



Calibration of Microscopic Traffic Simulation in an Urban Environment Using GPS-Data

Christopher Stang^{1,2} , and Klaus Bogenberger² 

¹ ZF Friedrichshafen AG, Germany

² Chair of Traffic Engineering and Control, Technical University of Munich, Germany

*Correspondence: Christopher Stang, christopher2.stang@tum.de

Abstract. Accurate traffic models are of decisive importance for well-founded traffic engineering and represent the basic framework for comprehensive simulation studies as modelling of traffic demand. Using traffic count and speed measurements of road segments is a common approach for the calibration of a realistic traffic simulation although the data acquisition process can be at very extensive costs. From an academical point of view, there have been many studies addressing the problem of calibration. In this respect, the microscopic simulation software SUMO offers the usage of the tools flowrouter and routesampler for generating network simulations on the base of traffic count measurements. In this paper, we propose a robust method for the calibration of microscopic traffic simulations by using vehicle count and speed measurements from collected GPS-data. The developed approach is a two-step optimization process: The application of integer linear programming (ILP) as a priori optimization is followed by adopting an evolutionary algorithm for minimizing the a posteriori deviation between real and simulated traffic data. As a proof of concept, the proposed method is tested in a subnetwork model of the inner city of Friedrichshafen and compared with the ready-to-use tools from SUMO. The suggested method indicates a promising correlation between simulated and real traffic data showing better calibration results in comparison to the aforementioned functions SUMO provides. Since the approach is network-independent, it also offers the possibility of large-scale traffic calibration.

Keywords: Traffic Simulation, Calibration, Integer Linear Programming, Evolutionary Algorithm

1. Introduction

As a part of traffic engineering, the realistic modelling of travel demand is an imperative component for investigating and evaluating different traffic policies and measures. A common approach of generating simulative traffic demand is the usage of real traffic data. The origin of such data can be manifold ranging from induction loops, manual counts or even centrally collected GPS-data of vehicles. Independent of the source of the data, traffic demand modelling is always combined with an optimization and a calibration process since reality must be reproduced in the best possible way. In this respect, Ciuffo et al. [1] provide a very good overview of calibration methods for traffic simulations.

SUMO offers different tools to generate traffic demand from traffic counts, namely flowrouter and routesampler. Both tools use edge-based counts for producing demand, whereby the idea of both approaches is different. As described in [2], the tool flowrouter calculates routes and corresponding traffic flows by solving a maximum flow problem with given counting

data. On the other hand, routesampler uses an initial set of routes in combination with counting data and samples the routes without exceeding the target counts. In this sense, the sampling process is formulated as a linear programming problem. However, both approaches just consider a priori optimization resulting in not negligible deviations compared to the specified count values. Furthermore, both tools do not offer any possibility for speed calibration.

In this paper, we propose a robust method for the calibration of microscopic traffic simulations by using vehicle count and speed measurements from collected GPS-data. The developed approach is a two-step optimization process: The application of integer linear programming (ILP) as a priori optimization is followed by adopting an evolutionary algorithm for minimizing the a posteriori deviation between real and simulated traffic data. As a proof of concept, the proposed method is tested in a subnetwork model of the inner city of Friedrichshafen and compared with the ready-to-use tools from SUMO. The remainder of the paper is organized as follows. The second section gives an overview of the considered network and demand data for the city of Friedrichshafen. In the third section the proposed methodology for the optimization and calibration is presented. The fourth section shows the results of the study, and in the final section a conclusion and outlook are given.

2. Network

For simulative designing of travel demand, a subnetwork of Friedrichshafen with one main track ranging from Henri-Dunant-Strasse to Löwentaler-Strasse is chosen. The selected track has a total length of approximately three km. In this respect, figure 2-1 shows the network described with its main track colored in red and the corresponding SUMO network. For getting realistic vehicle traffic assumptions, a dataset for 100 segments along the track with its adjacent access and exit roads was requested at the GPS-supplier TomTom. Eventually, the final dataset entails a timeframe of 36 months from 2017 to 2019 containing representative information about vehicle counts and speed distribution for weekdays and weekend days. For the sake of simplification, the dataset of each segment finally is aggregated resulting in one representative working day and weekend day with a time interval of two hours for each month of each year.

