Submitted to *Operations Research*
manuscript (Please, provide the mansucript number!)

Carolina Osorio
Civil and Environmental Engineering Department, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, osorioc@mit.edu

Michel Bierlaire
Transport and Mobility Laboratory, School of Architecture, Civil and Environmental Engineering, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, michel.bierlaire@epfl.ch

This paper proposes a simulation-based optimization (SO) method that enables the efficient use of complex stochastic urban traffic simulators to address various transportation problems. It presents a metamodel that integrates information from a simulator with an analytical queueing network model. The proposed metamodel combines a general-purpose component (a quadratic polynomial), which provides a detailed local approximation, with a physical component (the analytical queueing network model), which provides tractable analytical and global information. This combination leads to an SO framework that is computationally efficient and suitable for complex problems with very tight computational budgets.

We integrate this metamodel within a derivative-free trust region algorithm. We evaluate the performance of this method considering a traffic signal control problem for the Swiss city of Lausanne, different demand scenarios and tight computational budgets. The method leads to well-performing signal plans. It leads to reduced, as well as more reliable, average travel times.

Key words: Simulation-based optimization, Traffic control, Metamodel, Queueing

1. Introduction

Microscopic urban traffic simulators embed the most detailed traffic models. They represent individual vehicles and can account for vehicle-specific technologies/attributes. They represent individual travelers and embed detailed disaggregate behavioral models that describe how these travelers make travel decisions (e.g. departure time choice, mode choice, route choice, how travelers respond to real-time traffic information, how they decide to change lanes). They also provide a detailed representation of the underlying supply network (e.g. variable message signs, public transport priorities). Thus, these traffic simulators can describe in detail the interactions between vehicle performance (e.g. instantaneous energy consumption, emissions), traveler behavior and the underlying transportation infrastructure, and yield a detailed description of traffic dynamics in urban networks.

These simulators can provide accurate network performance estimates in the context of what-if analysis or sensitivity analysis. They are therefore often used to evaluate a set of predetermined transportation strategies (e.g., traffic management or network design strategies). Nevertheless, using them to derive appropriate strategies, i.e., to perform simulation-based optimization (SO), is an intricate task.

We focus on transportation problems of the following form:

$$\min_{x\in\Omega} f(x;p) \equiv E[F(x;p)] \tag{1}$$

The objective function f is usually the expected value of a stochastic network performance measure, F. The probability distribution function of F depends on the deterministic decision or control vector x and on deterministic exogenous parameters p. The feasible space Ω consists of a set of general (e.g., nonconvex) constraints that link x to p and f. Closed-form differentiable expressions are available for the constraints.

For instance, a traffic signal control problem can take F as the vehicle travel time and x as the green times for the signalized lanes. Elements such as the total demand or the network topology are captured by p. Every simulation run leads to realizations of F, and involves sampling from the numerous probability distributions that account for uncertainty in, for instance, driver behavior (e.g., route choice for individual drivers) or traffic generation (e.g., headways of vehicles entering the network). For a given x, assume we have observed r independent realizations of F, denoted $F_1(x;p), \dots, F_r(x;p)$. Then, the objective function is approximated by the sample average:

$$\hat{f}(x;p) = \frac{1}{r}\sum_{i=1}^{r} F_i(x;p). \tag{2}$$

In this methodology, the sample size r is kept constant across all points and iterations. Its value for the empirical results of Section 6 is given in Section 4.3.

The various disaggregate traffic models embedded within a microscopic simulator make it a detailed model, but lead to nonlinear objective functions containing potentially several local minima. These objective functions have no available closed-form; we can only derive estimates for them. Additionally, evaluating these estimates accurately is computationally expensive because it involves running numerous replications. As nonlinear, stochastic, generally constrained and computationally-expensive problems, simulation-based urban transportation problems are complex to address.

In order to efficiently address the challenging urban transportation problems that arise in practice, SO algorithms that can identify strategies with improved performance within few simulation runs are needed. This paper proposes a metamodel SO technique that identifies such strategies under very tight computational budgets (i.e., it has good short-term performance). The technique is based on a novel metamodel formulation that enables the combination of detailed, yet intractable, simulated information with less detailed, yet highly tractable, information from an analytical traffic model.

The computational budget is defined as a maximum number of simulation runs n_{max}. Microscopic traffic simulators are typically used in practice under very tight computational budgets. This paper considers similar budgets. For problems with decision vectors of dimension d, the total number of simulation runs (i.e. the computational budget) ranges between d and $5d$. For instance, in Section 6.2 we consider a constrained traffic signal control problem with a decision vector of dimension 51 and a computational budget of 150. Recent work has extended the approach proposed in this paper, enabling it to address larger-scale constrained problems ($d = 100$) considering the same tight computational budget of 150 simulation runs (Osorio and Chong 2012).

This paper is structured as follows. We present a literature review of metamodel SO methods (Section 2). We present the metamodel in Section 3. The metamodel is then embedded within a derivative-free trust region algorithm, which is tailored to the needs of our context (Section 4). We then show how this methodology applies to a fixed-time traffic signal control problem (Section 5), and present empirical results evaluating its performance (Section 6).

2. Literature review

There are three types of approaches to address SO problems: 1) direct-search methods (for reviews see Conn et al. (2009b), Kolda et al. (2003)), 2) stochastic gradient methods (where the gradient of the simulation response is estimated), 3) metamodel (or surrogate) methods. The stochasticity inherent in traffic simulation outputs along with their high computational cost, and their often unavailable source code, makes the accurate estimation of derivatives an expensive task. In order to derive SO techniques for problems with very tight computational budgets, we focus on metamodel SO techniques.

A metamodel (or surrogate model) is an analytical approximation of the objective function. Metamodel optimization methods iterate over two main steps. First, the metamodel is fitted based on a set of simulated observations. Second, it is used to perform optimization and derive a trial point (in this paper the term point refers to a given decision vector value x). The performance of the trial point can be evaluated by the simulator, which leads to new observations. As new observations become available the accuracy of the metamodel can be improved (step 1), leading ultimately to better trial points (step 2).

Metamodels are typically deterministic functions that are much less expensive to evaluate than the underlying simulator. By replacing the stochastic response of the simulation by a deterministic function, deterministic optimization techniques can be used. Furthermore, by using metamodels that are inexpensive to evaluate, the number of objective function evaluations is no longer a limitation. The main limitation remains the number of simulation runs needed such that an accurate metamodel can be built and well-performing trial points can be derived.

Recent reviews of metamodels are given by Conn et al. (2009b), by Barton and Meckesheimer (2006) and by Søndergaard (2003). Metamodels are classified in the literature as either physical or functional metamodels (Søndergaard 2003, Serafini 1998). Physical metamodels consist of application-specific metamodels. Their functional form and parameters have a physical or structural interpretation.

Functional metamodels are generic (i.e., general-purpose) functions that are chosen based on their analytical tractability, but do not take into account any information with regards to the specific objective function, let alone the structure of the underlying problem. They are often a linear combination of basis functions from a parametric family.

The most common choice for a functional metamodel is the use of low-order polynomials (e.g., linear or quadratic). Quadratic polynomials are used, at least asymptotically, as surrogates in most trust region methods (Conn et al. 2000). They are also used in sequential response surface methodologies for unconstrained and constrained SO problems (Marti 2008, Kleijnen 2008). Spline models have also been used, although their use within an SO framework has focused on univariate or bivariate functions, and as Barton and Meckesheimer (2006) mention: “unfortunately, the most popular and effective multivariate spline methods are based on interpolating splines, which have little applicability for SO”. Radial basis functions (Oeuvray and Bierlaire 2009, Wild et al. 2008) and Kriging surrogates (Kleijnen et al. 2010, Booker et al. 1999) have also been proposed.

Functional metamodels are the most common metamodels used to perform simulation-based optimization, since they can be used to approximate any objective function. Nonetheless, they capture little information about the structure of the underlying problem and often require a large number of simulation runs to be fitted. They can provide accurate local approximations (i.e., approximations of the objective function near simulated points), yet often fail to provide suitable approximations in non-simulated regions. This limits their use for applications with tight computational budgets.

We propose a metamodel that combines a functional component with a physical component. The purpose of the functional component is to provide a detailed local approximation of the function. The physical component should provide a good, yet less detailed, global approximation. It improves the computational tractability of the SO method, by overcoming the limitations of the simulation model, namely an inability to provide closed-form continuous expressions for the performance measures and their first-order derivatives.

