Simulation and optimization: a short review

Michel Bierlaire *

November 13, 2014

Report TRANSP-OR 141113
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

*École Polytechnique Fédérale de Lausanne (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Transport and Mobility Laboratory, Switzerland, michel.bierlaire@epfl.ch

Abstract

This review discusses some issues related to the use of simulation in transportation analysis. Potential pitfalls are identified and discussed. An overview of some methods relevant to the use of an advanced simulation tool in an optimization context is also provided.

1 A personal note

I have met Matthew Karlaftis for the first time when he was a PhD student, at the meeting of the Transportation Research Board in DC. He was the type of person that you notice immediately: full of energy, really smart, and always smiling. We continued to interact here and there as our career developed. When he invited me to be a plenary speaker at the International Conference on Engineering and Applied Sciences Optimization, I immediately accepted. I proposed him to talk about the role of simulation in transportation research in general, with a special focus on simulation-based optimization. His reaction was enthusiastic. Unfortunately, Matt did not attend the plenary lecture. The announcement of his death was made just after the plenary session, creating a shockwave of emotions and bewilderment. This article builds on the material of this lecture, and is dedicated to the memory of a great colleague and friend.

2 The pitfalls of simulation

The complexity of transport systems requires advanced decision support methodologies. It is the consequence of the interaction of many different elements, including transport infrastructure, travelers, goods, vehicles, information and communication technologies, to cite just a few. The analyst has only partial access to the exact interactions, and is faced with a great deal of uncertainty. An important source of uncertainty is related to the strong behavioral dimensions of transportation systems, as the joint choices of many individuals in terms of mode choice, departure time choice and route choice, among others, have a huge impact on the level of service of the system. The resulting congestion generate important economical and social costs (the cost of congestion has been reported to be more than $120 billion in the US in 2012 (Schrank et al., 2012), and €80 billion in Europe in 2013 (European Commission, 2013)), justifying the need for advanced decision support systems in order to mitigate the negative externalities of congestion.

Decision support systems rely on detailed mathematical models that describe the underlying reality. These models may be included in an optimization framework to identify the best configuration of the system at hand. This applies at the planning stage and at the operational level. The mathematical models are designed in such a way that (i) they capture the causal effects linking the many state variables of the system and (ii) they account for the uncertainty.

Unfortunately, it is easy to confound the two effects (causalities and uncertainty). The following example illustrates that potential confusion, using a simple experiment that can easily be organized in front of an audience.

2.1 A simple experiment

The experiment is performed on a map of a city center, with the claimed objective to investigate the impact of safety measures. We assume that the number of accidents on each road on the map is a random variable that can be represented by throwing two dice. Therefore, we simulate the number of accidents for the reference year by throwing two dice for each road. For instance, we obtain the following values for 20 roads: 4, 5, **9**, **9**, **10**, 8, **10**, 8, 4, 6, 3, 4, 5, 6, 4, 8, 5, 7, 3, 4. Now, you select the four most dangerous roads, that is roads 3, 4, 5 and 7 (in bold above) in our example. On each of these roads, you place a figurine representing a speed limit enforcement system. And you claim that you will now evaluate the impact of these measures by comparing the number of accidents before and after the installation of the figurines. For each of the dangerous roads, you throw again the dice to generate the number of accidents after the speed enforcement. And you obtain 6, 7, 4 and 7, say. You conclude that your speed enforcement system has reduced the number of accidents from 38 to 24, that is an impressive reduction of about 37%.

The fallacy is obvious: in this experiment, there is absolutely no causal effect between the presence or absence of the figurine and the second set of accident statistics. There are actually two wrong conclusions. First, the identification of the most dangerous roads is based on one random instance of the accidents scenarios. It is clear from the set up of the experiment that each road is equally dangerous, in average. Second, the calculation of the decrease of the number of accidents is also based on a single instance. It is interesting to note that, with such a setup, there is more than 98% chance to "prove" that the number of accidents decreases. The correct way to perform the experiments would have been to generate many instances before hand to identify the most dangerous roads. The conclusion would have been that a number of about 7 accidents occur on each road in average. If the same analysis would be performed after the figurine have been installed, the conclusion would also have been close to 7. It would then be possible to perform statistical tests to check if the two values are significantly different or if the difference is only due to the stochasticity of the system.

There are two lessons that can be learned from this experiment. First, if the experiment is transposed in real life, it is easy to imagine similar situations where causal effects have allegedly been proved although they did not exist. Contrarily to simulation environment, it is not always possible to repeat the same experiment a sufficient number of times in real life to capture the stochasticity of the system. Second, with respect to the simulated experiment itself, it is clearly insufficient to draw conclusions based only on one realization of a stochastic system, or a stochastic model.

2.2 Simulation in a nutshell

A simulation tool is meant to capture the complexity of a system, as well as its stochasticity. The model of the system, that is its mathematical formulation that captures its simplified representation, is composed of the following elements, as illustrated by Figure 1.

- The state variables x characterize the configuration of the system at a given point in time (static models) or characterize this configuration at several points in time (dynamic models).
- The external input variable y characterize elements that are outside the system that may influence its configuration.
- The control variables u characterize the actions that can be taken by the engineer or the policy maker to modify the configuration of the system.

- The indicators z that characterize the level of performance of the system.