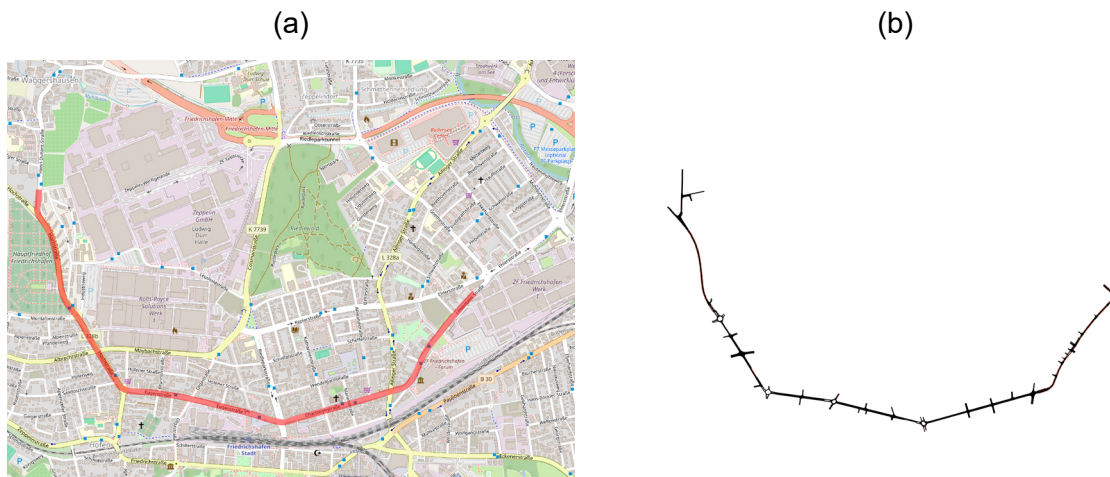


Figure 2-1. Real network of Friedrichshafen (a) [3], SUMO network of Friedrichshafen (b)

Furthermore, the obtained GPS-data are extrapolated to the total traffic population by TomTom. Following picture shows the aggregation process exemplarily for one of the 36 months. It has to be mentioned that the final aggregation process as well as the data preparation and network generation in SUMO were carried out not by the authors but by TomTom and a department at ZF Friedrichshafen AG, respectively.