3. Metamodel

The metamodel combines information from two traffic models: a simulation model and an analytical network model. We first present these two models, we then describe how they are combined.

Simulation model. We use a microscopic traffic simulator that models the behavior of individual drivers within the network. Trips are generated based on an origin-destination matrix, along with a headway model. Driver behavior is modeled using car following, lane changing, gap acceptance and route choice models. In this work we use the Aimsun simulator. For a detailed description of its behavioral models see TSS (2008). When simulating the performance of a set of points $\{x^1, x^2, \dots\}$ we obtain estimates of their performance $\{\hat{f}(x^1;p), \hat{f}(x^2;p), \dots\}$ (presented in Equation (2)).

Analytical queueing network model The analytical urban traffic model is formulated in Osorio and Bierlaire (2009b), it is based on the more general queueing network model of Osorio and Bierlaire (2009a). Alternatively, for large-scale networks the model of Osorio and Chong (2012) can be used, and for dynamic control problems the model in Osorio et al. (2011) is suitable. In this paper, we use the queueing model of Osorio and Bierlaire (2009b) to formulate a metamodel for simulation-based optimization.

The queueing model formulation combines ideas from traditional urban traffic models, several national urban transportation norms and queueing theory. An urban road is mapped into a set of queues. Finite capacity queues are used in order to account for the limited space capacity of roads. The finite capacity queueing theory (FCQT) notion of *blocking* is used to describe where congestion arises and how it propagates throughout the network. By resorting to FCQT, the model captures the key traffic interactions and the underlying network structure, e.g., how upstream and downstream queues interact, and how this interaction is linked to network congestion. The model consists of a system of nonlinear equations (which are given in the Appendix). It is formulated based on a set of exogenous parameters q that capture the network topology, the total demand, as well as the turning probabilities . A set of endogenous variables y describe the traffic interactions, e.g., spillback probabilities, spillback diffusion rates. For a detailed description of the elements of q and y, we refer the reader to the Appendix. For a given decision vector x, the network model yields the objective function $T(x,y;q)$, which is a deterministic approximation of $f(x;p)$ (Equation (1)).

We recall here the notation that we have introduced so far:

x decision vector;
T approximation of the objective function derived by the queueing model;
$\hat{f}$ estimate of the objective function derived by the simulation model;
y endogenous queueing model variables;
q exogenous queueing model parameters;
p exogenous simulation model parameters.

We now describe how $\hat{f}$ and T are combined to derive the metamodel m. The proposed metamodel m combines a functional component with a physical component as follows.

$$m(x,y;\alpha,\beta,q) = \alpha T(x,y;q) + \phi(x;\beta), \tag{3}$$

where ϕ is the functional component, α and β are parameters of the metamodel.

In this paper, the metamodel is embedded within a trust region (TR) derivative-free algorithm. The main idea of trust region methods is to build, at each iteration, a model of the objective function which one “trusts” in a neighborhood of the current iterate, the *trust region*. The most common model choice is a quadratic polynomial. Additionally, numerical experiments for derivative-free TR methods indicate that quadratic polynomials with a diagonal second derivative matrix are often more efficient than full quadratic polynomials (Powell 2003). We therefore define ϕ as a quadratic polynomial in x with a diagonal second derivative matrix.

$$\phi(x;\beta) = \beta_1 + \sum_{j=1}^{d} \beta_{j+1} x_j + \sum_{j=1}^{d} \beta_{j+d+1} x_j^2, \tag{4}$$

where d is the dimension of x, x_j and β_j are the j^{th} components of x and β, respectively. Other formulations, such as minimum Frobenius norm quadratic models, can also be suitable for this framework.

At each iteration of a trust region algorithm the objective function is evaluated at a set of points. The metamodel is then constructed based on objective function observations. Traditionally, trust region methods fit the polynomial via interpolation. In this framework, we fit the metamodel via regression. At each iteration of the SO algorithm, the metamodel is fitted using the simulated observations obtained at the current iteration, as well as all simulated observations collected at previous iterations.

The parameters β and α of the metamodel are fitted by solving a least squares problem. At a given iteration, the model approximates the objective function in a neighborhood of the current iterate. In order to give more importance to observations that correspond to points that are near the current iterate, we associate weights with each observation. Assume at iteration k we have simulated the performance of a set of n_k distinct points $\{x^1, x^2, \dots x^{n_k}\}$, then the least squares problem is formulated as follows.

$$\min_{\alpha,\beta} \sum_{i=1}^{n_k} \left\{ w_{ki} \left(\hat{f}(x^i;p) - m(x^i, y^i; \alpha, \beta, q) \right) \right\}^2 + (w_0.(\alpha - 1))^2 + \sum_{i=1}^{2d+1} (w_0.\beta_i)^2, \tag{5}$$

where x^i represents the i^{th} point, with corresponding endogenous queueing model variables y^i and simulated estimate $\hat{f}(x^i;p)$. The weight associated at iteration k with the i^{th} estimate is denoted w_{ki}. The parameter w_0 represents a fixed weight.

The first squared term of Equation (5) represents the weighted distance between the simulated estimates and the metamodel predictions. The next two squared terms measure the distance between the parameters and their initial values. These terms ensure that the least squares matrix is always of full rank (regardless of the number or location of the simulated points). The initial values used here (one for α and zero for β) lead to an initial metamodel that is based only on the queueing model. This is of interest when starting off the algorithm with few or even no simulated observations.

The weights, w_{ki}, capture the importance of each point with regards to the current iterate. We use what is known as the *inverse distance* weight function (Atkeson et al. 1997) along with the Euclidean distance. This leads to the following weight parameters: $w_{ki} = 1/(1 + \|x_k - x^i\|_2)$, where x_k is the current iterate, and x^i is the i^{th} point.

The weight of a given point is therefore inversely proportional to its distance from the current iterate. This allows us to approximately have a Taylor-type behavior, where observations corresponding to local points have more weight. The least squares problem is solved using the Matlab routine *lsqlin* (Mathworks, Inc. 2008).

4. Optimization algorithm

4.1. Multi-model algorithms

In order to integrate the proposed metamodel within an existing optimization method, we resort to multi-model (or hybrid) algorithms, which allow for an arbitrary metamodel to be used. These methods share a common motivation, which is to combine the use of models with varying evaluation costs (low versus high-fidelity models, or coarse versus fine models). Multi-model algorithms include that of Carter (1986) (see references herein for previous multi-model frameworks), Booker et al. (1999), Alexandrov et al. (2001), and Bandler et al. (2006). Alexandrov and Lewis (2001) give a comparison of some of these multi-model algorithms.

Conn et al. (2009a) recently proposed a trust region derivative-free framework for unconstrained problems. It builds upon the *Basic TR algorithm* of Conn et al. (2000). In this paper, we choose to tailor the algorithm of Conn et al. (2009a) and integrate the proposed metamodel within it. Other approaches, such as that of Booker et al. (1999), can also be tailored to integrate the proposed metamodel. Our choice is based on the following reasons. Among the two main strategies used to ensure global convergence, line search and trust region methods, the latter are more appropriate for our context since they "extend more naturally than line search methods to models that are not quadratics with positive definite Hessians" (page 13 of Carter (1986)). The most common approach for fitting metamodels within a trust region (TR) framework is interpolation. Nevertheless, for noisy functions we believe that regression is more appropriate since it is less sensitive to the inaccuracy of the observations. This algorithm allows for arbitrary metamodels and makes no assumption on how these metamodels are fitted (interpolation or regression). Given the difficulty and cost of obtaining accurate gradient estimates for simulation-based urban transportation problems, derivative-free (DF) methods are appealing since they do not require derivative estimations.

4.2. Algorithm

In this section, we omit references to the exogenous parameters of the simulation model, i.e., we denote $f(x;p)$ as $f(x)$.

0. **Initialization.**

Define for a given iteration k: $m_k(x,y;\nu_k,q)$ as the metamodel (denoted hereafter as $m_k(x)$), x_k as the iterate, Δ_k as the trust region radius, $\nu_k=(\alpha_k,\beta_k)$ as the vector of parameters of m_k, n_k as the total number of simulation runs carried out up until and including iteration k, u_k as the number of successive trial points rejected, ε_k as the measure of stationarity (norm of the derivative of the Lagrangian function of the TR subproblem with regards to the endogenous variables) evaluated at x_k.

