

# A simulation-based optimization framework for urban traffic control \*

Carolina Osorio<sup>†</sup>

Michel Bierlaire<sup>‡</sup>

December 3, 2010

Report TRANSP-OR 101203  
Transport and Mobility Laboratory  
School of Architecture, Civil and Environmental Engineering  
Ecole Polytechnique Fédérale de Lausanne  
[transp-or.epfl.ch](http://transp-or.epfl.ch)

---

\*This research was partially supported by the Swiss National Science Foundation grant 205320-117581

<sup>†</sup>Civil and Environmental Engineering Department, Massachusetts Institute of Technology, 1-232, Cambridge, Massachusetts 02139, USA, [osorioc@mit.edu](mailto:osorioc@mit.edu)

<sup>‡</sup>Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, [michel.bierlaire@epfl.ch](mailto:michel.bierlaire@epfl.ch)

# Abstract

Microscopic urban traffic simulators embed numerous behavioral models that describe individual driver decisions (e.g. route choice, departure time choice, response to information) to capture the interactions between complex travel demand patterns and network supply phenomena. These disaggregate models make them detailed tools appropriate to perform scenario-based or sensitivity analysis. Nevertheless, this detail leads to nonlinear, stochastic and evaluation-expensive models. Their integration within an optimization framework to use them to identify traffic management schemes remains a difficult and challenging task. In order to perform computationally efficient simulation-based optimization for congested urban networks, information from the simulation tool should be combined with information from a network model that analytically captures the structure of the underlying problem.

This paper presents a metamodel that combines the information from a microscopic traffic simulation model with an analytical queueing 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. We integrate this metamodel within a derivative-free trust region optimization framework.

We evaluate the performance of this method with a fixed-time signal control problem for subnetworks of the city of Lausanne, considering different demand scenarios and tight computational budgets. The performance of the signal plans derived by the proposed method is compared to that of other methods, including an existing signal plan for the city of Lausanne.

The method leads to well-performing signal plans for both small and larger samples. It leads to reduced, as well as more reliable, average travel times. This combination of a general-purpose metamodel, needed for its asymptotic properties, and an analytical network model, which provides tractable analytical information, leads to a simulation-based optimization framework that is computationally efficient and suitable for complex problems with tight computational budgets.

Keywords: *Traffic control, Simulation-based Optimization, Metamodel, Queueing*

## 1 Introduction

Microscopic urban simulators capture in detail the behavior of drivers as well as their interaction with the network infrastructure. They can provide accurate network performance estimates in the context of scenario-based analysis or sensitivity analysis. They are therefore often used to evaluate traffic management schemes. Nevertheless, using them to derive appropriate management schemes (i.e. to perform optimization) is a complex task.

An optimal traffic management scheme can be formulated as:

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

where the objective is to minimize the expected value of a suitable network performance measure,  $f$ . This performance measure is a function of a decision or control vector  $x$ , endogenous variables  $z$  and exogenous parameters  $p$ . The feasible space  $\Omega$  consists of a set of constraints that link  $x$  to  $z$ ,  $p$  and  $f$ .

For instance, a traffic signal control problem can take  $f$  as the average 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  $p$ , while  $z$  accounts, for instance, for the capacities of the signalized lanes.

The various disaggregate traffic models embedded within the simulator make it a detailed model, but lead to stochastic nonlinear objective functions containing potentially several local minima. These objective functions have no available closed-form; we can only derive estimates for  $E[f(x, z; p)]$ . Additionally, evaluating these estimates is computationally expensive because they involve running numerous replications. As a nonlinear, stochastic and computationally-expensive problem, it is complex to address.

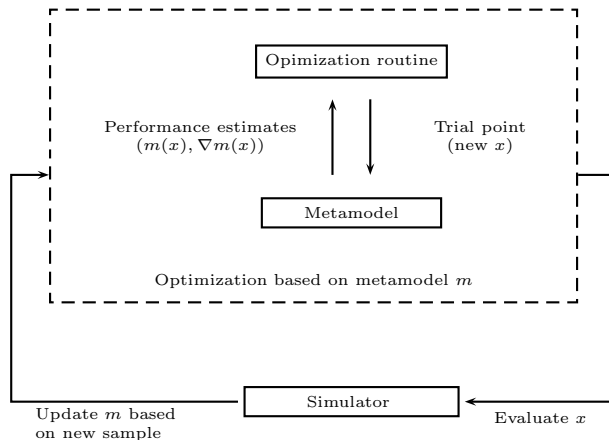


Figure 1: Metamodel simulation-based optimization methods. Adapted from Alexandrov et al. (1999).

## Metamodel methods

As is detailed in Section 2, one approach to perform simulation-based optimization (SO) is to build an analytical model of the objective function based on a sample of simulated observations. This analytical model is referred to as a metamodel or a surrogate model. This family of approaches is known as metamodel methods. Once the metamodel is fitted, it is used to perform optimization.

This approach is depicted in Figure 1. The metamodel is denoted as  $m$ . This figure illustrates the two main steps of metamodel methods. Firstly, the metamodel is constructed based on a sample of simulated observations. Secondly, once the metamodel  $m$  has been fitted, the optimization process derives a trial point based on the predictions and properties of  $m$ .

These steps can then be iterated as follows. For a given sample the metamodel is fitted, the optimization problem is solved, deriving a trial point. Then the performance of this trial point can be evaluated by the simulator, which leads to a new observation. As new observations become available the accuracy of the metamodel can be improved, leading ultimately to better trial points.

Metamodels are typically deterministic functions that are cheaper to evaluate. By replacing the stochastic response of the simulation by a deterministic function, deterministic optimization techniques can be used. Furthermore, by using metamodels that are cheap 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.

The most common metamodels used to perform simulation-based optimization are general-purpose models (e.g. polynomials, splines) that can be used to approximate any objective function, but capture little information about the structure of the underlying problem. Furthermore, they require a large initial sample to be fitted, and are thus inappropriate for applications with a tight computational budget.

The SO problems that arise in practice are challenging: they are constrained stochastic problems with nonlinear objective functions that are computationally expensive to evaluate. The key to developing SO methods that can address complex problems is achieving computational efficiency, i.e. devising frameworks that lead to improved performance measures with few simulation runs.

In order to perform SO for congested urban networks given a tight computational budget, we propose to combine the generic metamodels with network models that analytically capture the structure of the underlying problem. This approach combines detail and realism, provided by the simulation model, with tractability, provided by the analytical network model. This combination improves the short-term behavior

of SO algorithms and, therefore, achieves computational efficiency.

## Derivative-free optimization

Both the noise inherent in simulation outputs along with their high computational cost, makes the accurate estimation of derivatives an expensive and difficult task. When derivative information is either unavailable, available at a high cost or unreliable, then derivative-free (DF) optimization methods are an appropriate and common approach to tackle such problems.

Given the lack of derivative information, DF methods typically perform less well than their derivative-based counterparts. In particular, the scale of the problems that can be efficiently tackled is substantially reduced. Currently, the most recent DF methods are limited to around 20 variables for unconstrained problems and their convergence is typically rather slow (Conn et al., 2009b), not to mention their limitations in the presence of constraints. By using a surrogate that integrates structural information, we expect to be able to address both larger and constrained problems more efficiently.

Furthermore, DF methods are often used when function evaluations are computationally expensive. They therefore attempt to identify trial points with improved performance given a fixed, and typically tight, computational budget. We expect the added structural information of the metamodel to allow the identification of good trial points even for tight computational budgets. In this paper, we evaluate the performance of the proposed method considering scenarios with tight computational budgets and assuming that there are initially no observations available.

This paper is structured as follows. We present a literature review of the SO methods focusing on metamodel methods, and a review of the existing optimization algorithms that allow for arbitrary metamodels (Section 2). We present both the metamodel (Section 3) and the optimization algorithm (Section 4) that are used. 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 two types of approaches to address SO problems. Firstly, there is the family of direct-search methods, which rely only on objective function evaluations and do not resort to any direct or indirect (i.e. explicit or implicit) derivative approximation or model building. In these methods, the search is based on a pre-determined geometric pattern or grid. For instance, formulations of the popular simplex algorithm (Nelder and Mead, 1965) for SO problems have been proposed (Barton and Ivey, 1996). Reviews of direct methods are given in Fu et al. (2005) and in Andradóttir (1998).

Secondly, there are the methods that do use gradient information. In this paper, we focus on this second family of methods. Barton and Meckesheimer (2006) review such methods and classify them into direct gradient and metamodel methods. Direct gradient methods estimate the gradient of the simulation response, whereas metamodel methods use an indirect-gradient approach by computing the gradient of the metamodel, which is a deterministic function.