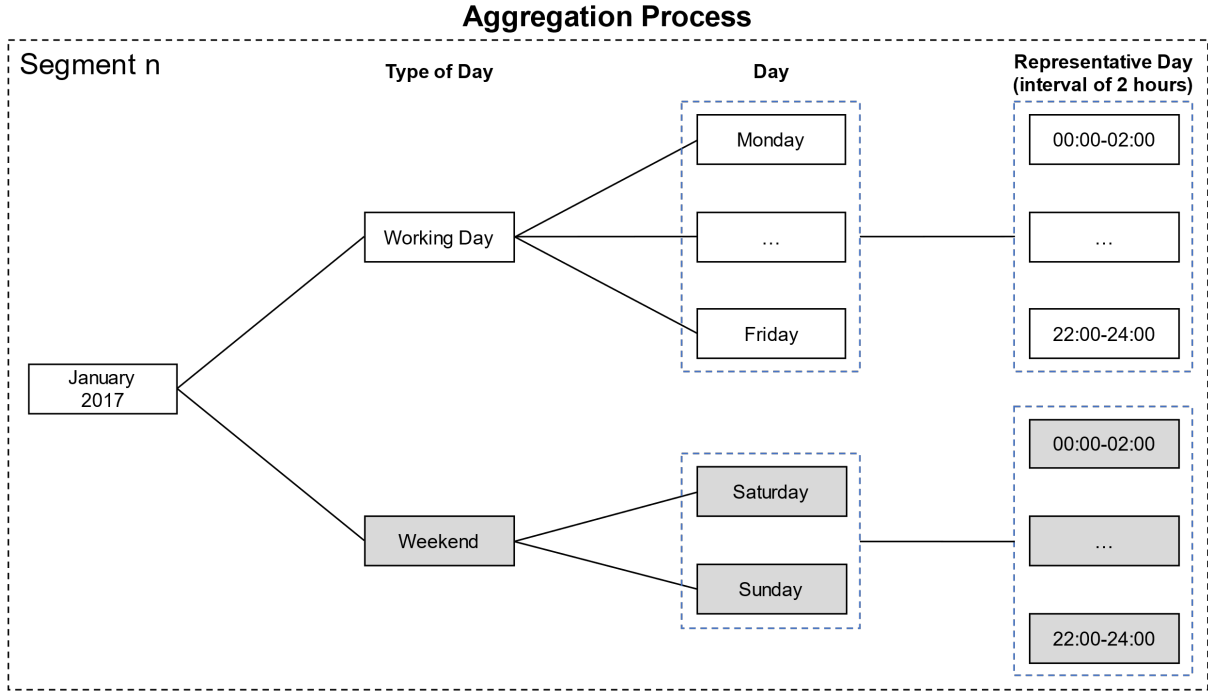


Figure 2-2. Aggregation process

3. Methodology

The following sections describe the proposed methodology for the optimization and calibration of microscopic traffic simulations by using traffic measurements from collected GPS-data. As mentioned above, the approach consists of two optimization loops, namely inter linear programming for a priori optimization of the traffic counts as well as an evolutionary algorithm for a posteriori optimization of counts and mainly speed distribution. Since each segment of the TomTom-dataset can be interpreted as detector loop in SUMO, we will call them “detectors” and edges containing such ones “detector edges” in the following explanations.

3.1 A priori optimization

The first step of the a priori optimization is very similar to the tool routesampler and starts with creating an initial set of routes for the considered network. This can be realized with the program randomtrips offered by SUMO and generating random routes. As a kind of pre-filter, only those of the generated routes are considered which include detector edges. With this initial start point the mathematical problem can be formulated as follows: How often does every route have to be chosen to match with the real vehicle counts of each detector edge?

To solve this problem, conventional integer linear programming with the following form will be adopted [4]:

$$\begin{aligned}
 & \min \mathbf{c}^T \mathbf{x} \\
 & \text{subject to:} \\
 & \mathbf{b}_l \leq \mathbf{A}\mathbf{x} \leq \mathbf{b}_u \\
 & \mathbf{x} \geq \mathbf{0} \\
 & x_i \in \mathbb{Z}
 \end{aligned} \tag{3-1}$$

In concrete terms, \mathbf{c} contains the coefficients of the object function. The vectors \mathbf{b}_l and \mathbf{b}_u represent the lower and upper bounds of the vehicle counts, whereby the lower limit will be

set with zeros and the upper limit with the recorded vehicle counts of each detector edge D_i ($i = 1, \dots, m$). The decision variables of x are constrained to be non-negative.

$$\mathbf{b}_u = \begin{bmatrix} \text{counts}, D_1 \\ \text{counts}, D_2 \\ \vdots \\ \text{counts}, D_{m-1} \\ \text{counts}, D_m \end{bmatrix} \quad (3-2)$$

Creating matrix A is a two-stage process: The first step includes setting up an auxiliary matrix $A' \in \mathbb{Z}^{m \times n}$ with all detector edges placed in the same order, i.e., $a'_{ij} = D_i$. Here, n is equivalent to the total number of initialized routes. Matrix A' is then transformed by running through each route and checking if the route contains a detector edge. If this is the case, the position of the corresponding detector edge in matrix A' is set to one, otherwise to zero. The result of the projection represents the binary matrix A .