The constants $\eta_1,\gamma,\gamma_{inc},\varepsilon_c,\bar{\tau},\bar{d},\bar{u},\Delta_{max}$ are given such that: $0<\eta_1<1$, $0<\gamma<1<\gamma_{inc}$, $\varepsilon_c>0$, $0<\bar{\tau}<1$, $0<\bar{d}<\Delta_{max}$, $\bar{u}\in\mathbb{N}^*$. Set the total number of simulation runs permitted (across all points) n_{max}, this determines the computational budget. Set the number of simulation replications per point r (defined in Equation (2)).

Set $k = 0, n_0 = 1, u_0 = 0$. Determine x_0 and Δ_0 ($\Delta_0 \in (0, \Delta_{max}]$).

Compute T and $\hat{f}$ at x_0, fit an initial model m_0 (i.e., compute ν_0).

1. **Criticality step.** If $\varepsilon_k \leq \varepsilon_c$, then switch to *conservative mode* (detailed in Section 4.3).

2. **Step calculation.** Compute a step s_k that reduces the model m_k and such that $x_k + s_k$ (the trial point) is in the trust region (i.e. approximately solve the TR subproblem).

3. **Acceptance of the trial point.** Compute $\hat{f}(x_k + s_k)$ and

$$\rho_k = \frac{\hat{f}(x_k) - \hat{f}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

- If $\rho_k \geq \eta_1$, then accept the trial point: $x_{k+1} = x_k + s_k,\ u_k = 0$.

- Otherwise, reject the trial point: $x_{k+1} = x_k,\ u_k = u_k + 1$.

Include the new observation in the set of sampled points ($n_k = n_k + r$), update the weights w and fit the new model m_{k+1}.

4. **Model improvement.** Compute $\tau_{k+1} = \frac{\|\nu_{k+1} - \nu_k\|}{\|\nu_k\|}$. If $\tau_{k+1} < \bar{\tau}$, then improve the model by simulating the performance of a new point x, which is sampled from a distribution such as the one defined in Section 4.3. Evaluate T and $\hat{f}$ at x. Include this new observation in the set of sampled points ($n_k = n_k + r$). Update m_{k+1}.

5. **Trust region radius update.**

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_{inc}\Delta_k, \Delta_{max}\} & \text{if } \rho_k > \eta_1 \\ \max\{\gamma\Delta_k, \bar{d}\} & \text{if } \rho_k \leq \eta_1 \text{ and } u_k \geq \bar{u} \\ \Delta_k & \text{otherwise} \end{cases}$$

If $\rho_k \leq \eta_1$ and $u_k \geq \bar{u}$, then set $u_k = 0$.

If $\Delta_{k+1} \leq \bar{d}$, then switch to *conservative mode.*

Set $n_{k+1} = n_k, u_{k+1} = u_k,\ k = k + 1$.

If $n_k < n_{max}$, then go to Step 1. Otherwise, stop.

4.3. Algorithmic details

Asymptotic properties. Recall that the primary purpose of the proposed methodology is to enable the identification of points with improved performance under very tight computational budgets. To achieve this, we tailor the Conn et al. (2009a) algorithm such as to yield an algorithm with good short-term performance.

In order to ensure global convergence for unconstrained problems, the Conn et al. (2009a) algorithm requires that within a uniformly bounded number of steps a *certifiably fully linear* model (i.e., a model that satisfies first-order Taylor-type bounds) can be obtained. Whenever a certificate of full linearity is required, our algorithm switches to the *conservative mode.* For the complex transportation problems and the tight computational budgets that we consider, this mode has never been triggered in any of our experiments.

If the asymptotic performance (and therefore the conservative mode) of the algorithm is of interest, we suggest to follow the lines of traditional SO techniques that use a quadratic polynomial as a metamodel (i.e., set α of Equation (3) to zero), along with traditional sampling strategies and statistical tests at various steps of the algorithm (Shapiro et al. 2009, Bharadwaj and Kleywegt 2008, Bastin et al. 2006, Kleywegt and Shapiro 2001, Shapiro 2000). Nonetheless, this will come at the cost of a much larger computational budget, and is therefore out of the scope of this paper.

Criticality step. The criticality step of the algorithm ensures that if the measure of stationarity goes under a given threshold, then the model can be made *certifiably fully linear* so that its stationarity measure can be trusted.

Step calculation. Details regarding the TR subproblem are given for the traffic signal control problem in Section 5.2.

Acceptance of the trial point. The actual reduction of the objective function is compared to the reduction predicted by the model, this determines whether the trial point is accepted or rejected.

Model improvement step. This step aims to sample points that can improve the accuracy of the metamodel. Such points can, for instance, improve the geometric properties of the sampled space (e.g., attempting to fully span the feasible space such that a full rank least squares matrix is obtained, or in the case of interpolation methods improving the poisedness of the sample (Conn et al. 2008a; Conn et al. 2008b)). To sample we draw uniformly from the feasible space.

TR radius update. In the Conn et al. (2009a) algorithm the TR radius can be reduced if the model is *fully linear* but has not performed well. As long as $\alpha \neq 0$, we cannot certify whether the model is *fully linear*. We therefore reduce the TR radius after $\bar{u}$ successive trial points have been rejected. If the TR radius reaches a lower bound $\bar{d}$, then we switch to the *conservative mode* in order to obtain a *fully linear* model. In our experiments, this was never necessary.

Sample sizes. The simulator is called at steps 3 and 4 of the algorithm to evaluate the performance of a trial point and a model improvement point, respectively. Given the tight computational budgets considered in this paper, each time the simulator is used to evaluate the performance of a point only one replication is run (i.e., r of Equation (2) equals 1).

Algorithmic parameters. The following values are used for the parameters of the TR algorithm: $\Delta_{max} = 10^{10}$, $\Delta_0 = 10^3$, $\eta_1 = 10^{-3}$, $\gamma = 0.9$, $\gamma_{inc} = 1.2$, $\epsilon_c = 10^{-6}$, $\bar{\tau} = 0.1$, $\bar{d} = 10^{-2}$, $\bar{u} = 10$, $w_0 = 0.1$. Typical values for TR parameters are given in Carter (1986).

Convergence of constrained DF TR algorithms. As detailed by Conn et al. (2009b), DF TR methods are a relatively recent topic. The Conn et al. (2009a) algorithm ensures global convergence for unconstrained problems. Conn et al. (1998) propose a method to solve problems with general constraints using an unconstrained TR algorithm. The transportation problems that we are interested in solving fall into the category of what they call *easy* constraints. These are general constraints that are continuously differentiable and whose first-order partial derivatives can be computed relatively cheaply (with regards to the cost of evaluating the objective function). In their approach, they include such constraints in the TR subproblem, which ensures that all trial points are feasible. Conn et al. (2009b) mention that such an approach is often sufficient in practice.

Here we use the TR algorithm proposed by Conn et al. (2009a) for unconstrained methods, and extend its use to constrained problems as Conn et al. (1998) suggest. That is, we include the constraints in the TR subproblem to ensure that all trial points are feasible. Section 5.2 formulates the TR subproblem for a signal control problem.

5. Traffic signal control

5.1. Problem formulation

We illustrate the use of this framework with a traffic signal control problem. A review of signal control formulations is given in Appendix A of Osorio (2010). Traffic signal setting strategies can be either fixed-time or traffic-responsive strategies. In this paper we consider fixed-time (also called pre-timed) strategies. These strategies use historical traffic data, and yield one traffic signal setting for the considered time of day. The traffic signal optimization problem is solved offline.

We consider the same problem as in Osorio and Bierlaire (2009b). In this problem, the signal plans of several intersections are determined jointly. For a given intersection and a given time interval (e.g., evening peak period), a fixed-time signal plan is a cyclic (i.e., periodic) plan that is repeated throughout the time interval. The duration of the cycle is the time required to complete one sequence of signals. For example, the cycle times of the model considered in Section 6 are of 90 or 100 seconds. A phase is defined as a set of traffic streams that are mutually compatible and that receive identical control.