Direct gradient methods are inspired from their deterministic counterpart: searching along gradient descent directions. These algorithms usually take the form of stochastic approximation (SA) techniques, which stem from the seminal paper of Robbins and Monro (1951). SA techniques assume either that there are direct gradient observations available or approximate the gradient via objective function observations. Combining SA with finite difference estimation of the gradient leads to the Kiefer-Wolfowitz algorithm (Kiefer and Wolfowitz, 1952). SA methods combined with simultaneous perturbation gradient estimation

have also been proposed (Spall, 2003; Spall and Chin, 1997). These methods are suitable when accurate gradient estimations can be obtained.

There have been significant advances and novel approaches for gradient estimation (Fu, 2006; Fu et al., 2005). Nevertheless, methods that rely on direct derivative information often require more function evaluations, and are therefore inappropriate for applications with a limited computational budget. Additionally, by resorting to a metamodeling approach the stochastic response of the simulation is replaced by a deterministic metamodel response function, such that deterministic optimization techniques can be used. This paper focuses on metamodel methods.

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, whose 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 approach is the use of low-order polynomials (e.g. linear or quadratic). Quadratic polynomials are used as surrogates in most trust region methods (Conn et al., 2000), as well as 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 (Ouvray and Bierlaire, 2009; Wild et al., 2008) and Kriging surrogates (Kleijnen et al., 2010, Booker et al., 1999) have also been proposed.

The existing metamodels consist of either physical or functional components. The metamodel proposed here goes beyond existing approaches by combining both a physical and a functional component. It combines an analytical network traffic model with a quadratic polynomial. The physical component is captured by the traffic model, whose parameters have a structural interpretation. For a given problem, the traffic model will yield a different functional form for the objective function. The functional component is captured by the general purpose quadratic polynomial, which ensures asymptotic metamodel properties (full linearity) needed for convergence analysis, and in particular asymptotic global convergence for unconstrained problems.

In order to integrate the proposed metamodel within an existing optimization method, we review the algorithms that allow for an arbitrary metamodel. These methods are called multi-model or hybrid methods. They 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).

A trust region optimization framework for unconstrained problems allowing for multiple models was proposed by Carter (1986) (see references herein for previous multi-model frameworks). His work analyses the theoretical properties and derives a global convergence theory for several types of multi-model algorithms. It allows for nonquadratic models as long as at least one model is a standard quadratic with uniformly bounded curvature.

The Approximation and Model Management Optimization/Framework (AMMO or AMMF) is a trust region framework for generating and managing a sequence of metamodels. There are several versions of the algorithm: for unconstrained problems (Alexandrov et al., 1998), bound constrained (Alexandrov et al., 2000), inequality constrained (Alexandrov et al., 1999) and generally constrained problems (Alexandrov et al., 2001). Although no restrictions are imposed on the type of surrogates allowed, it is a first-order method that requires that the model and the objective function, as well as their first-order derivatives,

coincide at each major (or accepted) iterate. Thus the metamodel must always behave as a first-order Taylor series approximation. This is a strong restriction if the function is noisy and expensive to evaluate.

The Surrogate-Management Framework (SMF) proposed by Booker et al. (1999) is a derivative-free method for bound constrained problems. It is based on a direct search technique called pattern search. Since direct search techniques typically require many function evaluations, they use a surrogate model of the objective function to improve the performance of the algorithm. The surrogate model used is an interpolated Kriging model.

The Space Mapping (SM) technique and its many versions (Bandler et al., 2006; Bandler et al., 2004) is a simulation-based optimization technique that uses two metamodels: a fine and a coarse model. Both models are often simulation-based. The coarse model is constructed based on a transformation of the endogenous variables (“space mapping”) that minimizes the error for a sampled set of high-fidelity response values. Nevertheless, SM relies on the assumption that via a transformation of the endogenous variables the coarse model will exhibit the physical/mathematical properties of the fine model (Alexandrov and Lewis, 2001) and as Bandler et al. (2004) mention “the required interaction between coarse model, fine model and optimization tools makes SM difficult to automate within existing simulators”. Alexandrov and Lewis (2001) give a comparison of the AMMO, the SMF and the SM methods.

Conn et al. (2009a) recently proposed a trust region derivative-free framework for unconstrained problems. This framework allows for arbitrary metamodels and makes no assumption on how these metamodels are fitted (interpolation or regression). To ensure global convergence, a model improvement algorithm guarantees that the models achieve a uniform local behavior (i.e. satisfy Taylor-type bounds) within a finite number of steps.

Derivative-free (DF) methods do not require nor do they explicitly approximate derivatives. Resorting to a DF algorithm, rather than to first or second order algorithms, is appropriate when the derivatives are difficult to obtain, unreliable or computationally expensive to evaluate. This is the case for noisy problems, for problems where the evaluation of the objective function is computationally expensive, or for problems where the simulation source code is unavailable (Moré and Wild, 2009). In the field of urban transportation, most simulators fall into all three of these categories. Thus we opt for a DF approach.

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 Hessians” (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.

The framework proposed by Conn et al. (2009a), as a derivative-free TR method that allows for arbitrary models and does not impose interpolation, is therefore particularly appealing. We integrate the proposed metamodel within this framework.

### 3 Metamodel

The metamodel combines information from two 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 the microscopic traffic simulator AIMSUN (TSS, 2008b). It 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. For a detailed description of the behavioral models see TSS (2008a).

For a given sample of observations, let  $x^i$  denote the  $i^{\text{th}}$  point in the sample with corresponding endogenous simulation variables  $z^i$ , and realization  $\hat{f}(x^i, z^i; p)$  of the performance measure  $f(x, z; p)$  (presented in Equation (1)).

**Analytical network model.** The analytical network model is the physical component of the metamodel: its parameters should have a physical or structural interpretation. It should be a good, yet less detailed, approximation of the underlying system, and should analytically capture the structure of the underlying problem.

The purpose of an analytical network model in this framework is also to improve the computational tractability of the optimization method, by overcoming the limitations of the simulation model, i.e. provide closed-form continuous expressions for the performance measures and their first-order derivatives.

The analytical model chosen for this framework satisfies these criteria. It is an analytical urban traffic model formulated in Osorio and Bierlaire (2009b). Alternatively, for large-scale networks the model of Osorio (2010) can also be used.

In Osorio and Bierlaire (2009b), this queueing model is formulated for urban traffic, calibrated, and used both to solve a fixed-time signal control problem and to investigate the added value of analytically accounting for between-queue interactions when addressing signal control problems. In this paper, this model is used to formulate a metamodel to perform simulation-based optimization.

We briefly recall the main components of this model. The model uses the finite capacity queueing theory (FCQT) notion of *blocking* to describe where congestion arises and how it propagates throughout the network. By resorting to FCQT, it 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. 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. The model yields detailed queue length distributions. For a given decision vector  $x$ , the network model yields the objective function  $T(x, y; q)$ , which is a deterministic approximation of  $E[f(x, z; p)]$  (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;
- $\hat{f}$  sample of performance measure derived by the simulation model;
- $y$  endogenous queueing model variables;
- $z$  endogenous simulation 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 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 approach is to use a quadratic polynomial. The proposed metamodel combines a quadratic polynomial with a deterministic approximation of the objective function, provided by the analytical network model. The functional form of  $m$  is:

$$m(x, y; \alpha, \beta, q) = \alpha T(x, y; q) + \phi(x; \beta), \tag{2}$$

where  $\phi$  is a quadratic polynomial in  $x$ ,  $\alpha$  and  $\beta$  are parameters of the metamodel.

The polynomial  $\phi$  is quadratic in  $x$  with a diagonal second derivative matrix. This choice is based on existing numerical experiments for derivative-free TR methods, which show that these types of quadratic polynomials are often more efficient than full quadratic polynomials (Powell, 2003).

$$\phi(x; \beta) = \beta_1 + \sum_{j=1}^d \beta_{j+1} x_j + \sum_{j=1}^d \beta_{j+d+1} x_j^2, \quad (3)$$

where  $d$  is the dimension of  $x$ ,  $x_j$  and  $\beta_j$  are the  $j^{\text{th}}$  components of  $x$  and  $\beta$ , respectively.

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, the simulator and the queueing model are evaluated at one (in some cases two) point(s). The metamodel is fitted using the observations obtained at the current iteration, as well as all observations collected at previous iterations.

The parameters  $\beta$  and  $\alpha$  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 to each observation. The least squares problem is formulated as follows.

$$\min_{\alpha, \beta} \sum_{i=1}^{n_k} \left\{ w_{ki} \left( \hat{f}(x^i, z^i; p) - m(x^i, y^i; \alpha, \beta, q) \right) \right\}^2 + (w_0 \cdot (\alpha - 1))^2 + \sum_{i=1}^{2d+1} (w_0 \cdot \beta_i)^2, \quad (4)$$

where  $x^i$  represents the  $i^{\text{th}}$  point in the sample, with corresponding endogenous simulation variables  $z^i$ , endogenous queueing model variables  $y^i$  and observation  $\hat{f}(x^i, z^i; p)$ . The sample size at iteration  $k$  is  $n_k$ . The weight associated at iteration  $k$  to the  $i^{\text{th}}$  observation is denoted  $w_{ki}$ . The parameter  $w_0$  represents a fixed weight.