$$A' = \begin{bmatrix} D_1 & D_1 & \dots & D_1 \\ D_2 & D_2 & \dots & D_2 \\ \vdots & \vdots & \vdots & \vdots \\ D_{m-1} & D_{m-1} & \dots & D_{m-1} \\ D_m & D_m & \dots & D_m \end{bmatrix} \Rightarrow A = [a_{ij}]_{m \times n} \text{ with } a_{ij} = \begin{cases} 1 & \text{if } a'_{ij} \in \text{Route } j, \\ 0 & \text{else.} \end{cases} \quad (3-3)$$

The resulting inequality of the integer linear programming problem can be written with following structure.

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{(m-1)1} & a_{(m-1)2} & \dots & a_{(m-1)n} \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \leq \begin{bmatrix} \text{counts}, D_1 \\ \text{counts}, D_2 \\ \vdots \\ \text{counts}, D_{m-1} \\ \text{counts}, D_m \end{bmatrix} \quad (3-4)$$

Route 1
Route n
Real vehicle counts

The objective of the optimization problem is defined as the maximization of the sum of the vehicles counts along all routes. Consequently, the corresponding minimization problem can be derived in a simple way.

$$\begin{aligned} \mathbf{c}^T &= \left[\sum_{i=1}^m a_{i1} \quad \sum_{i=1}^m a_{i2} \quad \dots \quad \sum_{i=1}^m a_{in} \right] \\ \max F(x) &= \sum_{i=1}^m a_{i1} \cdot x_1 + \dots + \sum_{i=1}^m a_{in} \cdot x_n = \sum_{j=1}^n \sum_{i=1}^m a_{ij} \cdot x_j \\ \min F(x) &= - \sum_{j=1}^n \sum_{i=1}^m a_{ij} \cdot x_j \end{aligned} \quad (3-5)$$

3.2 A posteriori optimization

As the result of the integer linear programming problem only represents the multipliers of each route to meet the vehicle counts before any simulation process, a second optimization loop is necessary making the calibration also of the recorded mean vehicle speed next to vehicle counts possible. This is realized by a combination of the ILP algorithm with an evolutionary algorithm where the first one serves as an initial start point for the optimization through an

evolutionary algorithm. Figure 3-1 shows the typical structure of an evolutionary algorithm with the main operators initialization, selection, recombination, mutation and reinsertion. These operators describe the characteristics of a simply structured evolutionary algorithm extended by the empirical data of TomTom and the elements of integer linear programming. An evolutionary algorithm starts with an initial population of parameter-sets and calculates the objective to optimize (in our case the deviation of vehicle counts and speed). A defined number of parameter-sets are stored and combined with each other. This describes the selection and recombination process. Prior to reinsertion into the initial quantity, the recombined parameter-sets are slightly modified by mutation around a specified range. The whole process is repeated until the termination criteria are reached. In our case, the parameter-set consists of the multipliers x for each route and the maximum allowed vehicle speed v of each detector edge. For further details of evolutionary algorithms in general, the authors refer to [5].