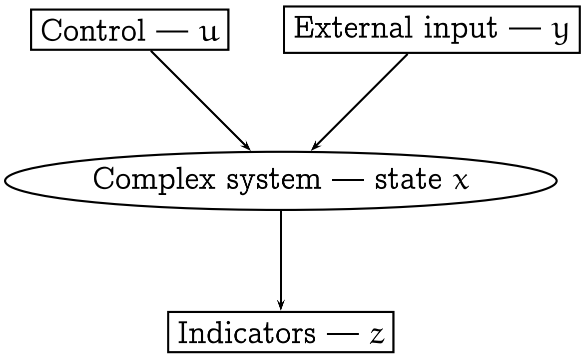


Figure 1: Elements of a simulation

As an example of a complex system, consider the network of streets in a city center designed to carry the vehicular traffic. The state variables x characterize the current state of the traffic using quantities such as flows, densities, or speeds for each time interval of the time horizon under interest. The external input variables y characterize the travel demand, that can be represented for instance by time-dependent origin-destination matrices. The control variables u can be, for instance, the configuration of the traffic lights at each intersection of the city center. And the performance indicator z can be the total travel time, or the total throughput on the network during the time horizon under interest.

The relationship between these variables is represented by a mathematical model, that can be explicit or implicit. An explicit model writes

$$z = h(x, y, u), \tag{1}$$

where the values of the performance indicators are derived from the values of the other variables. This model explicitly captures the causal effects occurring in the system. Due to the high complexity of the system, it is common to decompose the function h into a set of models, each of them capturing different aspects of the system, and to model the interaction among these subsystems. In this case, h may not necessarily be represented by a closed form mathematical equation, but by a possibly sophisticated piece of software.

An implicit model writes

$$h(x, y, u, z) = 0, \tag{2}$$

where the computation of the value of the performance indicators involves the solution of systems of equations or fixed point problems. It typically happens when several subsystems interact with each other in a circular way, and the analyst is interested in some sort of equilibrium. In a traffic assignment context, the travel time on the network is a function of the flow of vehicles on each link, which itself is a function of the decisions of travelers in terms of destination, departure time and route. And these decisions are based on the expected travel time itself. Using the notations introduced above, this can be written as

$$z = h'(x, y, u, z) \tag{3}$$

where z would be the travel time in this example, or, equivalently

$$h(x, y, u, z) = z - h'(x, y, u, z) = 0. \tag{4}$$

Implicit models usually involve more computation efforts, but achieve the same objective than explicit models: to derive indicators of performance of a system in a given configuration. Therefore, in the remaining of this review, we focus mainly on the explicit formulation.

As said above, the role of the h function is to capture the complexity of the model through the description of the causal relationships among variables. In order to account for uncertainty, these variables must be considered as random variables. To emphasize that, we denote them by capital letters. The explicit simulation can then be written as

$$Z = h(X, Y, U) + \varepsilon_Z, \tag{5}$$

where ε_Z captures the modeling errors embedded in the mathematical representation h. The key question for explicit simulation is: given the distribution of the random variables X, Y, U and ε_Z, what is the distribution of Z?

Except in rare cases, that are usually only of academic interest, the function h is too complex to derive the distribution of Z from the mathematical properties of random variables. Instead, we rely on simulation. It consists in drawing a sample[1] of R instances of X, Y, U and ε_Z, denoted by x^r, y^r, u^r and ε_Z^r. For each $r = 1, \dots, R$, an instance z^r of Z is calculated using the h function:

$$z^r = h(x^r, y^r, u^r) + \varepsilon_Z^r. \quad (6)$$

If R is sufficiently large, the empirical distribution of the z^r is a good approximation of the distribution of Z. In particular, the empirical CDF

$$F_e(x) = \frac{1}{R}\,\#\{z^r \leq x\}, \quad (7)$$

where $\#\{z^r \leq x\}$ is the number of instances z^r that are less or equal to x, is a good approximation of the CDF of Z. More precisely, if $F(x)$ is the true CDF of Z, than for any $x \in \mathbb{R}$,

$$E[F_e(x)] = F(x) \quad (8)$$

and

$$\mathrm{var}(F_e(x)) = \frac{1}{R}F(x)(1 - F(x)). \quad (9)$$

Any statistic associated with Z, such as the mean, the variance, the mode, or any quantile, can be approximated by the calculation of the corresponding statistic on the sample of z^r. For example, the mean of the random variable can be approximated as follows:

$$E(Z) = \int_{z=-\infty}^{+\infty} z f(z) dz \approx \frac{1}{R}\sum_{r=1}^{R} z^r. \quad (10)$$

It is of primary importance that the number of draws R is sufficiently large for the results of the simulation to be meaningful. As we illustrated with the simple experiment described in Section 2.1, conclusions based on low values of R may be misleading as the empirical CDF (7) is a poor approximation of the real one.

There is a trend in transportation research to make the h function in (5) more and more complex, involving pieces of software highly demanding in

[1] It is not necessarily an easy task. We refer the reader to Ross (2006) for a detailed description of the methodology.

terms of computational time. It is therefore often not feasible to generate a number R of draws which is sufficiently high to obtain a decent approximation of the real distribution of Z. In this case, the quantities calculated from the few number of draws are just random numbers that do not reflect anything meaningful about the system. The exact number of required draws R is problem dependent, and must be based on the estimation of the simulation error (see Ross, 2006, chapter 7).

They are several techniques to reduce the variance of the output of the simulator, such as antithetic draws (Hammersley and Morton, 1956), or control variates methods (Iglehart and Lewis, 1979). The achieved variance reduction can sometimes be dramatic, to that the savings on computational efforts are substantial. We refer again the interest reader to Ross (2006) for a detailed coverage of the topic.