The first squared term of Equation (4) represents the weighted distance between the simulated observations 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 of full rank. The initial values used here (one for  $\alpha$  and zero for  $\beta$ ) 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 observations.

The weights,  $w_{ki}$ , capture the importance of each point with regards to the current iterate. The work of Atkeson et al. (1997) gives a survey of weight functions and analyzes their theoretical properties. We use what is known as the *inverse distance* weight function along with the Euclidean distance. This leads to the following weight parameters:

$$w_{ki} = \frac{1}{1 + \|x_k - x^i\|_2}, \quad (5)$$

where  $x_k$  is the current iterate, and  $x^i$  is the  $i^{\text{th}}$  sample 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* (The Mathworks, 2008).



## 4 Optimization algorithm

For an introduction to trust region (TR) methods, we refer the reader to Conn et al. (2000). They summarize the main steps of a TR method in the *Basic trust region algorithm*. The method proposed by Conn et al. (2009a) builds upon the *Basic TR algorithm* by adding two additional steps: a model improvement step and a criticality step. For a detailed description, see Conn et al. (2009a). In this section, we first present the algorithm, we then provide details on the purpose of each step and on its implementation.

### 4.1 Algorithm

#### 0. Initialization.

Define for a given iteration  $k$

- $m_k$  as the metamodel,
- $x_k$  as the iterate,
- $\Delta_k$  as the trust region radius,
- $\alpha_k$  and  $\beta_k$  as the metamodel parameters,
- $\nu_k$  as the vector of parameters of  $m_k$ ,  $\nu_k = (\alpha_k, \beta_k)$ ,
- $n_k$  as the sample size,
- $u_k$  as the number of successive trial points rejected,
- $\epsilon_k$  as the measure of stationarity evaluated at  $x_k$ .

Set

- an initial point  $x_0$ ,
- an upper bound for the trust region radius  $\Delta_{max} > 0$ ,
- an initial trust region radius  $\Delta_0 \in (0, \Delta_{max}]$ ,
- the parameters  $\eta_1, \gamma, \gamma_{inc}, \epsilon_c, \bar{\tau}, \bar{d}, \bar{u}$  such that
  - $0 < \eta_1 < 1$ ,  
 $\eta_1$  determines the threshold for the acceptance of trial points,
  - $0 < \gamma < 1 < \gamma_{inc}$ ,  
 $\gamma$  and  $\gamma_{inc}$  determine, respectively, the decrease and increase factors for the trust region radius,
  - $\epsilon_c > 0$ ,  
 $\epsilon_c$  sets the threshold for ensuring full linearity for small measures of stationarity,
  - $0 < \bar{\tau} < 1$ ,  
 $\bar{\tau}$  determines the threshold for sampling new points to improve the models accuracy,
  - $0 < \bar{d} < \Delta_{max}$ ,  
 $\bar{d}$  determines the trust region radius threshold,
  - $\bar{u} \in \mathbb{N}^*$ ,  
 $\bar{u}$  determines the threshold for consecutive unsuccessful trial points without decreasing the trust region radius.
- the maximum number of function evaluations (i.e. simulation runs) permitted  $n_{max}$ .
- Compute  $T$  and  $\hat{f}$  at  $x_0$ , fit an initial model  $m_0$ , and compute  $\nu_0$ .
- Set  $k = 0, n_0 = 1, u_0 = 0$ .

1. **Criticality step.** If  $\epsilon_k \leq \epsilon_c$ , then switch to *conservative mode* (detailed in Section 4.2).
2. **Step calculation.** Compute a step  $s_k$  that reduces the model  $m_k$  and such that  $x_k + s_k$  is in the trust region (i.e. approximately solve the TR subproblem).<sup>1</sup>
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 sample set ( $n_k = n_k + 1$ ), 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 sampling a new point  $x$ . The sampling distribution is defined in Section 4.2. Evaluate  $T$  and  $\hat{f}$  at  $x$ . Include this point in the sample set ( $n_k = n_k + 1$ ). Update  $m_{k+1}$ .

5. **Trust region radius update.**

- If  $\rho_k > \eta_1$ , then increase the trust region radius:  
 $\Delta_{k+1} = \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}.$
- Otherwise,
  - if  $u_k \geq \bar{u}$ , then reduce the trust region radius:  
 $\Delta_{k+1} = \max\{\gamma\Delta_k, \leq \bar{d}\}, u_k = 0,$
  - otherwise,  $\Delta_{k+1} = \Delta_k.$
- If  $\Delta_{k+1} \leq \bar{d}$ , then switch to *conservative mode*.

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

Set  $k = k + 1.$

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

---

<sup>1</sup>The Basic TR algorithm requires *sufficient reduction* of the model, the Conn et al. (2009a) algorithm relaxes this assumption and ensures convergence when accepting trial points that provide simple decrease.

## 4.2 Algorithmic details

A given iteration  $k$  of the algorithm considers a metamodel  $m_k$ , an iterate  $x_k$  and a TR radius  $\Delta_k$ . Each iteration consists of 5 steps.

**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 improved so that its stationarity measure can be trusted. The model is then said to be *certifiably fully linear* (i.e. it satisfies Taylor-type bounds). We assume throughout that we cannot certify whether the model is fully linear. If at a given iteration, the measure of stationarity does go under the criticality threshold then a purely quadratic metamodel along with an appropriate sampling strategy (e.g. Monte Carlo, Quasi-Monte Carlo) can be used in order to obtain an accurate gradient estimate and to certify full linearity. This is denoted as the *conservative mode* in the algorithm.

**Step calculation.** Approximately solve the TR subproblem to yield a trial point. 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.** Either certify that  $m_k$  is *fully linear* (i.e. satisfies Taylor-type bounds) in the TR or attempt to improve the accuracy of the metamodel.

At each iteration, we run the simulator at the trial point,  $x_k + s_k$  (Step 3 of the algorithm). In order to diversify the set of sampled points, we may sample points other than the trial points. This step attempts to improve the accuracy of the model, by improving 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., 2009b)). We do so if the condition  $\tau_{k+1} < \bar{\tau}$  is satisfied. 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. Since we assume throughout that we cannot certify whether the model is *fully linear*, we reduce the TR radius after  $\bar{u}$  successive trial points have been rejected. If the TR radius reaches a lower bound  $\bar{d}$ , then a quadratic polynomial with an appropriate sampling strategy is used, and as mentioned previously, we can ensure that within a uniformly bounded number of sampling steps the model will be *fully linear*.

**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.** As detailed by Conn et al. (2009b), DF TR methods are a relatively recent topic. The algorithms developed so far are derived based on sound theoretical properties that lead to a solid global convergence theory, but they are mostly formulated for unconstrained problems. The Conn et al. (2009a) algorithm ensures global convergence for unconstrained problems.

Unfortunately, the optimization problems encountered in practice are rarely unconstrained. Conn et al. (2009b) review constrained DF algorithms, and confirm that for constrained problems “currently, there is no convergence theory developed for TR interpolation-based methods”, not to mention TR methods that allow for regression models.

Conn et al. (1998) propose a method to solve problems with general constraints using an unconstrained TR algorithm. The traffic management problems that we are interested in solving fall into the category of what they denote as *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 the different formulations, as well as the definitions of the traffic signal terms used hereafter, is given in Osorio and Bierlaire (2009b). We consider the same problem as in Osorio and Bierlaire (2009b), i.e. 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. In other words, the set of lanes associated with each stage as well as the sequence of stages are both known. 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;
- $\mathcal{I}$  set of intersection indices;
- $\mathcal{P}_I(i)$  set of phase indices of intersection  $i$ .

The problem is traditionally formulated as follows:

$$\min_{x,z} E[f(x, z; p)] \tag{6}$$

subject to

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

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

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

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

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

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

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

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

where  $x_k$  is the current iterate,  $\Delta_k$  is the current trust region radius,  $\alpha_k$  and  $\beta_k$  are the current metamodel parameters, and  $h_2$  denotes the queueing model. Equation (11) represents the queueing model, which consists of a system of nonlinear equations. These equations are given explicitly in Osorio and Bierlaire (2009b) (Equations (1)-(6),(9) in that paper). The analytical form of  $T$  is also detailed in Osorio and Bierlaire (2009b). Constraint (12) is the TR constraint. It uses the Euclidean norm (Conn et al., 2009a). 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.

For a problem with  $\ell$  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  $4\ell$  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*, which 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  $10^{-2}$ . The measure of stationarity,  $\epsilon_k$ , mentioned in the algorithm (Section 4.1) corresponds to the norm of the derivative of the Lagrangian function of this problem with regards to the endogenous variables.

### 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,  $\phi$ , 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)). To reduce the correlation between the explanatory variables of the metamodel, 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, 2008b). 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.