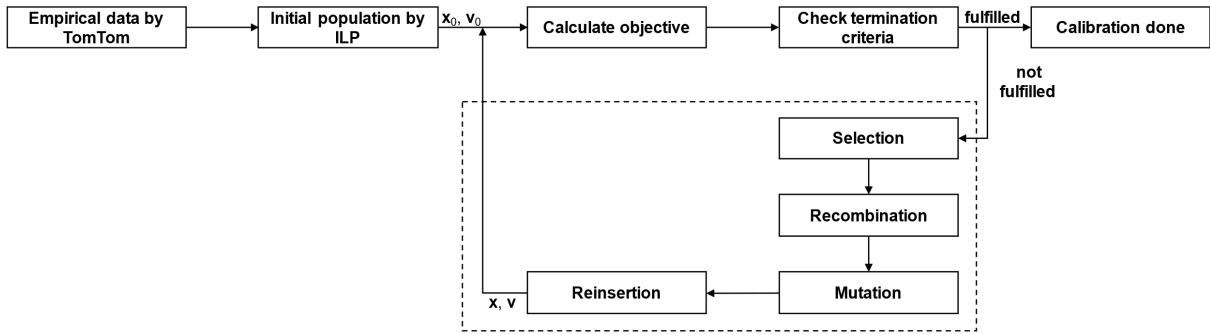


Figure 3-1. A posteriori optimization, following [5]

In order to not obtain only the same parameter-set of multipliers for the initial population, equation (3-5) is slightly changed by the variable $c \in \mathbb{R}$ modified for each element of the population randomly between one and two.

$$\min F(x) = - \sum_{j=1}^n \left(\sum_{i=1}^m a_{ij} \right)^c \cdot x_j \quad (3-6)$$

4. Simulation results

As a proof of concept, the proposed methods of optimization and calibration are tested with the described network model of the inner city of Friedrichshafen for one representative day of the available dataset and compared with the ready-to-use tools flowrouter and routesampler. For the routesampler and the ILP algorithm the same initial routes created by randomtrips are applied. Hereinafter, the results of the SUMO-tools, the ILP algorithm and the combined approach of ILP and evolutionary algorithm are presented. In general, SUMO-version 1.19.0 and the HiGHS algorithm [6] provided by the scipy library [7] for solving the ILP problem are used.

In the following, the Mean-Absolute-Error MAE is used as goodness-to-fit measure for the comparison between real and simulated traffic measurements. Figure 4-1 shows the pre- and post-simulation results of the vehicle counts for the tools flowrouter, routesampler and the ILP algorithm. The pre-simulation results for flowrouter and routesampler are calculated by determining the vehicle counts for each detector within one time interval through the corresponding route- and flow-files, respectively. The option --respect-zero is set for flowrouter meaning that detectors with no available data in one or more time intervals are also taken into account. As one can recognize, the designed ILP algorithm shows better results in terms of vehicle counts in comparison to the tools flowrouter and routesampler for every time interval of the selected day. The difference between the pre- and post-simulation results can be explained by a possible time overlap of one route starting in the time interval of creation but

ending in the subsequent one. As mentioned before, the ILP algorithm just focusses on a priori optimization of the vehicle counts and not of the mean speed. Nevertheless, the resulting values of the mean speed of each time interval will be depicted for the sake of completeness. For a better understanding of the results, the mean values of the recorded traffic data over all detector edges for every time interval are presented in addition.