2.3 The flaw of averages

The concept of "flaw of average" has been introduced by Savage (2012) to emphasize that "plans based on average assumptions are wrong on average". Among many analogies, he compares the use of averages to that of a drunk walking down the highway, trying to follow the yellow line separating the two lanes. The state of the drunk at his average position (that is, right on the yellow line) is "alive". But the average state of the drunk is "dead", as illustrated in Figure 2.

In the context of simulation, the expected value of a nonlinear function of random variables is not equal to the value of the nonlinear function evaluated at the average value of its arguments, that is

$$E[Z] = E[h(X, Y, U) + \varepsilon_z] \neq h(E[X], E[Y], E[U]) + E[\varepsilon_z]. \tag{11}$$

The only exception is when h is a linear function of its inputs, which is relatively rare when modeling complex transportation systems.

Another issue is the over-emphasis that is put on the mean for the analysis of complex systems. Even if the mean is correctly calculated using (10), it is not necessarily a useful or sufficient quantity to consider. Focusing only on the mean is not a flaw from a methodological point of view (like the previous one), but it may have considerable impacts on operational decisions. Consider the following simple example of a street with a capacity of 2000 vehicles per hour. It is equipped with traffic lights at a

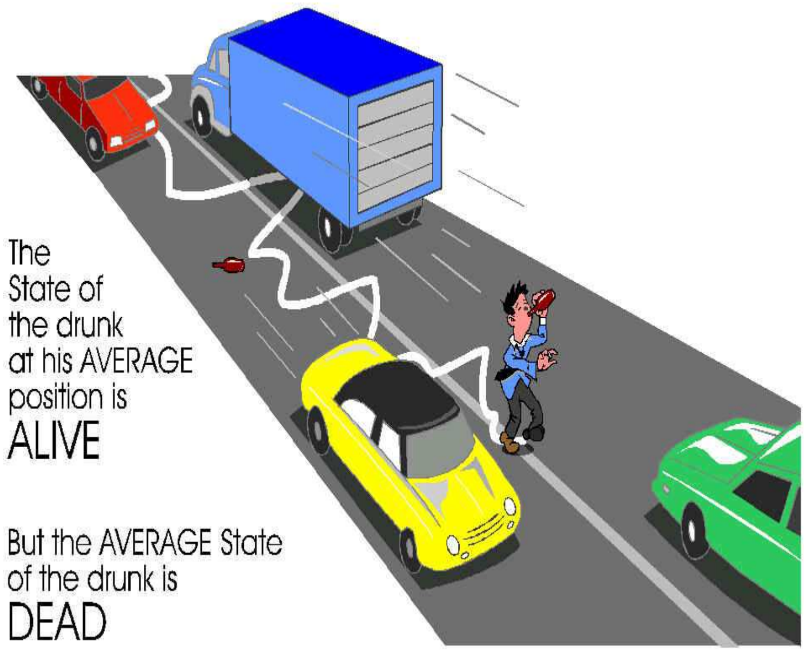


Figure 2: Illustration of the flow of average by Savage (2012)

pedestrian crosswalk, configured with 30 seconds green followed by 30 seconds red. There is a constant arrival rate of 2000 vehicles per hour during 30 minutes. Due to the possible presence of pedestrians running the red light, the capacity of the intersection drops from 100% to 30%, with 80% probability. We are interested in the time spent by travelers in the system. The results of the simulation are reported in Figure 3, where histogram represents the distribution of the variable Z under analysis in this context.

It appears that the indicator of interest (here the time spent in the system) follows a multi modal distribution. The mode (that is, the value with the highest probability to occur) is a quantity of interest to the analyst. And the existence of multiple modes is particularly important to identify, as each mode may be associated with a specific regime of the system. Looking only at the mean may be completely misleading, as the mean may be associated with a low probability of occurrence, like in this example where it is equal to 686.

When the indicator of interest follows a unimodal symmetric distribution such as the normal distribution, the mode coincides with the mean,

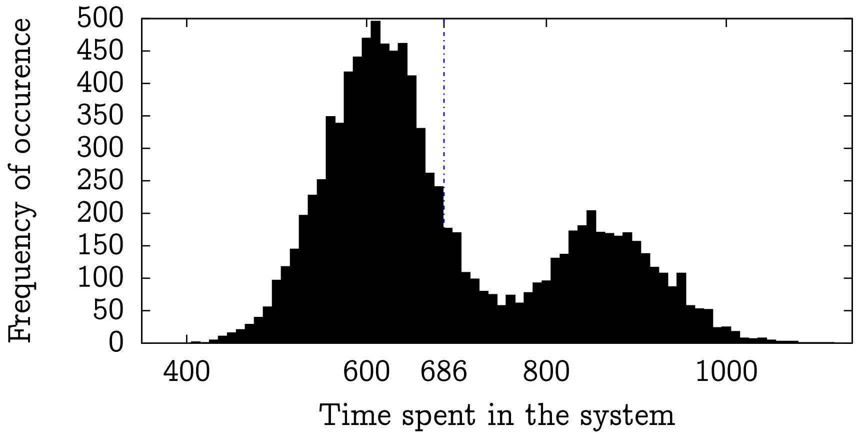


Figure 3: Distribution of time spent in the intersection

which is easier to calculate. It is just important to keep in mind that complex simulators may produce distributions that are multi modal. It is important to investigate the full distribution before deciding what feature of the distribution is the most useful for the application at hand. The use of standard histograms, like in the example above, are strongly recommended.