To refer to the different metamodels we use the notation of Equation (2). In both sections, we compare the performance of the signal plans derived by the use of three models:

- the proposed metamodel,  $m$ ,
- a quadratic polynomial with diagonal second derivative matrix, (i.e. the metamodel consists of  $\phi$ ),
- the queueing model,  $T$ . This is the procedure proposed in Osorio and Bierlaire (2009b). Namely, this procedure uses 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 (Figure 2). Demand arises at the nine centroids nearest to these two intersections. The simulation setup considers a 20 minute scenario, 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 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

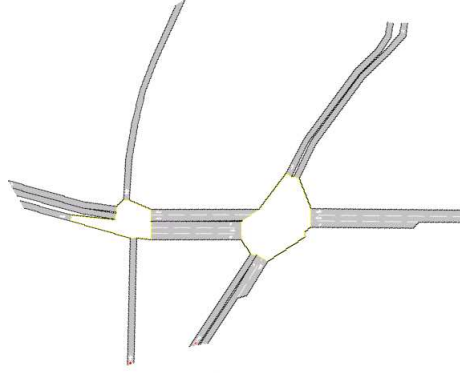


Figure 2: Lausanne city subnetwork with 2 controlled intersections.

performance for increasing sample sizes. 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 sample size, we consider the 10 signal plans derived and evaluate their performance by running 50 replications of the simulation model. All simulations are preceded by a 15 minute warm-up period. 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 3 considers a set of four plots. Each plot displays results for the initial plan, and the plans derived by both methods for a given sample size. Each plot considers a different sample size. Plots 3(a)-3(d) consider, respectively, sample sizes 10, 50, 100 and 150. The cdf labelled  $x_0$  denotes the cdf of the initial random plan. The numbers denote the corresponding sample sizes, e.g. the cdf labeled “ $A\phi$  10” corresponds to the signal plans proposed by  $A\phi$  with a sample of size 10.

For all four sample sizes,  $Am$  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 plans proposed by  $A\phi$  do not provide improvement when compared to the initial plan.

Figure 4 considers the same experiments with a different initial plan. In this case, the conclusions for sample size 10 (plot 4(a)) remain the same as before. As of sample size 50 (plot 4(b)),  $A\phi$  leads to signal plans with improved performance compared to the initial plan. The plans of  $Am$  outperform those of  $A\phi$  and the initial plans. For sample sizes 100 and 150 (plots 4(c) and 4(d)), the plans derived by  $Am$  and  $A\phi$  have similar performance. Once again,  $Am$  leads to variance reduction. These four plots also illustrate how the performance of the plans derived by  $A\phi$  improves with the sample size.

In all these cases with tight computational budgets,  $Am$  leads to signal plans with good performance and improved performance compared to the plans derived by  $A\phi$  and to the initial plan. It also leads to a reduction in the variance of the average travel times across signal plans and replications.

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 5 considers the signal plans derived by  $Am$  and  $A\phi$  at sample sizes 10, 20, 30 and 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 at sample sizes 10, 20, 30 and 40. At a sample size of 10, the plans of both  $Am$  and  $A\phi$  lead to improved average travel times, when compared to the initial plan. As the sample size increases,  $A\phi$  leads to plans with

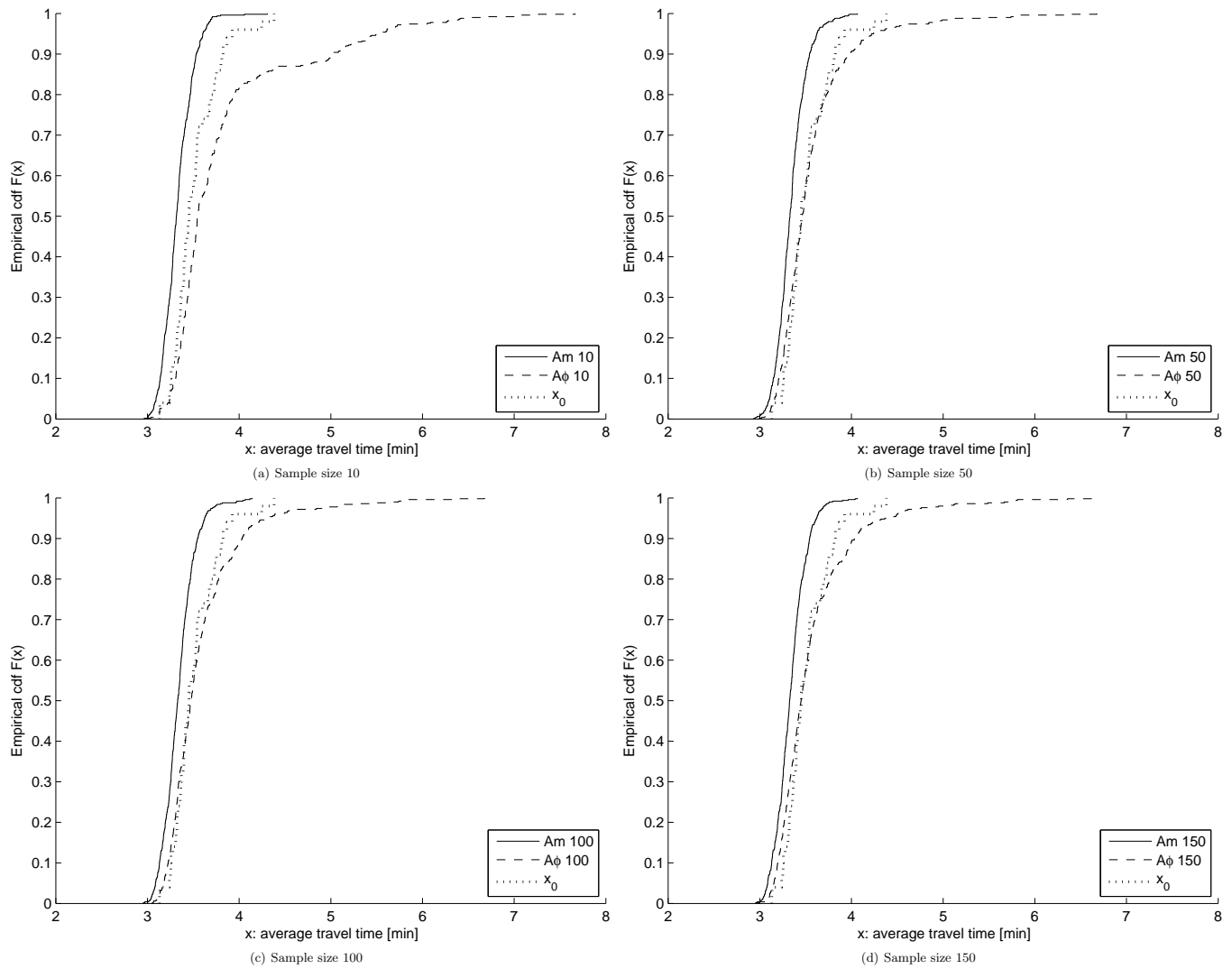


Figure 3: Empirical cdf's of the average travel times considering an initial random signal plan.