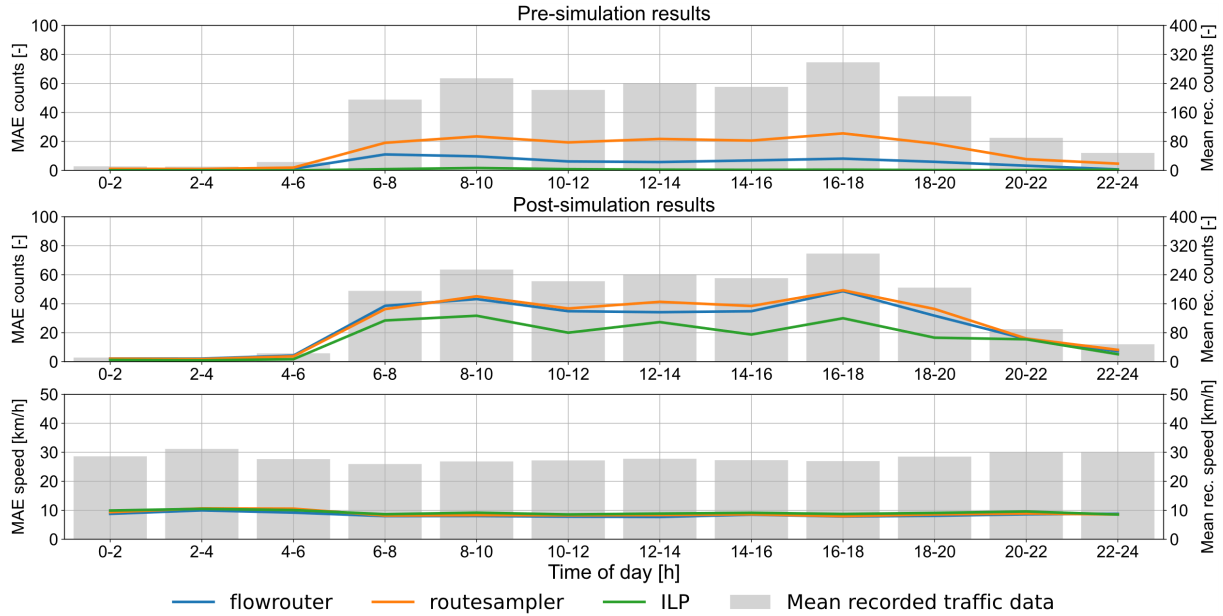


Figure 4-1. ILP optimization, compared to flowrouter and routesampler

The a posteriori optimization with the combination of ILP and evolutionary algorithm (EA) exemplarily will be applied for one single time interval of the day with sufficient traffic demand (6-8 a.m.). To consider the overlap with the preceding time intervals and to get the best comparison with the ILP algorithm, those intervals are just optimized by ILP. The simulation parameters are set according to table 4-1. Termination criteria are defined as a compromise between computation time and accuracy. As one obviously can see from figure 4-2, the results show the potential of the two-step optimization process improving the match with the mean speed and even the vehicle counts with respect to the ILP algorithm.

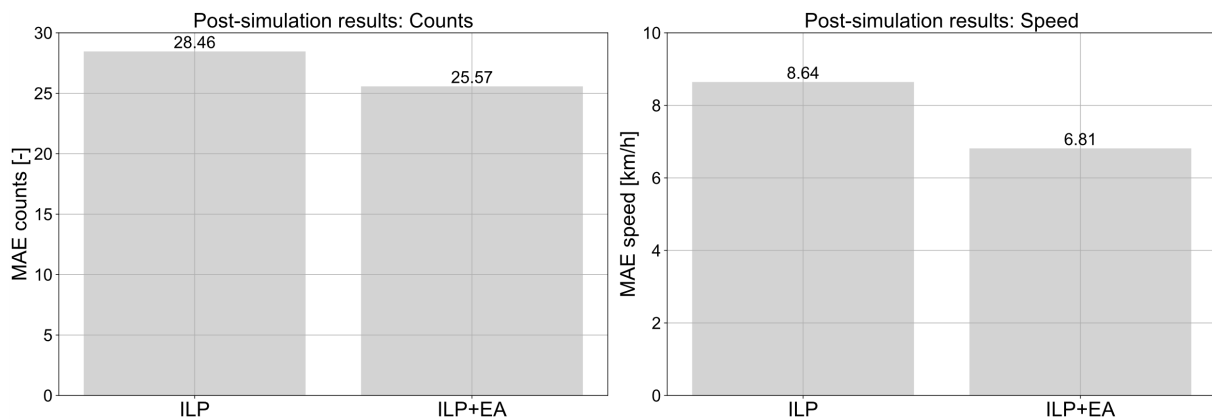


Figure 4-2. ILP+EA optimization

Table 4-1. Simulation parameters of a posteriori optimization

Parameters	
Changing parameter c of equation (3-6)	[1...2]
Number of initial parameter-sets/initial population	10
Maximum number of reinserted parameter-sets	5
Mutation range after recombination	$\pm 10\%$ for \mathbf{x} and \mathbf{v}
Termination criteria	MAE for count/speed < 10/4.5 km/h OR Iterations > 2000

5. Conclusion

In this paper, a robust method for the calibration of microscopic traffic simulations by using vehicle count and speed measurements from collected GPS-data is presented. The core of the calibration includes two optimization loops with an integer linear programming and evolutionary algorithm. In this respect, the application of the described evolutionary algorithm can be interpreted as an extension of the integer linear programming algorithm making a posteriori calibration of the mean speed values next to vehicle counts realizable.