In addition to the mode(s), quantities that are particularly useful for practical applications are the quantiles. They provide the value x such that the indicator of interest is below x with probability p. They allow to obtain measures of performance that exclude the extreme cases that have a low probability to occur.

3 Simulation-based optimization

In order to use the model (5) in an optimization context, we assume that the control variables U are deterministic and that the random variables X, Y and ε_Z are such that it is possible to generate draws from their distribution. Let $S(Z)$ be the statistic of Z that must be optimized, that is the mean, the variance, the mode, or any quantile, for instance. We denote $S^R_e(Z)$ the approximation of this statistic obtained by simulation using R draws. The e index emphasize that we use the empirical CDF to calculate that statistic. The optimization problem under interest can be written as:

$$\min_u f(u) = S(Z) = S(h(X, Y, u) + \varepsilon_Z). \tag{12}$$

Relying on simulation, the problem simplifies to

$$\min_u f_R(u) = S_e^R(z^1, \ldots, z^R) = S_e^R(h(x^1, y^1, u) + \varepsilon_Z^1, \ldots, h(x^R, y^R, u) + \varepsilon_Z^R). \tag{13}$$

For instance, if the statistic of interest is the mean, we have

$$\min_u f_R(u) = S_e^R(z^1, \ldots, z^R) = \frac{1}{R} \sum_{r=1}^{R} h(x^r, y^r, u) + \varepsilon_Z^r. \tag{14}$$

Note that, in many practical applications, the variables u must be restricted to meaningful values only, so that the optimization problem is subject to some constraints. There are several techniques to transform a constrained optimization problem into an unconstrained optimization problem. One of them consists in adding to the objective function a term that penalizes any value of u that violates the constraint. The discussion of these methods is out of the scope of this review, where we focus on the unconstrained optimization problem.

For instance, consider that we are using a traffic simulation tool such as MITSIM (Yang and Koutsopoulos, 1996), MatSIM (Raney et al., 2003), VISSIM (Fellendorf and Vortisch, 2010) or AIMSUN (Barceló and Casas, 2005), to cite just a few. We need to calibrate the parameters of the simulator using observed link flows (Balakrishna et al., 2007). In this case, the variables X represent the state of traffic, the variables Y represents the origin-destination matrix and the observed link flows, the vector u represents the parameters to be calibrated. the function h is the traffic simulator itself, and Z is the total squared difference between modeled and observed link flows. Finally, $S_e^R(Z)$ is the mean squared error across R runs of the simulator.

Once the simulator is calibrated, it can be used to optimize the settings of the traffic lights. In this case, X is again the state of traffic, Y is the origin-destination matrix, u represents the traffic light configuration, h is again the traffic simulator, Z is the total travel time spent in the system by the travelers. We may be interested to minimize its expected value (Osorio and Bierlaire, 2013), or, if we are interested in reliability, its variance (Chen et al., 2013). The statistic $S_e^R(Z)$ is defined accordingly.

The optimization problem (13) is particularly difficult to solve. First, calculating the value of the objective function for a given value of u usually requires a high amount of computation time. Second, the function f_R may not be differentiable in u, either because h itself is not differentiable in u, or the statistic S_e^R is not (it is the case namely with quantiles). And even when $f_R(u)$ is differentiable in u, the derivatives are not available when the calculation of h involves running a piece of software.

We review now various methods that can be considered to solve, at least partially, the optimization problem. We organize them into three categories. The "black box" algorithms (Section 4) that do not attempt to exploit the exact structure of f. They consider it as a black box that is able to provide $f(u)$ when u is provided. The "noise reduction" algorithms (Section 5) try to decrease the amount of computation time by varying the value of R across iterations. The "open box" algorithms aim at exploiting the structure of the system that is modeled by h.

4 Black box algorithms

The algorithm that is probably the most used by practitioners relying on simulation in transportation related decision-making is the scenario-based optimization method. It consists in generating a list of N scenarios $u_1, \ldots, u_N$. For each scenario i, the value $f(u_i)$ of the objective function is calculated. The scenario with the best value of the objective function is selected as the solution.

This method is simple, and the total computational effort can be controlled. It is usually good practice to start running a few scenarios, and to generate additional scenarios based on the insights gained by the results obtained. Unfortunately, this method does not perform a systematic investigation of the solution space, and relies entirely on the creativity of the analyst.

4.1 Algorithmic differentiation

A great deal of optimization algorithms for nonlinear problems rely on the derivatives of the objective function or, when it is not differentiable, on a sub-differential or on directional derivatives (Bertsekas, 1999, Nocedal and Wright, 2006, Bonnans et al., 2006). The difficulty in the context of simulation is that the function h and, sometimes, the calculation of the statistic S_e^R in (13) rely on a piece of software, and not on an analytical formula. The methodology to calculate the derivatives of a piece of software is called *algorithmic differentiation* or *automatic differentiation*. The main idea is that any implementation of the h function is a composition of fundamental building blocks such as arithmetic operations, exponentials, logarithms, trigonometric functions, and so on. Calculating the derivatives of h can therefore be seen as an appropriate application of the chain rule in differentiation. There are many challenges associated with that task, that have generated a large body of literature (see Griewank and Walther, 2008, Naumann, 2012 for recent reviews). Several pieces of software are available to download from the internet (see, for instance, Hogan, 2014). Obviously, it is necessary to have access to the source code of the simulation program in order to apply this methodology. It has proved useful in various contexts such as ocean modeling (Rückelt et al., 2010), computer vision (Pock et al., 2007) or fluid mechanics (Mohammadi and Pironneau, 2004). To the best of our knowledge, it has not yet been applied to the analysis of transportation systems.