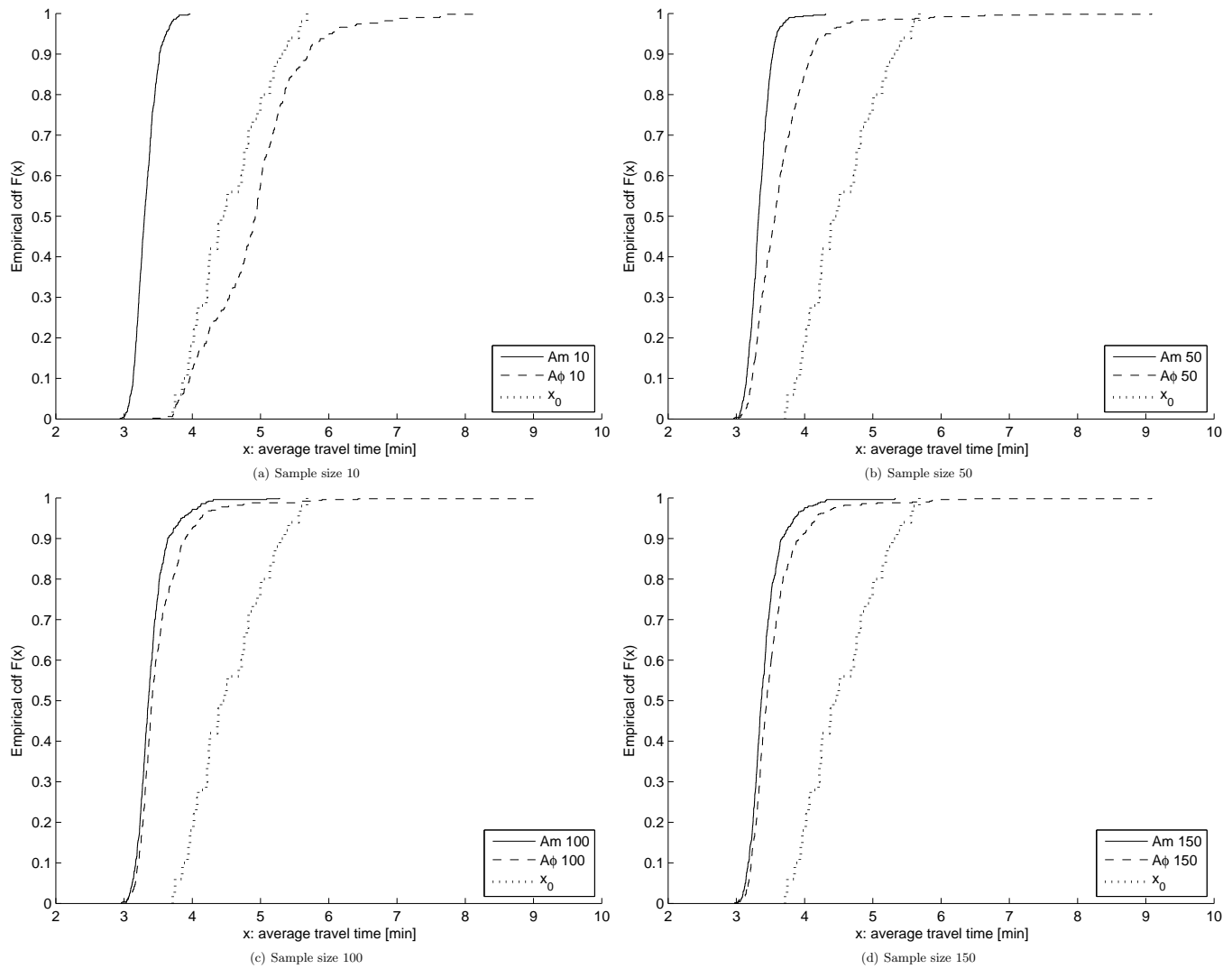


Figure 4: Empirical cdf's of the average travel times considering an initial random signal plan.

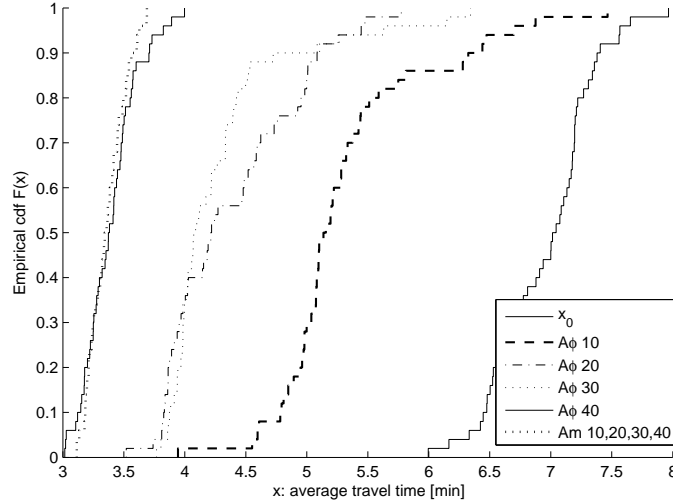


Figure 5: Empirical cdf's of the average travel times considering an initial random signal plan and evaluating the performance as the sample size increases from 10 to 40.

improved performance. At sample size 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 6 considers the signal plans proposed at sample sizes 40, 50, 100, 250, 500 and 750. For each signal plan, the cdf of the average travel times over the 50 replications are displayed. This figure shows that the performance of the plans is similar for sample sizes larger than 40. The signal plans proposed by  $A\phi$  at sample sizes 500 and 750 have a large variance in the average travel times.

Figure 7 considers the signal plans proposed by  $Am$  and  $A\phi$  at sample size 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.

We consider the full sample (750 observations) and test whether the metamodel parameters of the proposed model are different from zero. To do so we perform a t-test. The null hypothesis assumes that the given parameter is equal to zero, whereas the alternative hypothesis assumes it differs. We set the confidence level of the test to 0.05. The corresponding critical value is 1.96. Recall that there are a total of 23 parameters. Nine are statistically different from zero. These 9 parameters concern 5 linear terms, 2 quadratic terms, the queueing model parameter ( $\alpha$ ) and the intercept ( $\beta_1$ ). This indicates that the proposed metamodel indeed captures information about the relationship between the observed average travel times and the phase variables.

The results of this section indicate that for small to moderate sample sizes (compared to the dimension of the decision vector) the structural information provided by the proposed metamodel, allows  $Am$  to identify well-performing signal plans. As the sample size increases, the polynomial metamodel of  $A\phi$  improves its accuracy, and derives with a moderate sample size signal plans with similar performance to those of  $Am$ . The proposed method,  $Am$ , derives signal plans with reduced variance in the average travel times compared to both the initial plans and the plans proposed by  $A\phi$ . By comparing the plans of  $Am$  to those of  $AT$ , the results indicate that the simulated observations indeed improve the accuracy of the model, leading to signal plans that reduce the average travel times.

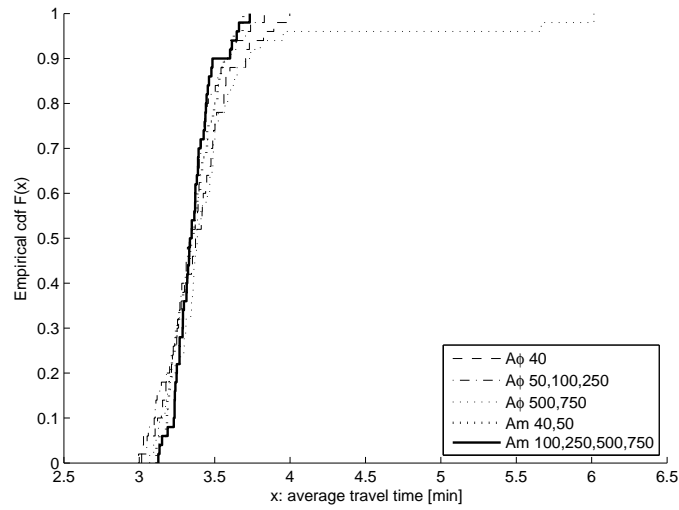


Figure 6: Empirical cdf's of the average travel times considering an initial random signal plan and evaluating the performance as the sample size increases from 40 to 750.

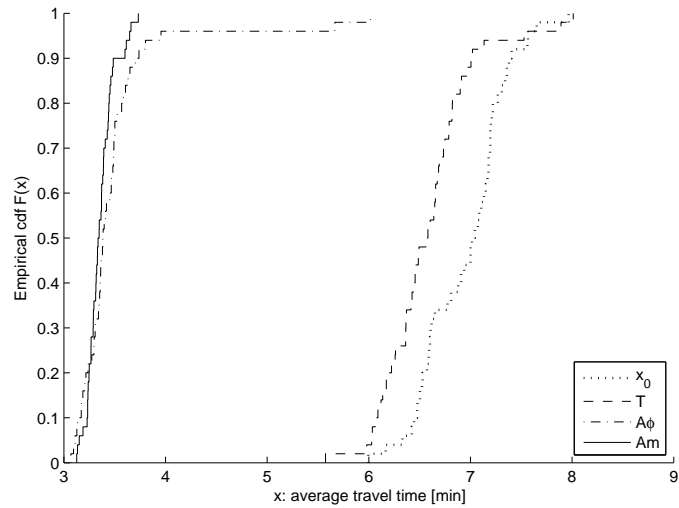


Figure 7: Empirical cdf's of the average travel times considering an initial random signal plan and evaluating the performance at sample size 750.

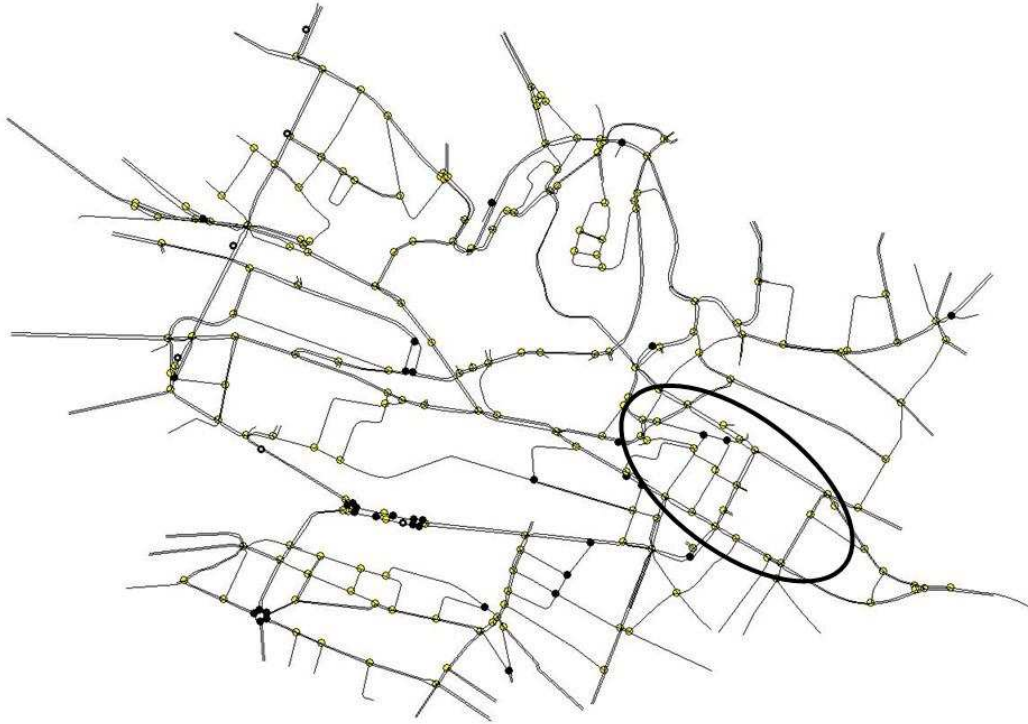


Figure 8: Lausanne city road network model.