The cycle of a signal plan is divided into a sequence of periods called stages. Each stage consists of a set of mutually compatible phases that all have green. The cycle may also contain all-red periods, where all streams have red indications. The difference between the cycle time and the sum of the all-red times is called the available cycle time. The ratio of the available cycle time and the cycle time is called the available cycle ratio.

In order to coordinate the signals of adjacent intersections (e.g., to create green waves along arterials), offsets variables are used. An offset is the difference in time between the reference points of the cycles of two intersections. The ratio of green times to cycle time is known as the green split.

In this paper, we consider a fixed-time signal control problem where the offsets, the cycle times and the all-red durations are fixed. The stage structure is also given, i.e., the set of lanes associated with each stage as well as the sequence of stages are both known. This is known as a stage-based approach. The decision variables of this problem consist of the green splits of the different intersections.

This is a traditional signal control problem, which is easy to solve for a single intersection. For multiple intersections this problem becomes difficult due to the need to account for the interactions between the vehicular queues of adjacent intersections.

To formulate this problem we introduce the following notation:

b_i	available cycle ratio of intersection i;
$x(j)$	green split of phase j;
x_L	vector of minimal green splits;
$\mathscr{I}$	set of intersection indices;
$\mathscr{P}_I(i)$	set of phase indices of intersection i.

The problem is traditionally formulated as follows:

$$\min_x f(x;p) \equiv E[F(x;p)] \tag{6}$$

subject to

$$\sum_{j \in \mathscr{P}_I(i)} x(j) = b_i, \ \forall i \in \mathscr{I} \tag{7}$$

$$x \geq x_L. \tag{8}$$

The decision vector x consists of the green splits for each phase. The objective is to minimize the expected travel time (Equation (6)). The linear constraints (7) link the green times of the phases with the available cycle time for each intersection. The bounds (8) correspond to minimal green time values for each phase. These have been set to 4 seconds according to the Swiss transportation norm (VSS 1992).

5.2. Trust region subproblem

At a given iteration k the TR subproblem includes three more constraints than the previous problem. It is formulated as follows:

$$\min_{x,y} \; m_k = \alpha_k T(x,y;q) + \phi(x;\beta_k) \tag{9}$$

subject to

$$\sum_{j \in \mathscr{P}_I(i)} x(j) = b_i, \; \forall i \in \mathscr{I} \tag{10}$$

$$h_2(x,y;q) = 0 \tag{11}$$

$$\|x - x_k\|_2 \leq \Delta_k \tag{12}$$

$$y \geq 0 \tag{13}$$

$$x \geq x_L, \tag{14}$$

where x_k is the current iterate, Δ_k is the current trust region radius, α_k and β_k are the current metamodel parameters, and h_2 of Equation (11) represents the queueing model (System of Equations (15) given in the Appendix). Constraint (12) is the TR constraint, using the Euclidean norm. The endogenous variables of the queueing model are subject to positivity constraints (Equation (13)). Thus, the TR subproblem consists of a nonlinear objective function subject to nonlinear and linear equalities, a nonlinear inequality and bound constraints. The analytical form of T for the signal control problem considered in this paper is given in the Appendix.

For a problem with ℓ lanes (i.e., queues), s signalized lanes, n endogenous phases and r controlled intersections, there are $5\ell + s + n$ endogenous variables, which consist of 5 endogenous queueing model variables per lane, the green splits for each signalized lane and the green splits for each phase. There are $\ell + s + r$ linear and 4ℓ nonlinear equations, as well as $5\ell + s + n + 1$ inequalities (lower bound constraints for all endogenous variables and the trust-region constraint).

This problem is solved with the Matlab routine for constrained nonlinear problems, *fmincon*, for which one option, which we use, resorts to a sequential quadratic programming method (Coleman and Li 1996, Coleman and Li 1994). We set the tolerance for relative change in the objective function to 10^{-3} and the tolerance for the maximum constraint violation to 10^{-2}.

5.3. Signal plan features

Sampling. The model improvement step of the algorithm attempts to diversify the set of sampled points by drawing points uniformly from the feasible space. A feasible signal plan is defined by Equations (7) and (8) (or equivalently Equations (10) and (14)). We draw uniformly from this space, using the code of Stafford (2006). Given this signal plan, we solve the network model (Equation (11)) following the procedure described in Osorio and Bierlaire (2009a).

Explanatory/independent variables. The polynomial component of the metamodel, ϕ, is a quadratic polynomial in the decision variables x, which are the phase variables of the different intersections. For a given intersection the phase variables are linked through the linear Equation (7) (or equivalently Equation (10)). In order to use independent explanatory variables, we exclude one phase per intersection. Thus for a set of i intersections and p phases, the polynomial is a function of $p - i$ phase variables, and has a total of $2(p - i) + 1$ coefficients.

6. Empirical analysis

We evaluate and illustrate the use of this framework with case studies based on road networks within the Swiss city of Lausanne. We use a calibrated microscopic traffic simulation model of the Lausanne city center. This model (Dumont and Bert 2006) is implemented with the Aimsun simulator (TSS 2008). Details regarding the Lausanne network are given in Osorio and Bierlaire (2009b).

We consider two Lausanne city subnetworks. Firstly, we consider a simplified demand distribution. Secondly, we analyze the performance of the method given the demand of the city of Lausanne for the evening peak hour, and control the plans of a larger set of intersections. For both subnetworks, we consider tight computational budgets in order to evaluate the ability of the proposed approach to address important traffic control problems in a computationally efficient manner. More detailed results can be found in Osorio and Bierlaire (2010).

To refer to the different metamodels we use the notation of Equation (3). In both sections, we compare the performance of the signal plans derived by the use of three models: (i) the proposed metamodel, m, (ii) a quadratic polynomial with diagonal second derivative matrix, (i.e., the metamodel consists of ϕ), (iii) the queueing model, T. Procedure (iii) is proposed in Osorio and Bierlaire (2009b), and resorts to the same algorithm as the one used to solve the TR subproblem. The corresponding algorithms are referred to as Am, $A\phi$ and AT, respectively.

6.1. Lausanne subnetwork with simplified demand distribution

We consider the Lausanne road network with a simplified demand distribution. We control a set of two adjacent signalized intersections. Demand arises at the nine centroids nearest to these two intersections. The simulation setup considers a 20 minute run time, preceded by a 15 minute warm-up time.

The subnetwork consists of 12 roads with 21 lanes, 13 of which are signalized. A total of 13 phases are considered variable (i.e., the dimension of the decision vector is 13). This leads to a polynomial ϕ with 23 coefficients. Since each lane is modeled as a queue, this subnetwork is modeled as a set of 21 queues. The corresponding TR subproblem consists of