4.2 Derivative free optimization

The motivation to use derivatives in nonlinear optimization is that they provide a good local model of the objective function, that can be linear (linesearch methods) or quadratic (trust region methods). Derivative free optimization aims also at developing a local model of the objective function.

Instead of using derivatives, the model interpolates the objective function at carefully selected points (see Conn et al., 2009 for an introduction). Many variants have been proposed, based on Lagrange polynomials (Powell, 2002), radial basis functions (Oeuvray and Bierlaire, 2009), or kriging (Sankaran et al., 2010). The derivative-based convergence theory for nonlinear optimization can be extended to derivative free methods (Conn et al., 1997), providing a convenient theoretical framework. From an operational point of view, the main challenge is to deal with the numerical issues associated with the interpolation problem.

Another family of derivative-free optimization algorithms is sometimes referred to as *direct search* or *pattern search* methods. The local model of the objective function is based on a geometric object (usually a simplex). The values of the function calculated at the edges of the object are used to modify its geometry in order to explore the shape of the objective function. The method proposed by Nelder and Mead (1965) is used a lot in practice, namely because it is implemented in Matlab. Unfortunately, it is considered as a heuristic as it may fail to converge to stationary points (McKinnon, 1998). Modern pattern search methods benefit from a formal convergence theory (see Torczon, 1997).

In general, the size of the problems that can be solved by derivative-free methods is low (Moré and Wild, 2009 present benchmark instances of dimension up to 12). A recent review by Rios and Sahinidis (2013) compares 22 state-of-the-art software implementing derivative-free optimization algorithms.

Therefore, these methods have to be adapted to the specific context of transportation applications, where the number of variables involved in the optimization problem is usually higher.

5 Noise reduction

The "noise" refers to the variability of the value of the objective function (13). Indeed, for a given value of u, the value of f varies if different set of draws are used in the simulation. This variation vanishes as R goes to infinity. Unfortunately, in practice, it is common that the computational burden of the simulation software precludes the use of values of R that are large enough to ignore the presence of the noise.

Interpolation methods used in derivative-free optimization algorithms mentioned in the previous section are particularly sensitive to the presence of noise. This is illustrated on a simple example, where we interpolate three points by a polynomial. In Figure 4, the "true" objective function (that is, computed without simulation) is used to characterize the interpolation points and to define the polynomial. In Figure 5, the noisy objective function (that is, estimated using simulation with a value of R relatively low) is used to characterize the interpolation points. The resulting polynomial is completely different from the previous one, and can actually be misleading if we use it to guess the general trend of the function. In this case, the function is concave, but the polynomial model interpolated on the noisy function is convex.

To circumvent this problem, Bierlaire and Crittin (2006) have proposed a method building local models of the objective function f using also sampled values of f. But in order to avoid the numerical issues associated

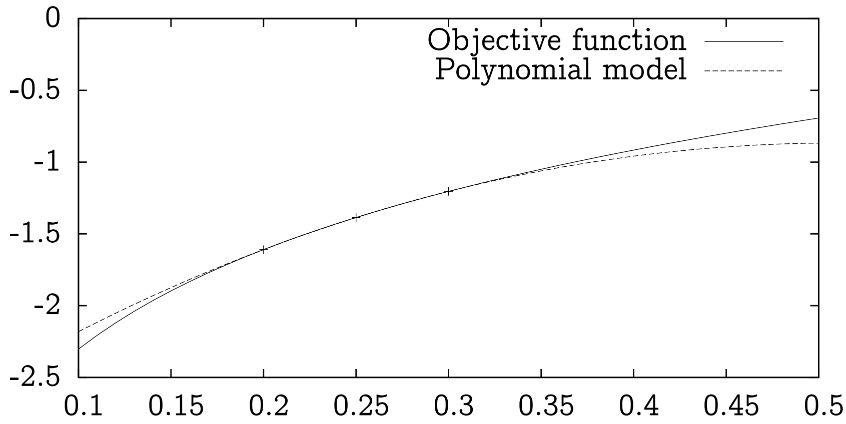


Figure 4: Polynomial interpolation of an objective function

with the interpolation-based algorithms, they calibrate this model through a least-squares fitting procedure. It has the advantage that the accuracy of the model can be improved using more sampled value of the objective function. In Figure 6, two additional values have been sampled and used to fit the polynomial model, that provides a better approximation of the general trend of the function (compared to the one in Figure 5), although it does not exactly match it at the sample points.

Using this method, and appropriate linear algebra to solve the least-squares problem, they are able to solve the consistent anticipatory route guidance problem on instances with more than 120000 variables. This problem is an implicit model (4) involving a simulation of the Irvine, Ca. network (618 links, 296 nodes, 627 OD pairs) from 4am to 8am, using a guidance horizon of 45-minute interval with update intervals of one minute. The simulation is performed by DynaMIT (Ben-Akiva et al., 2002).