## 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 8. The considered subnetwork is located in the city center, and is delimited in Figure 8 by an oval. The subnetwork is displayed in detail in Figure 9. It is also presented in detail in Osorio and Bierlaire (2009b). The considered scenario consists 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, 1378 function evaluations would be necessary to fit the full polynomial. This is because for a problem with  $n$  decision variables  $(n + 1)(n + 2)/2$  suitably sampled points (i.e. well posed (Conn et al., 2000; Conn et al., 2009b)) are necessary to fit the full quadratic.

We consider a tight computational budget, which is defined as a maximum number of simulation runs that can be carried out, and no initial observation available. The computational budget is set to 150 runs.

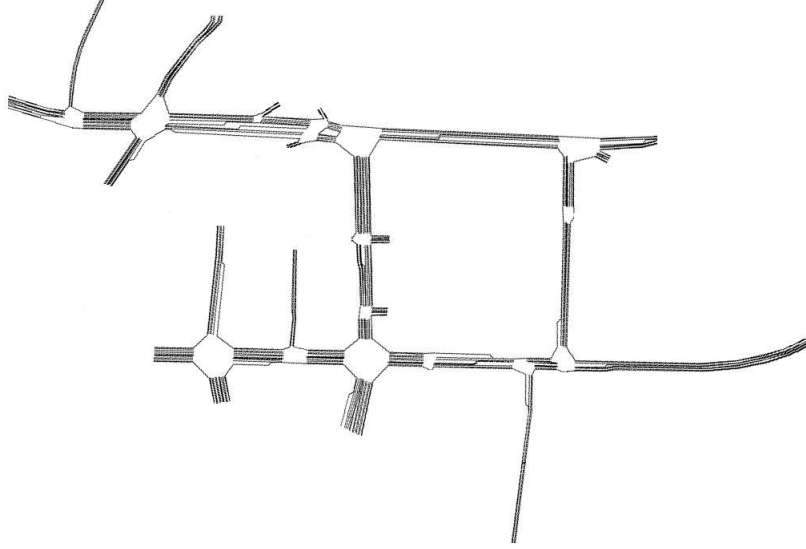


Figure 9: Subnetwork of the Lausanne city center.

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 10 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.

The same experiment is carried out with a different initial point. Figure 11 presents the corresponding results. Here, the same conclusions hold. The proposed method,  $Am$ , systematically yields signal plans with improved travel times, which is not the case for  $A\phi$ .

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 of Figure 12 considers one method, and displays the cdf's of the initial plan, and of the plans derived at sample sizes 10, 50, 100 and 150. For all sample sizes,  $Am$  (left plot) yields signal plans with improved distributions. The distributions also improve as the sample size increases.  $A\phi$  (right plot) yields the same plan as the initial plan for sample size 10, i.e. for all 10 runs of the algorithm, as of sample size 10 all trial points proposed by  $A\phi$  are rejected. As of sample size 50,  $A\phi$  identifies signals plans with improved performance compared to the initial plan. The distributions also improve as the sample size increases.

The same experiment is carried out with a different initial point. Figure 13 presents the corresponding results. For  $Am$ , the same conclusion holds. For sample sizes 10 and 50,  $A\phi$  yields signal plans with worse performance compared to the initial plan. As of sample sizes 100 and 150, it identifies plans with improved performance.

We now consider a scenario 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 14 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

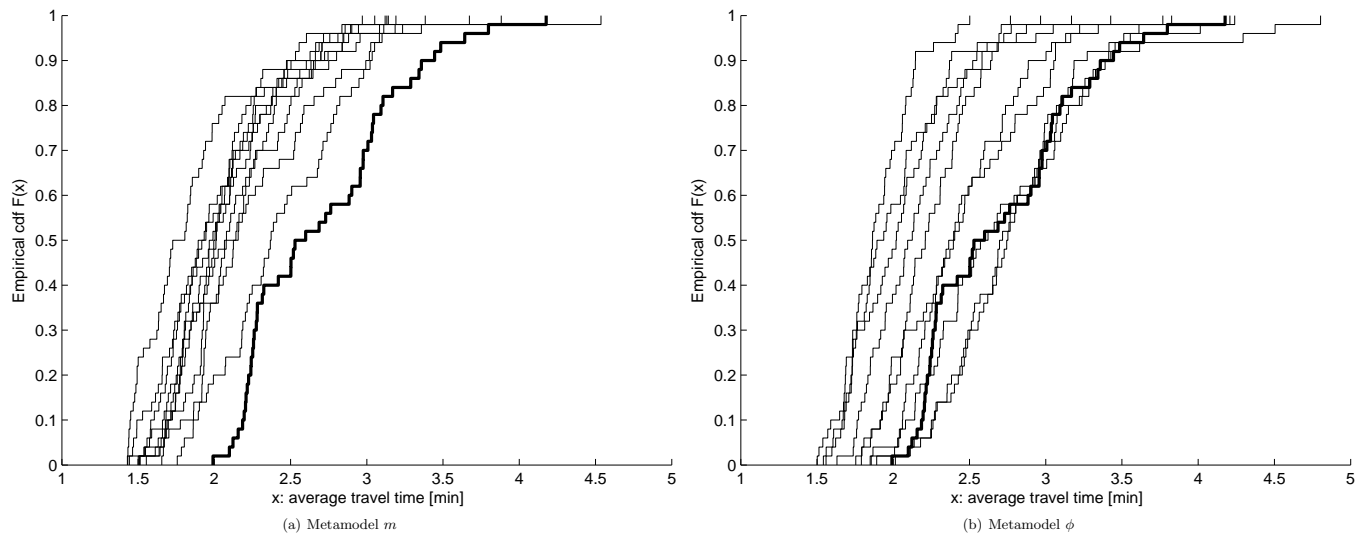


Figure 10: Empirical cdf's of the average travel times considering an initial random signal plan.

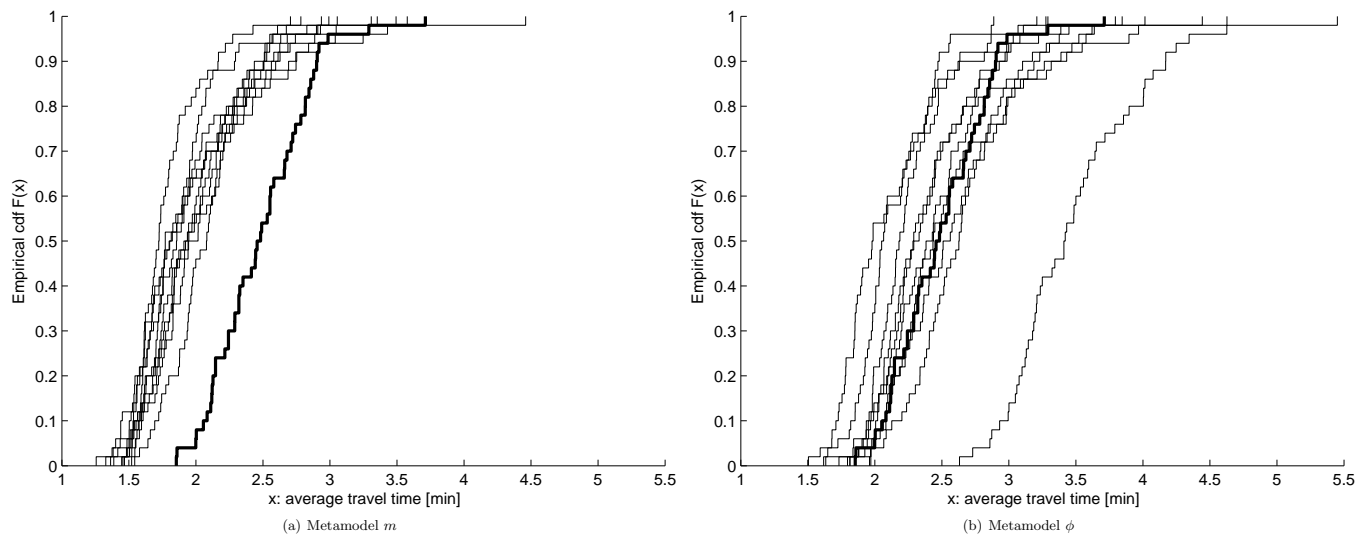


Figure 11: Empirical cdf's of the average travel times considering an initial random signal plan.