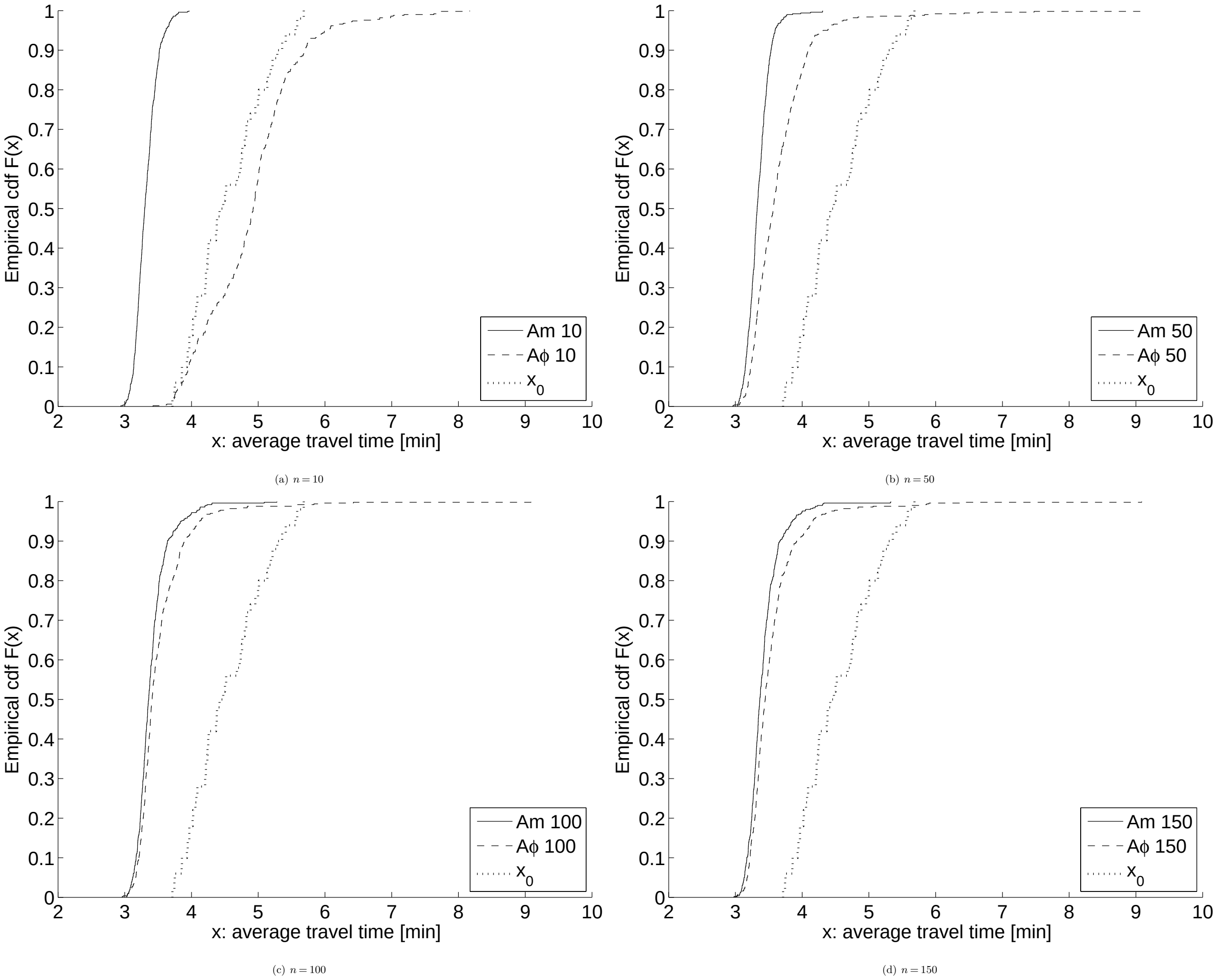
131 endogenous variables with their corresponding lower bound constraints, 84 nonlinear and 36 linear equalities.

Firstly, we consider a tight computational budget, which is defined as a maximum number of simulation runs that can be carried out. The computational budget is set to 150 runs. We consider the performance of the signal plans derived by Am and $A\phi$. We run each algorithm 10 times. We then compare their performance for increasing total number of simulation runs, which we denote n. Initially, no simulated observations are available, i.e., we start off with an empty sample. We initialize all runs with the same signal plan, which is a uniformly drawn signal plan generated with the method of Stafford (2006).

To compare the performance of both methods for a given number of simulation runs (i.e., a given n), we consider the 10 signal plans derived and evaluate their performance by running 50 replications of the simulation model. We then compare the empirical cumulative distribution function (cdf) of the average travel times over all 10 signal plans and 50 replications, i.e., each cdf consists of a set of 500 observations.

Figure 1 considers a set of four plots. Each plot displays results for the initial plan, and the plans derived by both methods for a given n. Each plot considers a different n value. Plots 1(a)-1(d) consider, respectively, n values of 10, 50, 100 and 150. The cdf labeled x_0 denotes the cdf of the initial random plan. The numbers denote the corresponding n values, e.g., the cdf labeled "$A\phi$ 10" corresponds to the signal plans proposed by $A\phi$ after running 10 simulation runs ($n = 10$).

For all four n values, Am leads to signals plans with improved average travel times, when compared to both the initial plan and the plans proposed by $A\phi$. It also yields reduced variance in the average travel times across signal plans and replications. The plan proposed by $A\phi$ for $n = 10$ (plot 1(a)) has worse performance than the initial plan. Once n reaches 50 (plot 1(b)), $A\phi$ leads to signal plans with improved performance compared to the initial plan. For n values of 100 and 150 (plots 1(c) and 1(d)), the plans derived



(a) $n = 10$

(b) $n = 50$

(c) $n = 100$

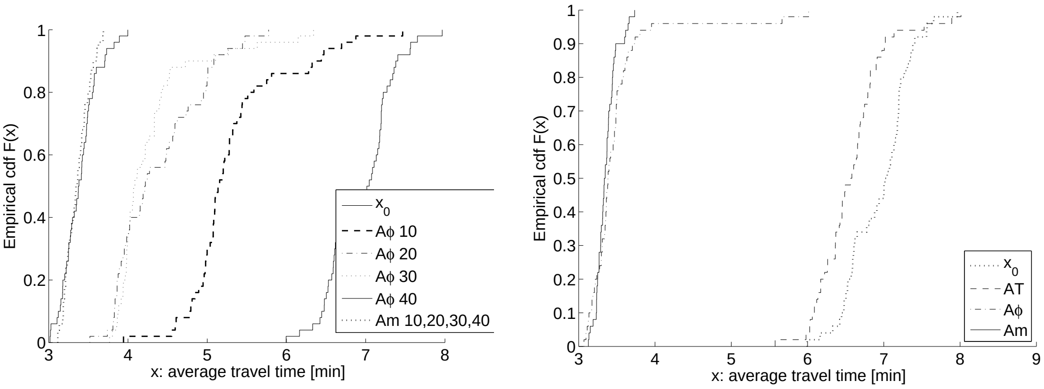
(d) $n = 150$

Figure 1 Empirical cdf's of the average travel times considering an initial random signal plan.

by Am and $A\phi$ have similar performance, with Am leading to signal plans with smaller variance.

Secondly, we allow for a larger computational budget. We allow for a total of 750 simulation runs. We run each method once. As before, we initialize both methods with a uniformly drawn initial signal plan and start off with an empty sample.

Figure 2(a) considers the signal plans derived by Am and $A\phi$ for $n \in \{10, 20, 30, 40\}$. For each signal plan, the figure displays the empirical cdf of the average travel times over the 50 replications. Once again, the cdf denoted x_0 corresponds to the initial random plan. The signal plan derived by Am is the same when running $10, 20, 30$ and 40 simulation runs. For $n = 10$, the plans of both Am and $A\phi$ lead to improved average travel times,



(a) $n \in \{10, 20, 30, 40\}$

(b) $n = 750$

Figure 2 Empirical cdf's of the average travel times considering an initial random signal plan and evaluating the performance as n increases from 10 to 40 (left plot) and at $n = 750$ (right plot).

when compared to the initial plan. As the number of simulation runs increases, $A\phi$ leads to plans with improved performance. At $n = 40$, its performance is similar to that of the signal plan proposed by Am. The latter leads to reduced variance in the average travel time.

Figure 2(b) considers the signal plans proposed by Am and $A\phi$ at $n = 750$, as well as the initial signal plan and the signal plan proposed by the queueing method, AT. It displays for each method the cdf of the average travel times. All three methods, Am, $A\phi$ and AT, lead to signal plans with improved performance compared to the random initial signal plan. The methods that use simulated observations throughout the optimization process, Am and $A\phi$, lead to improved signal plan performance when compared to the queueing method AT. The proposed method, Am, also yields the signal plan with the smallest variance in the average travel times.