Another method to reduce the noise induced by the high computational costs associated with simulation has been proposed by Bastin et al. (2006). The idea is to modify the value of R during the iterations of the optimization algorithm. Indeed, during early iterations, the algorithm does not necessarily need an accurate approximation of the objective function, and a relatively low value of R is sufficient. When the algorithm reaches a region close to the optimum, more precision is needed and R is increased. Bastin et al. (2006) provide a rigorous mechanism to adapt the value of R in the

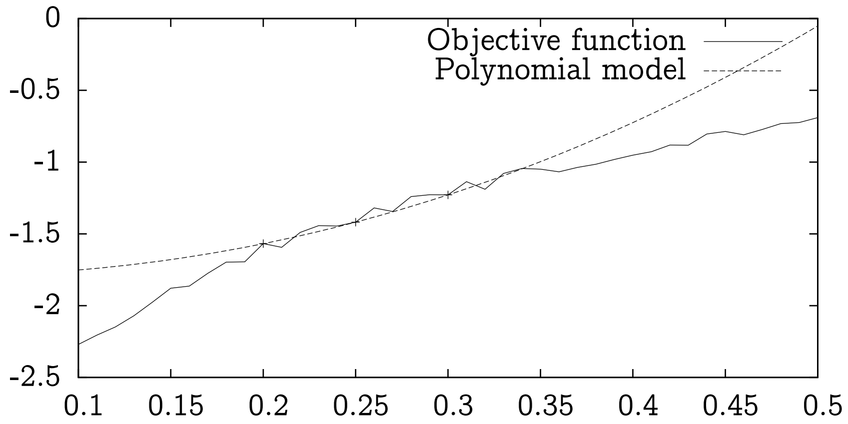


Figure 5: Polynomial interpolation of a noisy objective function

context of a trust region algorithm.

6 Open box algorithms

The methods introduced so far do not exploit the structure of the underlying problem. The function f is used as a black box that provides the value of the objective function and, sometimes, its derivatives. An analytical model of the function is then built based on this information. This model is often polynomial, inspired by Taylor's theorem that establishes that any nonlinear function can be approximated by a polynomial, and that provides a theoretical characterization of the approximation error. In the context of simulation-based optimization, such models are called *functional models*.

The philosophy of "open box" algorithms is to follow engineering practice, that suggests to use a *physical model* to perform optimization (Barthelemy and Haftka, 1993). However, if the analyst desires to rely on a simulation-based model, it is often because the tractable and analytical physical models do not provide the level of details required by the underlying application. Osorio and Bierlaire (2013) propose to combine the physical model and the functional model in order to benefit from the advantages of both: the functional model provides the convergence theory, that mathematically drives the algorithm, while the physical model provides an approximation that contains the relevant features of the system of interest, in the context of

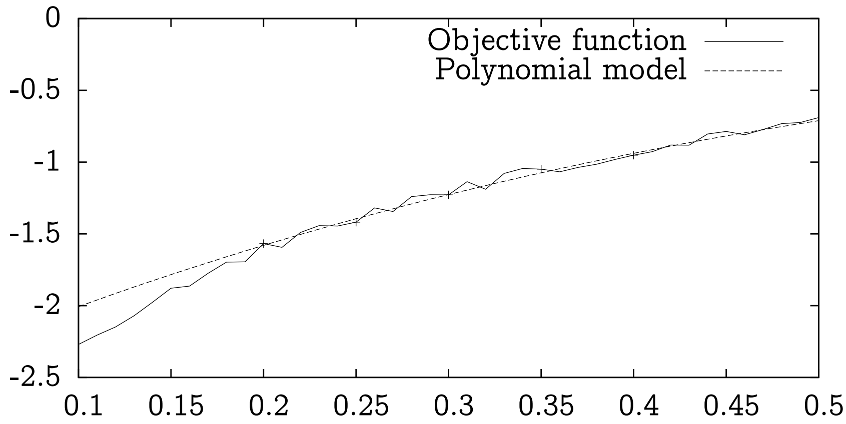


Figure 6: Least-square fitting of a noisy objective function

the application. They build a *metamodel* defined as

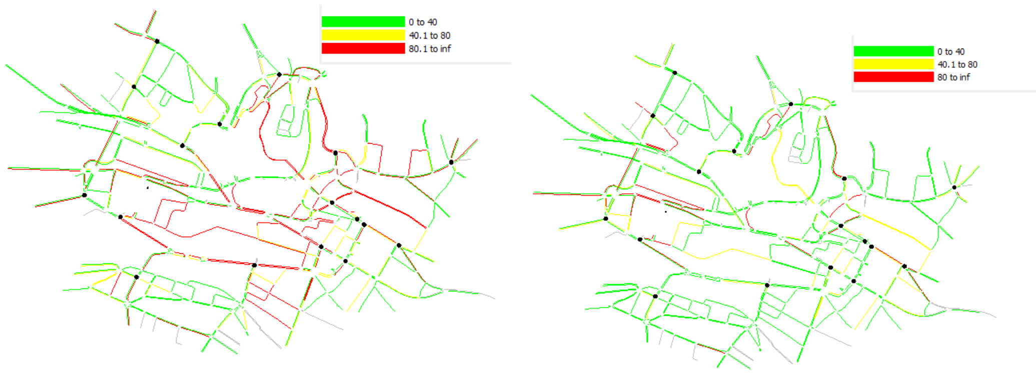
$$m(u, x; \omega, \beta, q) = \omega T(u, x, q) + \phi(u, \alpha) \tag{15}$$

where

- $T(\cdot)$ is the physical model,
- $\phi(\cdot)$ is the functional model,
- u is the vector of control variables,
- x is the vector of state variables,
- q is a vector of parameters of the physical model,
- α is a vector of parameters of the functional model,
- ω captures the relative importance of the two components in the metamodel.