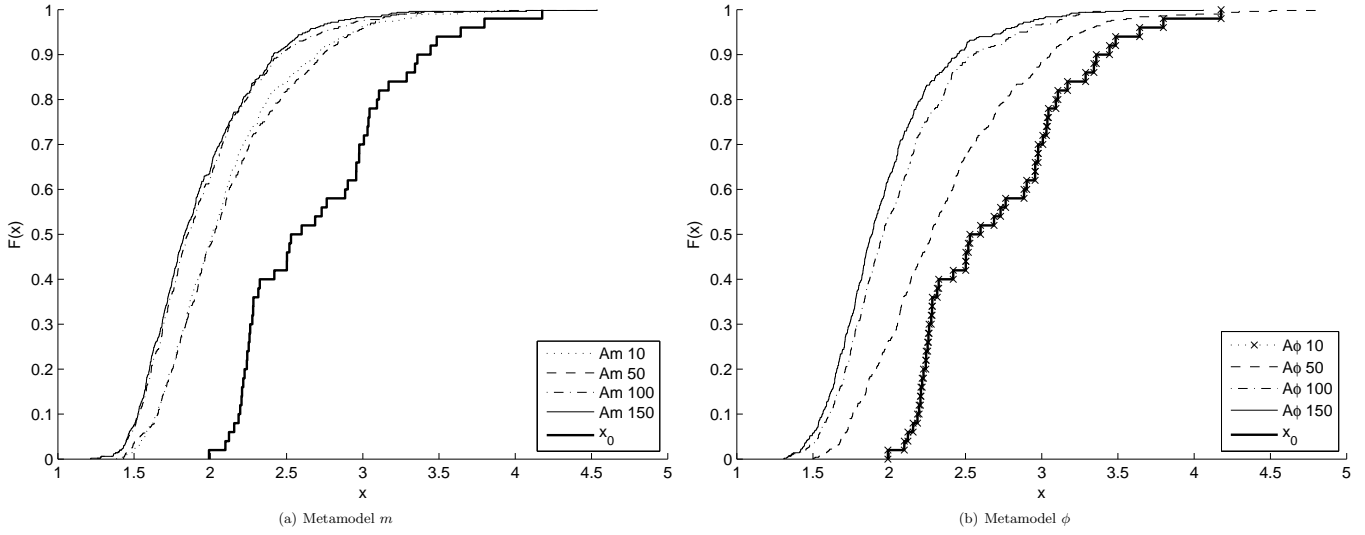


Figure 12: Empirical cdf's of the average travel times considering an initial random signal plans, running 10 instances of each method, and evaluating their performance at different sample sizes.

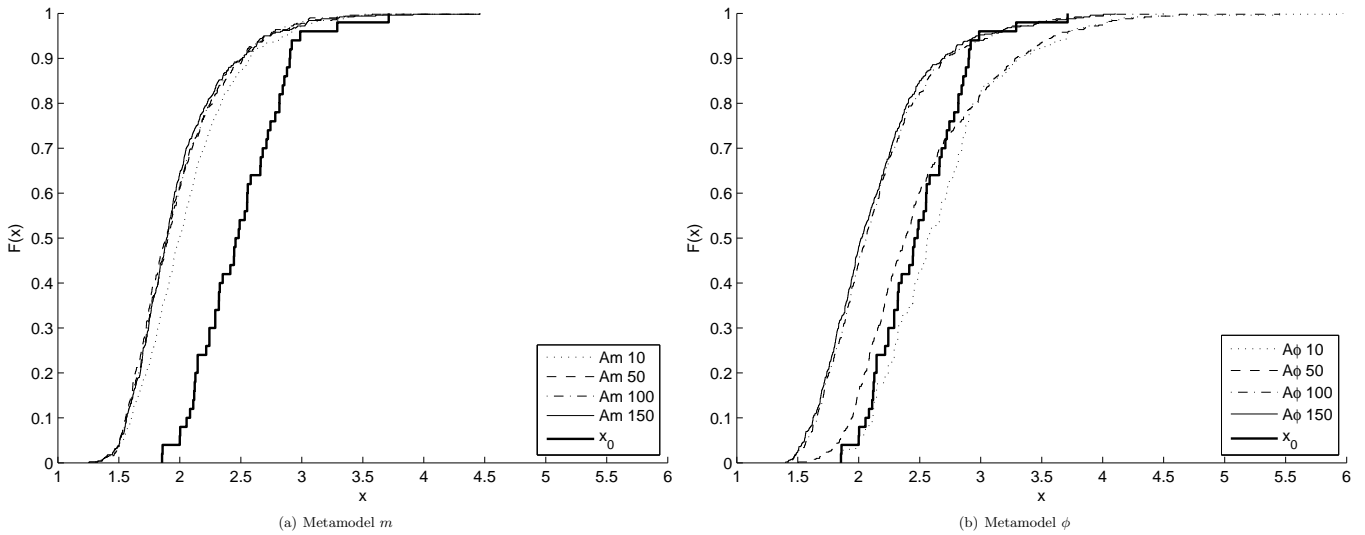


Figure 13: Empirical cdf's of the average travel times considering an initial random signal plans, running 10 instances of each method, and evaluating their performance at different sample sizes.

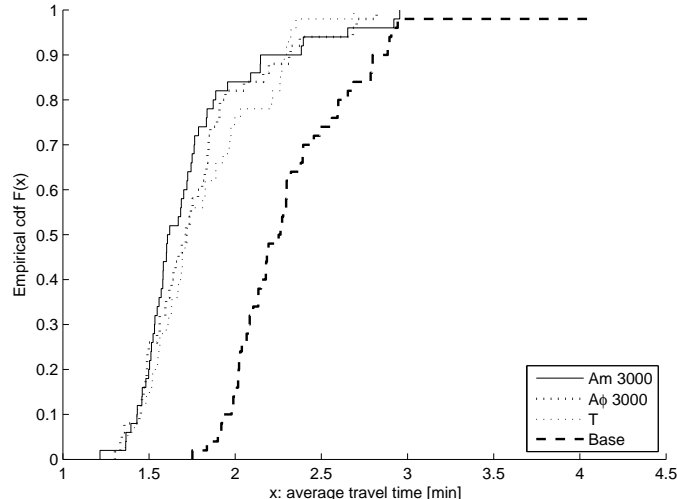


Figure 14: Empirical cdf's of the average travel times considering an initial random signal plans and allowing for 3000 simulation runs.

Lausanne (denoted *Base*).

All three methods (*Am*, *Aφ* 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.

We consider this sample of 3000 observations and test whether the metamodel parameters of the proposed model are statistically different from zero. We perform the corresponding t-tests as described in Section 6.1. In this case there are 86 model parameters. Sixteen are statistically different from zero. These 16 parameters concern 8 linear terms, 6 quadratic terms, the queueing model parameter ( $\alpha$ ) and the intercept ( $\beta_1$ ). This indicates that the proposed metamodel indeed captures information regarding the relationship between the observed average travel times and the decision variables.

We perform the same experiment, allowing for 3000 simulation runs, with a different initial point. Figure 15 presents the corresponding results. The same conclusions hold. We also test whether the proposed metamodel parameters are statistically different from zero. We have similar conclusions, with 18 parameters statistically different from zero.

We evaluate the computational time required by *Am*. We consider the two instances of 3000 simulations, previously presented. We present the times required by the optimization algorithm, and those by the simulation model. Figure 16 presents the empirical cdf of these times. The optimization algorithm is instantaneous in more than 95% of the points, whereas the simulation run time 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 this section indicate that for small sized samples (compared to the dimension of the decision vector) *Am* systematically identifies signal plans with improved performance, whereas limiting the metamodel to a quadratic polynomial (as does *Aφ*) may fail to do so. By providing structural information analytically via the network model, signal plans with improved performance are derived within a tight computational budget. For larger samples, *Am*, *Aφ* and *AT* yield signal plans with similar performance, and improved performance compared to an existing signal plan for Lausanne.