6.2. Lausanne subnetwork with evening peak hour demand

We evaluate the performance of the proposed method by considering a larger subnetwork of the Lausanne city center. The Lausanne city road network model is displayed in Figure 3(a). The considered subnetwork is located in the city center, and is delimited in

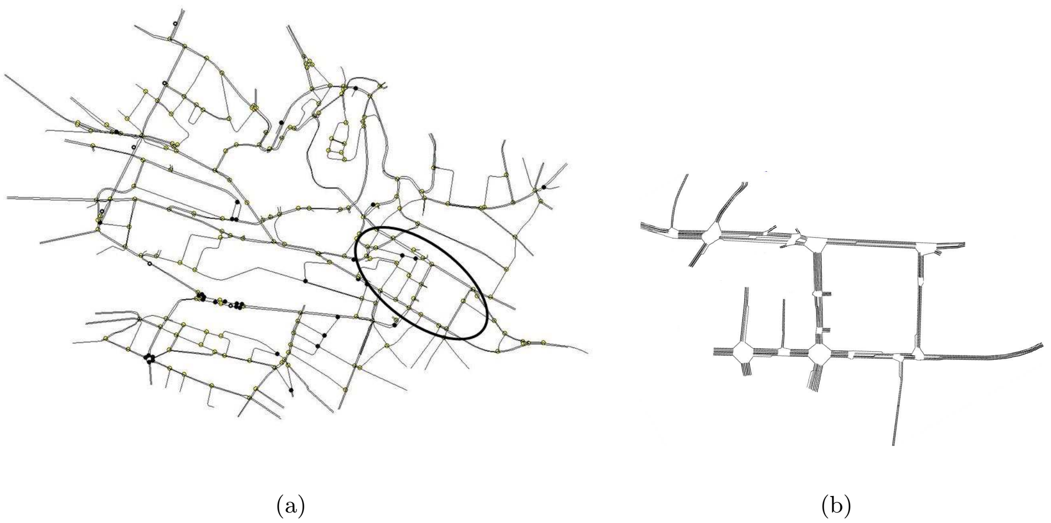


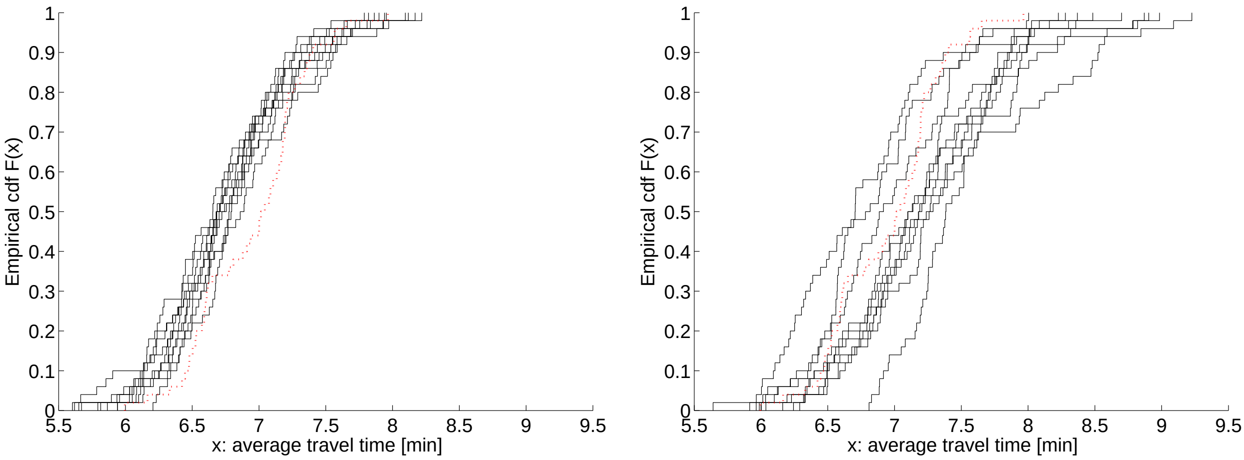
Figure 3 Lausanne city road network model (left) and subnetwork model (right).

Figure 3(a) by an oval. The subnetwork is displayed in detail in Figure 3(b). It is presented in detail in Osorio and Bierlaire (2009b). We use the demand of the evening peak period (17h-18h). The simulation outputs used both to fit the metamodel and to evaluate the performance of the derived signal plans are the subnetwork average travel times.

This subnetwork contains 48 roads and 15 intersections. The signalized intersections have a cycle time of either 90 or 100 seconds. Nine intersections are signalized and control the flow of 30 roads. There are 102 lanes, 60 of which are signalized. There are a total of 51 phases that are considered variable.

The queueing model of this subnetwork consists of 102 queues (one queue for each lane). The TR subproblem consists of 621 endogenous variables with their corresponding lower bound constraints, 408 nonlinear equality constraints, 171 linear equality constraints and 1 nonlinear inequality constraint.

Note that this problem is considered a large-scale problem for existing unconstrained DF methods, not to mention the added complexity of the nonlinear constraints. In particular, the problem has 51 decision variables. Thus if one were to resort to a classical interpolation-based quadratic polynomial surrogate, the evaluation of the performance of 1378 points would be necessary to fit the full polynomial, and in order to obtain an accurate evaluation,



(a) Metamodel m

(b) Metamodel ϕ

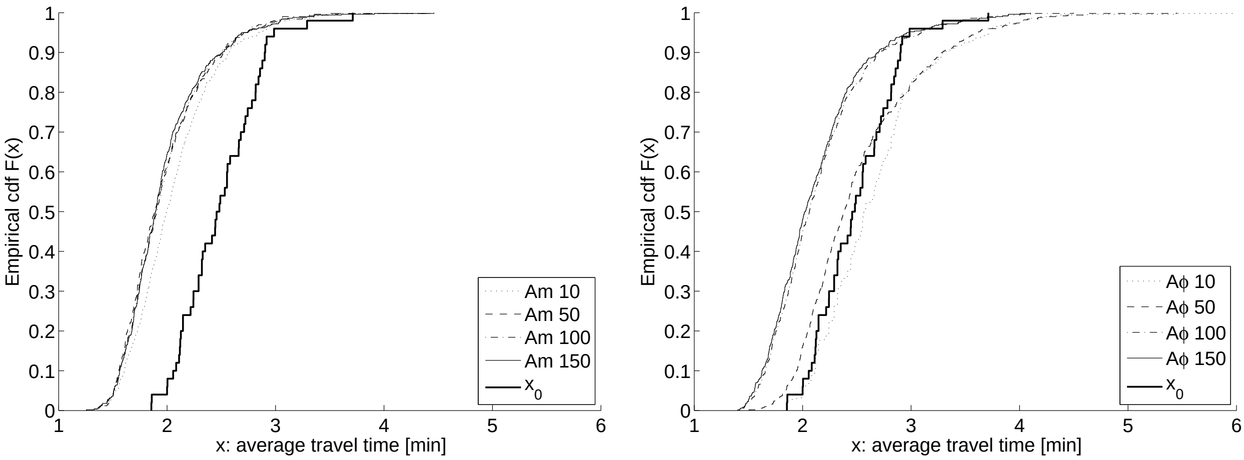
Figure 4 Empirical cdf's of the average travel times considering an initial random signal plan.

a large number of replications would need to be carried out at each point. This is because for a problem with n decision variables $(n+1)(n+2)/2$ suitably sampled points (i.e., well poised (Conn et al. 2000, Conn et al. 2009b)) are necessary to fit the full quadratic.

We allow for a maximum number of 150 simulation runs that can be carried out, and no initial observation available. For a given initial signal plan, we run the corresponding algorithm 10 times, deriving 10 signal plans. We then evaluate the performance of each of these signal plans by running 50 replications of the simulation model. All simulations are preceded by a 15 minute warm-up period. To compare the methods, we consider the empirical cdf of the average travel times.

Each plot of Figure 4 considers one method. The left plot considers Am, the right plot considers $A\phi$. Each plot displays 11 cdf's: the cdf of the initial plan (thick solid curve), and the cdf's of the 10 signal plans obtained with the corresponding method (thin solid curves). All 10 plans proposed by Am yield improved average travel times compared to the initial plan. For $A\phi$, there are cases where after 150 simulation runs the algorithm does not identify a plan with improved performance compared to the initial plan.

Hereafter, we consider the empirical cdf of the average travel times over all 10 signal plans and 50 replications, i.e., each cdf consists of a set of 500 observations. Each plot



(a) Metamodel m

(b) Metamodel ϕ

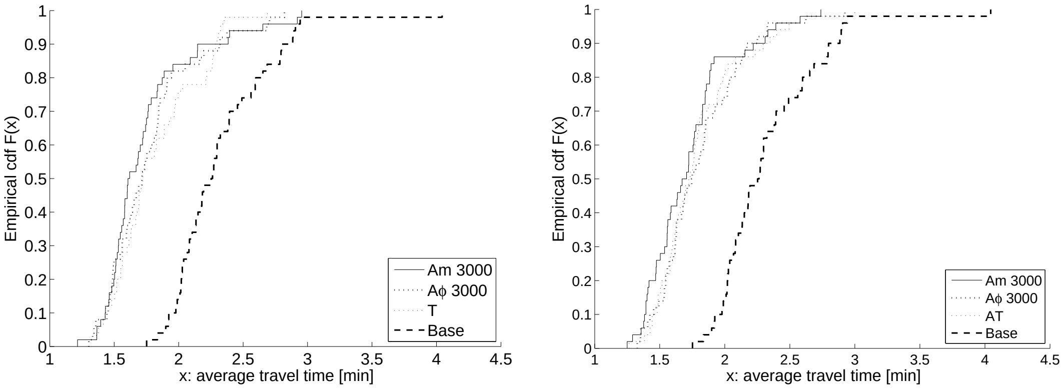
Figure 5 Empirical cdf's of the average travel times considering an initial random signal plan, running 10 instances of each method, and evaluating their performance at different n values.

of Figure 5 considers one method, and displays the cdf's of the initial plan, and of the plans derived after running 10, 50, 100 and 150 simulation runs. For all n values, Am (left plot) yields signal plans with improved distributions. The distributions also improve as n increases. For n values of 10 and 50, $A\phi$ (right plot) yields signal plans with worse performance compared to the initial plan. For n values of 100 and 150, it identifies plans with improved performance.