The proposed method of the sole application of integer linear programming was described, implemented and tested in a subnetwork of Friedrichshafen. The deviation of vehicle counts was evaluated and compared with the SUMO tools flowrouter and routesampler. The presented method performed better than the aforementioned functions provided by SUMO although the tool routesampler follows a similar approach according to the developers' documentation. The usage of an evolutionary algorithm combined with the ILP approach exemplarily was demonstrated for one time interval of one representative day thus better approximating the recorded mean vehicle speed. Since the approach is generic, it also offers the possibility of large-scale traffic calibration.

For further analysis, the authors will enhance the approach of the combined optimization for a larger time range including further information of the collected dataset. Next to the modelling of travel demand, this method will also provide the possibility of scenario generation for AD/ADAS-Testing with the help of a microscopic traffic simulation.

Data availability statement

The underlying subnetwork of Friedrichshafen was extracted from freely accessible Open-StreetMap data. The employed TomTom-dataset was acquired commercially by ZF Friedrichshafen AG for internal utilization and for this reason is currently not freely accessible.

Author contributions

Christopher Stang – conceptualization, formal analysis, investigation, methodology, software, validation, visualization, writing-original draft, writing-review and editing

Klaus Bogenberger – supervision, writing-review and editing

Competing interests

The authors declare that they have no competing interests.

References

- [1] B. Ciuffo, V. Punzo, and M. Montanino, "The Calibration of Traffic Simulation Models: Report on the assessment of different Goodness of Fit measures and Optimization Algorithms", 2012, doi: <https://www.doi.org/10.2788/7975>. Accessed: February 6, 2024. [Online]. Available: <https://publications.jrc.ec.europa.eu/repository/handle/JRC68403>
- [2] DLR. "SUMO User Documentation". <https://sumo.dlr.de/docs/index.html> (accessed February 6, 2024).
- [3] OpenStreetMap. [Map section Friedrichshafen][Map]. Map data from OpenStreetMap. Open Database License ODbL (<https://opendatacommons.org/licenses/odbl/>). <https://www.openstreetmap.org/#map=15/47.6602/9.4736> (accessed February 13, 2024)
- [4] M. Conforti, G. Cornuéjols, and G. Zambelli, "Integer Programming", Cham, Switzerland: Springer International Publishing, 2014, doi: <https://www.doi.org/10.1007/978-3-319-11008-0>
- [5] H. Pohlheim, "Evolutionäre Algorithmen: Verfahren, Operatoren und Hinweise für die Praxis", Berlin, Germany: Springer-Verlag Berlin Heidelberg GmbH, 2000, doi: <https://www.doi.org/10.1007/978-3-642-57137-4>
- [6] Q. Huangfu and J.A.J. Hall, "Parallelizing the dual revised simplex method", in *Mathematical Programming Computation*, vol. 10, no. 1, 2018, pp. 119-142, doi: <https://www.doi.org/10.1007/s12532-017-0130-5>
- [7] P. Virtanen, R. Gommers, T. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. van der Walt, M. Brett, J. Wilson, K. Millman, N. Mayorov, A. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C. Harris, A. Archibald, A. Ribeiro, F. Pedregosa, P. Mulbregt, and SciPy 1.0 Contributors. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", in *Nature Methods*, vol. 17, no. 3, 2020, pp. 261-272, doi: <https://www.doi.org/10.1038/s41592-019-0686-2>