The metamodel is then embedded in a trust-region framework (Conn et al., 2000), so that the algorithm exhibits the global convergence properties of this family of methods. They illustrate the concept on a simulation-based optimization of traffic lights in the city of Lausanne. The physical model is an analytical queueing model (Osorio and Bierlaire, 2009). The analytical information provided by the queueing network model allows to use less runs of the simulation package to identify good solutions of the simulation-based optimization problem.

Osorio and Chong (forthcoming) successfully apply the metamodel approach to optimize the traffic lights on a larger instance of the problem for the city of Lausanne (network with 603 links, 231 intersections, 17 controlled intersections, queueing model with 902 queues), and for an area of Manhattan, New-York city (Osorio, Chen, Marsico, Talas, Gao and Zhang, 2014).



(a) Initial signal plan

(b) Optimized signal plan

Figure 7: Average link travel times. The averages (in seconds) are taken over 50 simulation replications (Osorio and Chong, forthcoming)

The metamodel approach has been successfully applied to optimize vehicle-specific performance metrics, such as fuel consumption (Osorio and Nanduri, ta) and emissions (Osorio and Nanduri, 2013). Osorio, Flötteröd and Zhang (2014) use the approach for the calibration of the parameters of a simulation model.

As an illustration of the discussion in Section 2.3 about the need to consider other indicators than the mean, Chen et al. (2013) addresses a travel time reliable signal control problem, where the objective is to reduce time variability by minimizing the standard deviation of travel time. They derive the variance of the analytical model, and apply the methodology on the Lausanne network. They generate signal plans with low expected total travel time and the low standard deviation of total link travel time. These signal plans also have low across-replication variability of the travel time standard deviation.

We refer the reader to Barton and Meckesheimer (2006) and Søndergaard (2003) for more discussions about other metamodel approaches.

7 Conclusion

Simulation tools are pervasive in transportation systems analysis. Their flexibility allows to investigate many aspects relevant for both research and applications. This comes at the cost of complexity.

In this review, we have emphasized two major issues related to the use of simulator. The first is the danger to misuse the tool, when the number of draws is not sufficient, or when the considered statistic is not appropriate. The second is the difficulty to embed a complex simulation tool within an optimization framework.

Simulation must be seen as the detailed analysis of the distribution of a complex random variable. It provides the empirical cumulative distribution function, which is asymptotically converging to the true CDF. Therefore, if the number of draws is sufficiently large, any relevant statistic about the true distribution can be derived from the empirical CDF.

We have provided a short overview of simulation-based optimization. We have made the difference between “black-box” algorithms, that do not need information about the structure of the problem being simulated, and “open-box” algorithms, that exploit this structure in order to make the best of the computationally expensive data that are generated by the simulation. We have also presented “noise reduction” methods, that explicitly exploit in the optimization framework the stochastic nature of the output of the simulator, and try to make advantage out of it.

Most of the techniques presented in this note are not used a lot in the transportation literature, and even less in practice. We hope that the pointers to the literature provided in this review will help the researchers and the practitioners to make the best of our the complex simulation tools that they design and use.

References

Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H. N. and Wen, Y. (2007). Calibration of microscopic traffic simulation models:

Methods and application, *Transportation Research Record: Journal of the Transportation Research Board* **1999**(1): 198–207.

Barceló, J. and Casas, J. (2005). Dynamic network simulation with aimsun, *in* R. Kitamura and M. Kuwahara (eds), *Simulation approaches in transportation analysis*, Vol. 31 of *Operations Research/Computer Science Interfaces Series*, Springer, pp. 57–98.

Barthelemy, J.-F. and Haftka, R. T. (1993). Approximation concepts for optimum structural design—a review, *Structural optimization* **5**(3): 129–144.

Barton, R. R. and Meckesheimer, M. (2006). Metamodel-based simulation optimization, *Handbooks in operations research and management science* **13**: 535–574.

Bastin, F., Cirillo, C. and Toint, P. L. (2006). An adaptive monte carlo algorithm for computing mixed logit estimators, *Computational Management Science* **3**(1): 55–79.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. N. and Mishalani, R. (2002). Real time simulation of traffic demand-supply interactions within dynamit, *Transportation and network analysis: current trends*, Springer, pp. 19–36.

Bertsekas, D. (1999). *Nonlinear programming*, Athena Scientific optimization and computation series, Athena Scientific.
URL: *http://books.google.ch/books?id=TgMpAQAAMAAJ*

Bierlaire, M. and Crittin, F. (2006). Solving noisy large scale fixed point problems and systems of nonlinear equations, *Transportation Science* **40**(1): 44–63.

Bonnans, J., Gilbert, J., Lemarechal, C. and Sagastizábal, C. (2006). *Numerical Optimization: Theoretical and Practical Aspects*, Universitext, Springer.

Chen, X., Osorio, C. and Santos, B. (2013). Travel time reliability in signal control problem: Simulation-based optimization approach, *Proceedings of the Transportation Research Board (TRB) Annual Meeting January 13-17, 2013.*

Conn, A. R., Gould, N. I. M. and Toint, P. (2000). *Trust region methods*, MPS–SIAM Series on Optimization, SIAM.

Conn, A. R., Scheinberg, K. and Toint, P. L. (1997). On the convergence of derivative-free methods for unconstrained optimization, *Approximation theory and optimization: tributes to MJD Powell* pp. 83–108.