We now consider a case with a higher computational budget. We allow for 3000 simulation runs and consider a random initial point. In this case, we run the algorithm once. We then evaluate the performance of the derived plans by running 50 replications of the simulation model.

Figure 6(a) presents the cdf's of the average travel times across the 50 replications, considering the plans derived by Am, $A\phi$ and AT. It also presents the performance of an existing signal plan for the city of Lausanne (denoted $Base$).

All three methods (Am, $A\phi$ and AT) lead to improvement compared to the existing plan for the city of Lausanne. All three methods have similar performance, with Am leading to signal plans with slightly lower travel times. The method that consists of only



(a)

(b)

Figure 6 Empirical cdf's of the average travel times considering different initial random signal plans and allowing for 3000 simulation runs.

the queueing model and does not resort to the use of simulated observations, AT, has very good performance.

We perform the same experiment, allowing for 3000 simulation runs, with a different initial point. Figure 6(b) presents the corresponding results. The same conclusions as for Figure 6(a) hold.

We evaluate the computational time required by Am. We consider the two instances of 3000 simulations, previously presented. The optimization algorithm is instantaneous (less than 1 second) in more than 95% of the runs, whereas the simulation run time for one replication is typically between 1 and 2 minutes. This illustrates the computational cost of each simulation and the need to limit the number of evaluations of the simulation model.

The results of Section 6 indicate that the proposed metamodel systematically yields signal plans with improved performance when compared to the initial plans, whereas limiting the metamodel to a quadratic polynomial may fail to do so. The proposed technique also leads to plans with reduced variance in the average travel times, i.e., it increases travel time reliability.

7. Conclusions

This paper presents a simulation-based optimization framework for the design and management of congested urban networks. It proposes a metamodel that combines information from a traffic simulation tool and an analytical network model. It builds upon the classical approach of using a general-purpose quadratic polynomial as a metamodel, by combining the polynomial with the analytical network model. It integrates this metamodel within a derivative-free trust region optimization algorithm.

The performance of this approach is evaluated on a fixed-time signal control problem with two subnetworks of the city of Lausanne. We consider cases with tight computational budgets, and initialize all runs with an empty set of simulated observations. The analytical information provided by the queueing network model enables the identification of well performing trial points within few simulation runs, and continues to perform well as the number of runs increases. This illustrates the added valued of the structural information provided analytically by the network model. The method is computationally efficient and suitable for complex problems with tight computational budgets.

Efficiently tackling unconstrained high dimensional problems (e.g., more than 200 variables) is one of the main limitations of existing derivative-free methods, not to mention the added complexity of constrained and stochastic problems. The generic metamodels used in these algorithms require a moderate to large number of simulation runs to initially fit the metamodel of interest. The proposed combination of generic metamodels with application-specific physical metamodels allows the algorithm 1) to overcome the need for a substantial initial sample, 2) to identify trial points with good performance since the first iterations, and thus 3) to address high dimensional problems under tight computational budgets.

The accuracy and the computational efficiency of this method can be improved by investigating different sample size calculation and allocation strategies, as well as by integrating statistical tests at various steps of the algorithm, such as those presented in Shapiro et al. (2009).

This simulation-based optimization method enables the use of stochastic traffic simulators to go beyond a what-if analysis. With this framework, we can address complex problems that account, in detail, for the performance of individual vehicles (e.g., energy consumption, emissions), the behavior of individual drivers (e.g., reaction to real-time traffic information), as well as their impact at an urban scale. Recent work has used this framework to address energy-efficient traffic management problems (Osorio and Nanduri 2012), as well as to propose novel formulations for traditional transportation problems that account for higher-order distributional information (Chen et al. 2012). Such formulations can improve, for instance, the reliability or the robustness of transportation systems.

The proposed approach of combining a functional metamodel with an analytical queueing network model is of interest to efficiently address SO problems for other networked systems, ranging from healthcare networks (Osorio and Bierlaire 2009a) to biological protein synthesis networks (Osorio and Bierlaire 2012).

Acknowledgments

The authors thank Dr. Emmanuel Bert and Prof. André-Gilles Dumont for providing the Lausanne simulation model. This research was partially supported by the Swiss National Science Foundation grant 205320-117581.

References

Alexandrov, N M, R M Lewis. 2001. An overview of first-order model management for engineering optimization. *Optimization and Engineering* **2** 413–430.

Alexandrov, N M, R M Lewis, C R Gumbert, L L Green, P A Newman. 2001. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft* **38**(6) 1093–1101.

Atkeson, C G, A W Moore, S Schaal. 1997. Locally weighted learning. *Artificial Intelligence Review* **11** 11–73.

Bandler, J W, S Koziel, K Madsen. 2006. Space mapping for engineering optimization. *SIAG/Optimization Views-and-News* **17**(1) 19–26.

Barton, R R, M Meckesheimer. 2006. Metamodel-based simulation optimization. S G Henderson, B L Nelson, eds., *Handbooks in operations research and management science: Simulation*, vol. 13, chap. 18. Elsevier, Amsterdam, 535–574.

Bastin, F, C Cirillo, P Toint. 2006. An adaptive Monte Carlo algorithm for computing mixed logit estimators. *Computational Management Science* **3** 55–79.

Bharadwaj, V, A J Kleywegt. 2008. Derivative-free trust region algorithms for stochastic optimization. Tech. rep., School of Industrial and Systems Engineering, Georgia Institute of Technology.

Booker, A J, J E Dennis, P D Frank, D B Serafini, V Torczon, M W Trosset. 1999. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* **17** 1–13.

Carter, R G. 1986. Multi-model algorithms for optimization. Ph.D. thesis, Rice University.

Chen, X, C Osorio, B F Santos. 2012. A simulation-based approach to reliable signal control. *Proceedings of the International Symposium on Transportation Network Reliability (INSTR).*

Coleman, T F, Y Li. 1994. On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming* **67**(2) 189–224.

Coleman, T F, Y Li. 1996. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* **6** 418–445.

Conn, A R, K Scheinberg, P L Toint. 1998. A derivative free optimization algorithm in practice. *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. St. Louis, MO, USA.

Conn, Andrew R, Nicholas I M Gould, Philippe L Toint. 2000. *Trust-region methods*. MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, USA.

Conn, Andrew R, Katya Scheinberg, Luis N Vicente. 2008a. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming* **111**(1-2) 141–172.

Conn, Andrew R, Katya Scheinberg, Luis N Vicente. 2008b. Geometry of sample sets in derivative-free optimization: Polynomial regression and underdetermined interpolation. *IMA Journal of Numerical Analysis* **28** 721–748.

Conn, Andrew R, Katya Scheinberg, Luis N Vicente. 2009a. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM Journal on Optimization* **20**(1) 387–415.

Conn, Andrew R, Katya Scheinberg, Luis N Vicente. 2009b. *Introduction to derivative-free optimization*. MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, USA.

Dumont, A G, E Bert. 2006. Simulation de l'agglomération Lausannoise SIMLO. Tech. rep., Laboratoire des voies de circulation, ENAC, Ecole Polytechnique Fédérale de Lausanne.

Kleijnen, Jack P C. 2008. Response surface methodology for constrained simulation optimization: An overview. *Simulation Modelling Practice and Theory* **16**(1) 50–64.

Kleijnen, Jack P C, Wim van Beers, Inneke van Nieuwenhuyse. 2010. Constrained optimization in expensive simulation: Novel approach. *European Journal of Operational Research* **202**(1) 164–174.

