools for the management of highway infrastructure, the analysis and evaluation of transport systems, and the deployment of intelligent transportation systems. His work including data warehouses and software applications has been adopted by industry at an international scale. Two of his inventions are presently being used by industry; one presently patented, and the second is under review by the U.S. Patent and Trademark Office. A third invention received a patent from the U.S. Patent and Trademark Office and was licensed for commercialization.

JULIO LUNA received the B.S. degree in computer science from the University of Cauca, Popayan, Colombia, in 2017, where he is currently pursuing the M.S. degree in computer science.

His research interests include computational intelligence, data mining, machine learning, and deep learning.

CRISTIAN ERAZO received the B.S. degree in computer science from the University of Cauca, Popayan, Colombia, in 2017. He is currently pursuing the M.S. degree in computer science with the Center for Research and Advanced Studies, National Polytechnic Institute of Mexico. His research interests include computational intelligence, multi-objective optimization, and machine learning.

MARTHA MENDOZA received the B.S. degree in systems engineering and the M.S. degree in informatics from the Industrial University of Santander, Bucaramanga, in 1995 and 2004, respectively, and the Ph.D. degree in systems engineering and computation from the National University of Colombia, Bogota, in 2016.

From 1995 to 1999, she was a Lecturer with the Industrial University of Santander. Since 1999, she has been a Professor with the Computer Science Department, University of Cauca, Colombia. She has coauthored ten JCR articles and more than 40 SJR and Scielo articles. She has advised eight M.Sc. and more than 20 B.Sc. theses graduates. Her research interests include data warehousing, computational intelligence, and machine learning. She is classified as a Senior Researcher in the National System of Researchers, Colombia.