Conn, A. R., Scheinberg, K. and Vicente, L. N. (2009). *Introduction to derivative-free optimization*, Vol. 8 of *MPS-SIAM series on optimization*, Siam.

European Commission (2013). Together towards competitive and resource-efficient urban mobility. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions.

Fellendorf, M. and Vortisch, P. (2010). Microscopic traffic flow simulator vissim, *in* J. Barcelo (ed.), *Fundamentals of Traffic Simulation*, International Series in Operations Research and Management Science, Springer, pp. 63–93.

Griewank, A. and Walther, A. (2008). *Evaluating derivatives: principles and techniques of algorithmic differentiation*, Siam.

Hammersley, J. M. and Morton, K. W. (1956). A new monte carlo technique: antithetic variates, *Mathematical Proceedings of the Cambridge Philosophical Society* **52**: 449–475.

Hogan, R. J. (2014). Fast reverse-mode automatic differentiation using expression templates in c++, *ACM Trans. Math. Softw.* **40**(26): 1–16.

Iglehart, D. L. and Lewis, P. A. W. (1979). Regenerative simulation with internal controls, *J. Assoc. Comput. Mach.* **26**: 271–282.

McKinnon, K. I. (1998). Convergence of the nelder–mead simplex method to a nonstationary point, *SIAM Journal on Optimization* **9**(1): 148–158.

Mohammadi, B. and Pironneau, O. (2004). Shape optimization in fluid mechanics, *Annu. Rev. Fluid Mech.* **36**: 255–279.

Moré, J. J. and Wild, S. M. (2009). Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization* **20**(1): 172–191.

Naumann, U. (2012). *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*, number 24 in *Software, Environments, and Tools*, SIAM, Philadelphia, PA.
URL: *http://www.ec-securehost.com/SIAM/SE24.html*

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization, *The computer journal* **7**(4): 308–313.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer.
URL: *http://books.google.ch/books?id=VbHYoSyelFcC*

Oeuvray, R. and Bierlaire, M. (2009). BOOSTERS: a derivative-free algorithm based on radial basis functions, *International Journal of Modelling and Simulation* **29**(1): 26–36.

Osorio, C. and Bierlaire, M. (2009). An analytic finite capacity queueing network model capturing the propagation of congestion and blocking, *European Journal of Operational Research* **196**(3): 996–1007.

Osorio, C. and Bierlaire, M. (2013). A simulation-based optimization framework for urban traffic control, *Operations Research* **61**(6): 1333–1345.

Osorio, C., Chen, X., Marsico, M., Talas, M., Gao, J. and Zhang, S. (2014). Reducing gridlock probabilities via simulation-based signal control, *Proceedings of the International Symposium of Transport Simulation (ISTS).*

Osorio, C. and Chong, L. (forthcoming). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems, *Transportation Science* .

Osorio, C., Flötteröd, G. and Zhang, C. (2014). A metamodel simulation-based optimization approach for the efficient calibration of stochastic traffic simulators, *Proceedings of the International Symposium of Transport Simulation (ISTS).*

Osorio, C. and Nanduri, K. (2013). Emissions mitigation: coupling microscopic emissions and urban traffic models for signal control. MIT.
URL: *http://web.mit.edu/osorioc/www/papers/osoNanEmissionsSO.pdf*

Osorio, C. and Nanduri, K. (ta). Energy-efficient urban traffic management: a microscopic simulation-based approach, *Transportation Science* .

Pock, T., Pock, M. and Bischof, H. (2007). Algorithmic differentiation: Application to variational problems in computer vision, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **29**(7): 1180–1193.

Powell, M. J. (2002). UOBYQA: unconstrained optimization by quadratic approximation, *Mathematical Programming* **92**(3): 555–582.

Raney, B., Çetin, N., Völlmy, A., Vrtic, M., Axhausen, K. and Nagel, K. (2003). An agent-based microsimulation model of swiss travel: First results, *Networks and Spatial Economics* **3**(1): 23–42.

Rios, L. M. and Sahinidis, N. V. (2013). Derivative-free optimization: A review of algorithms and comparison of software implementations, *Journal of Global Optimization* **56**(3): 1247–1293.

Ross, S. M. (2006). *Simulation*, fourth edn, Academic Press.

Rückelt, J., Sauerland, V., Slawig, T., Srivastav, A., Ward, B. and Patvardhan, C. (2010). Parameter optimization and uncertainty analysis in a model of oceanic CO_2 uptake using a hybrid algorithm and algorithmic differentiation, *Nonlinear Analysis: Real World Applications* **11**(5): 3993–4009.

Sankaran, S., Audet, C. and Marsden, A. L. (2010). A method for stochastic constrained optimization using derivative-free surrogate pattern search and collocation, *Journal of Computational Physics* **229**(12): 4664–4682.

Savage, S. (2012). *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*, Wiley.

Schrank, D., Eisele, B. and Lomax, T. (2012). TTI's 2012 urban mobility report, *Technical report*, Texas A&M Transportation Institute, The Texas A&M University System.

Søndergaard, J. (2003). *Optimization using surrogate models by the space mapping technique*, PhD thesis, Technical University of Denmark, Lyngby, Denmark.

Torczon, V. (1997). On the convergence of pattern search algorithms, *SIAM Journal on Optimization* **7**(1): 1–25.

Yang, Q. and Koutsopoulos, H. N. (1996). A microscopic traffic simulator for evaluation of dynamic traffic management systems, *Transportation Research Part C: Emerging Technologies* **4**(3): 113–129.