Kleywegt, A J, A Shapiro. 2001. *Stochastic Optimization*. Handbook of Industrial Engineering, John Wiley, New York, 2625–2650.

Kolda, Tamara, Robert Michael Lewis, Virginia Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review* **45**(3) 385–482.

Marti, K. 2008. *Stochastic optimization methods*. Springer, Berlin, Germany.

Mathworks, Inc. 2008. *Optimization Toolbox Version 4. User's Guide Matlab*. Natick, MA, USA.

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

Osorio, C. 2010. Mitigating network congestion: analytical models, optimization methods and their applications. Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne.

Osorio, C, M Bierlaire. 2009a. 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, M Bierlaire. 2009b. A surrogate model for traffic optimization of congested networks: an analytic queueing network approach. Tech. Rep. 090825, Transport and Mobility Laboratory, ENAC, Ecole Polytechnique Fédérale de Lausanne.

Osorio, C, M Bierlaire. 2010. A simulation-based optimization approach to perform urban traffic control. *Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN)*. Tromsø, Norway. URL `http://web.mit.edu/osorioc/Public/Papers/osoBie10_so.pdf`.

Osorio, C, M Bierlaire. 2012. A tractable analytical model for large-scale congested protein synthesis networks. *European Journal of Operational Research* **219**(3) 588–597.

Osorio, C, L Chong. 2012. An efficient simulation-based optimization algorithm for large-scale transportation problems. *Winter Simulation Conference (WSC)*. Berlin, Germany.

Osorio, C, G Flötteröd, M Bierlaire. 2011. Dynamic network loading: a stochastic differentiable model that derives link state distributions. *Transportation Research Part B* **45**(9) 1410–1423.

Osorio, C, K Nanduri. 2012. Energy-efficient traffic management: a microscopic simulation-based approach. *International Symposium on Dynamic Traffic Assignment (DTA)*. Martha's Vineyard, USA.

Powell, M J D. 2003. On trust region methods for unconstrained minimization without derivatives. *Mathematical Programming* **97**(3) 605–623.

Serafini, D B. 1998. A framework for managing models in nonlinear optimization of computationally expensive functions. Ph.D. thesis, Rice University.

Shapiro, A. 2000. *Probabilistic Constrained Optimization: Methodology and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 282–304.

Shapiro, A, D Dentcheva, A Ruszczynski. 2009. *Lectures on Stochastic Programming: Modeling and Theory*, chap. 5. Modern Probability and Statistics, SIAM, Philadelphia.

Søndergaard, J. 2003. Optimization using surrogate models - by the Space Mapping technique. Ph.D. thesis, Technical University of Denmark.

Stafford, R. 2006. *The Theory Behind the 'randfixedsum' Function*. Http://www.mathworks.com/matlabcentral/fileexchange/9700.

TSS. 2008. *AIMSUN 5.1 Microsimulator Users Manual*. Transport Simulation Systems.

VSS. 1992. *Norme Suisse SN 640837 Installations de feux de circulation; temps transitoires et temps minimaux*. Union des professionnels suisses de la route, VSS, Zurich.

Wild, Stefan M, Rommel G Regis, Christine A Shoemaker. 2008. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing* **30** 3197–3219.

Appendix

Analytical network model

The physical component of the metamodel is an analytical and differentiable urban traffic model. Each lane of an urban road network is modeled as a set of finite capacity queues. In the following notation the index i refers to a given queue. We refer the reader to Osorio and Bierlaire (2009b) and to Osorio and Bierlaire (2009a) for details.

γ_i	external arrival rate;
λ_i	total arrival rate;
μ_i	service rate;
$\tilde{\mu}_i$	unblocking rate;
μ_i^{eff}	effective service rate (accounts for both service and eventual blocking);
ρ_i	traffic intensity;
P_i^f	probability of being blocked at queue i;
k_i	upper bound of the queue length;
N_i	total number of vehicles in queue i;
$P(N_i = k_i)$	probability of queue i being full, also known as the blocking or spillback probability;
p_{ij}	transition probability from queue i to queue j;
x_j	green split of phase j;
s_i	saturation flow rate of lane i [veh/h];
$\mathscr{D}_i$	set of downstream queues of queue i;
$\mathcal{S}$	set of indices of the signalized queues;
$\mathscr{P}(i)$	set of phase indices of signalized queue i.

The queueing network model is formulated as follows.

$$\left\{\begin{array}{lr}
\lambda_i = \gamma_i + \dfrac{\sum_j p_{ji}\lambda_j(1-P(N_j=k_j))}{(1-P(N_i=k_i))}, & (15a)\\[2ex]
\dfrac{1}{\tilde{\mu}_i} = \displaystyle\sum_{j\in\mathscr{D}_i} \dfrac{\lambda_j(1-P(N_j=k_j))}{\lambda_i(1-P(N_i=k_i))\mu_j^{\text{eff}}}, & (15b)\\[2ex]
\dfrac{1}{\mu_i^{\text{eff}}} = \dfrac{1}{\mu_i} + P_i^f \dfrac{1}{\tilde{\mu}_i}, & (15c)\\[2ex]
P(N_i=k_i) = \dfrac{1-\rho_i}{1-\rho_i^{k_i+1}}\rho_i^{k_i}, & (15d)\\[2ex]
P_i^f = \displaystyle\sum_j p_{ij}P(N_j=k_j), & (15e)\\[2ex]
\rho_i = \dfrac{\lambda_i}{\mu_i^{\text{eff}}}, & (15f)\\[2ex]
\mu_i - \displaystyle\sum_{j\in\mathscr{P}(i)} x_j s_i = 0, \ \forall i \in \mathcal{S} & (15g)\\[2ex]
 & (15h)
\end{array}\right.$$

In this model, the exogenous parameters of a given queue are $\gamma_i, \mu_i, p_{ij}, k_i$ and s_i. All other parameters are endogenous. When used to solve a signal control problem (as in this paper), the capacity of the signalized lanes become endogenous, which makes the corresponding service rates, μ_i, endogenous. In that case, the exogenous parameters are γ_i, p_{ij}, k_i and s_i.

The form of the analytical approximation of the objective function provided by the queueing model, $T(x, y; q)$, for the signal control problem of this paper is given by: $(\sum_i E[N_i])/(\sum_i \gamma_i(1 - P(N_i = k_i)))$, where $E[N_i]$ represents the expected number of vehicles in lane i, and is given by: $E[N_i] = \rho_i \left(\frac{1}{1-\rho_i} - \frac{(k_i+1)\rho_i^{k_i}}{1-\rho_i^{k_i+1}} \right)$. Details on the derivation of these expressions can be found in Osorio (2010).

Model improvement step

Figure 7 analyses the number of model improvement steps (step 4 in Section 4.2) that were carried out when running the experiments of Figure 4. Each plot of Figure 7 considers one initial point. Each plot displays two boxplots, one for each method. The method labeled 1 (resp. 2) is the proposed approach (resp. polynomial approach). Each boxplot summarizes 10 points, which are the total number of model improvement steps for each of the 10 runs of the algorithm (each run of the algorithm allowed for 150 simulations). For the first initial point (left plot), across all 10 runs of the algorithm, the proposed method requires fewer model improvement points than the polynomial. There is a similar but less marked trend for the second initial point (right plot).

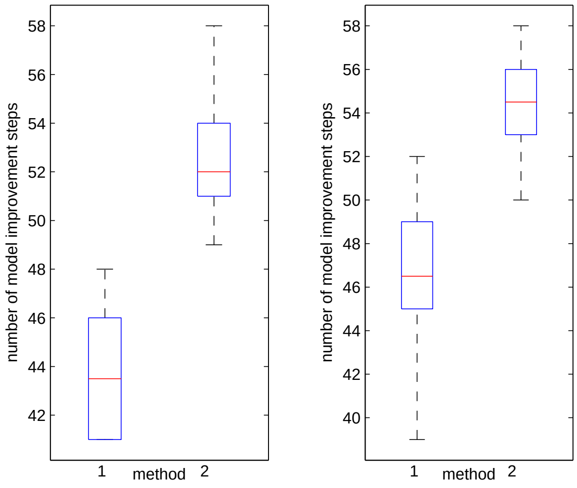


Figure 7 Number of model improvement steps carried out for each method and each initial point.