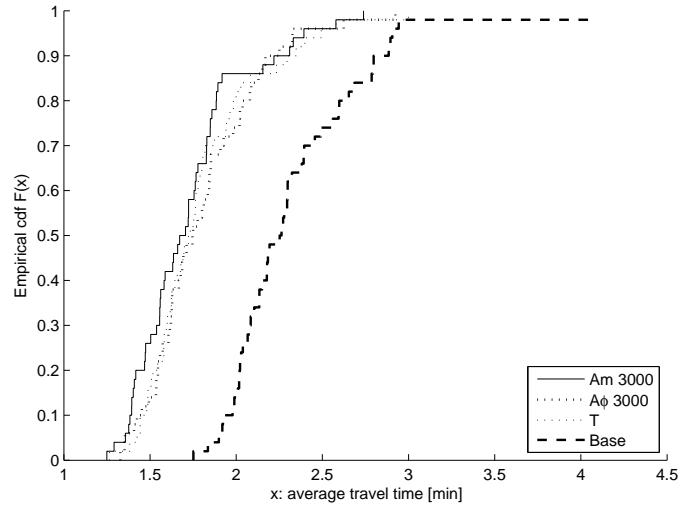


Figure 15: Empirical cdf's of the average travel times considering an initial random signal plans and allowing for 3000 simulation runs.

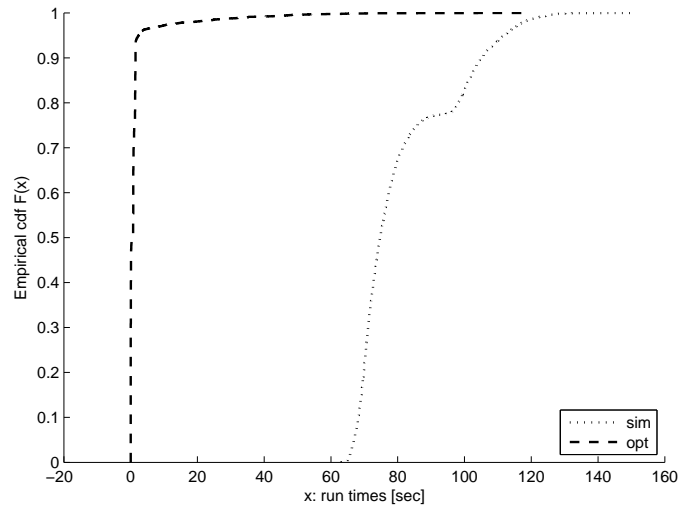


Figure 16: Empirical cdf's of the run times of the optimization algorithm and the simulation model.

## 7 Conclusions

This paper presents a simulation-based optimization framework for the 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 based on a fixed-time signal control problem with subnetworks of the city of Lausanne. The corresponding optimization problem is complex, consisting of a nonlinear objective function, linear and nonlinear equalities and inequalities. Two network sizes and demand scenarios are considered.

We consider scenarios with tight computational budgets, and initialize all runs with an empty sample of simulated observations. For small to moderate sized samples (small compared to the dimension of the decision vector), the proposed method systematically yields signal plans with improved performance when compared to the initial plans and to the plans derived by a traditional approach that limits the metamodel to a polynomial. It also leads to plans with reduced variance in the average travel times, i.e. it increases travel time reliability.

As the sample size increases, the method with a polynomial metamodel leads to signal plans with improved performance. With larger samples, it achieves similar performance to that of the proposed method. For larger samples, both methods perform either better or similarly than a method that limits the metamodel to the queueing model. All three methods lead to improved performance when compared to an existing signal plan for the city of Lausanne.

The analytical information provided by the queueing network model enables the identification of well performing trial points with very small samples, and continues to perform well as the sample size increases. This illustrates the added value of the structural information provided analytically by the network model. The method is therefore computationally efficient and suitable for complex problems with tight computational budgets.

Efficiently tackling unconstrained high dimensional problems (e.g. more than 100 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, e.g. quadratic polynomials, require a moderate to large sample to initially fit the metamodel of interest. By combining these generic metamodels with application-specific models that analytically capture the structure of the underlying problem, these algorithms can be used to tackle high dimensional problems under tight computational budgets. This added structure overcomes the need for a substantial initial sample, and provides meaningful trial points since the very first iterations.

The general-purpose component of the metamodel, the polynomial, ensures asymptotic metamodel properties (full linearity) needed for convergence analysis. More research is needed to investigate whether other functional forms (e.g. splines, radial basis functions) or other interaction terms can improve the accuracy of the metamodel. We will also investigate the sensitivity of the method to the numerous algorithmic parameters.

The accuracy of the network component of the metamodel can also be improved by calibrating its exogenous parameters as the simulated observations are collected. In particular, we expect the calibration of the exogenous parameters that depend on the decision vector, such as the turning probabilities, to further improve the models accuracy.

This accuracy and the computational efficiency of this method can be further improved by investigating different computational resource allocation schemes, e.g. increasing the number of points sampled in the model improvement step, or investigating different sampling strategies.

## 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., Dennis, J. E., Lewis, R. M. and Torczon, V. (1998). A trust region framework for managing the use of approximation models in optimization, *Structural Optimization* **15**: 16–23.
- Alexandrov, N. M. and Lewis, R. M. (2001). An overview of first-order model management for engineering optimization, *Optimization and Engineering* **2**: 413–430.
- Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L. and Newman, P. A. (1999). Optimization with variable-fidelity models applied to wing design, *Technical Report CR-1999-209826*, NASA Langley Research Center, Hampton, VA, USA.
- Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L. and Newman, P. A. (2001). Approximation and model management in aerodynamic optimization with variable-fidelity models, *Journal of Aircraft* **38**(6): 1093–1101.
- Alexandrov, N. M., Nielsen, E. J., Lewis, R. M. and Anderson, W. K. (2000). First-order model management with variable-fidelity physics applied to multi-element airfoil optimization, *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, USA.
- Andradóttir, S. (1998). A review of simulation optimization techniques, *Proceedings of the 30th conference on Winter simulation*, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 151–158.
- Atkeson, C. G., Moore, A. W. and Schaal, S. (1997). Locally weighted learning, *Artificial Intelligence Review* **11**: 11–73.
- Bandler, J. W., Cheng, Q., Dakrouy, A., Mohamed, A., Bakr, M., Madsen, K. and Søndergaard, J. (2004). Space mapping: The state of the art, *IEEE Transactions on Microwave Theory and Techniques* **52**(1): 337–360.
- Bandler, J. W., Koziel, S. and Madsen, K. (2006). Space mapping for engineering optimization, *SIAG/Otimization Views-and-News* **17**(1): 19–26.
- Barton, R. R. and Ivey, J. S. (1996). Nelder-mead simplex modifications for simulation optimization, *Management Science* **42**(7): 954–973.
- Barton, R. R. and Meckesheimer, M. (2006). Metamodel-based simulation optimization, in S. G. Henderson and B. L. Nelson (eds), *Handbooks in operations research and management science: Simulation*, Vol. 13, Elsevier, Amsterdam, chapter 18, pp. 535–574.
- Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V. and Trosset, M. W. (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*, PhD thesis, Rice University.

- Coleman, T. F. and Li, Y. (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. and Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds, *SIAM Journal on Optimization* **6**: 418–445.
- Conn, A. R., Gould, N. I. M. and Toint, P. L. (2000). *Trust-region methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, USA.
- Conn, A. R., Scheinberg, K. and Toint, P. L. (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, A. R., Scheinberg, K. and Vicente, L. N. (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, A. R., Scheinberg, K. and Vicente, L. N. (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. and Bert, E. (2006). Simulation de l’agglomération Lausannoise SIMLO, *Technical report*, Laboratoire des voies de circulation, ENAC, Ecole Polytechnique Fédérale de Lausanne.
- Fu, M. C. (2006). Gradient estimation, in S. G. Henderson and B. L. Nelson (eds), *Handbooks in operations research and management science: Simulation*, Vol. 13, Elsevier, Amsterdam, chapter 19, pp. 576–616.
- Fu, M. C., Glover, F. W. and April, J. (2005). Simulation optimization: a review, new developments, and applications, in M. E. Kuhl, N. M. Steiger, F. B. Armstrong and J. A. Joines (eds), *Proceedings of the 2005 Winter Simulation Conference*, Piscataway, New Jersey, USA, pp. 83–95.
- Kiefer, J. and Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function, *The Annals of Mathematical Statistics* **23**(3): 462–466.
- Kleijnen, J. P. C. (2008). Response surface methodology for constrained simulation optimization: An overview, *Simulation Modelling Practice and Theory* **16**(1): 50–64.
- Kleijnen, J. P. C., van Beers, W. and van Nieuwenhuysse, I. (2010). Constrained optimization in expensive simulation: Novel approach, *European Journal of Operational Research* **202**(1): 164–174.
- Marti, K. (2008). *Stochastic optimization methods*, Springer, Berlin.
- Moré, J. and Wild, S. (2009). Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization* **20**(1): 172–191.
- Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization, *The Computer Journal* **7**(4): 308–313.
- 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. (2010). *Mitigating network congestion: analytical models, optimization methods and their applications*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne.

- Osorio, C. and Bierlaire, M. (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. and Bierlaire, M. (2009b). A surrogate model for traffic optimization of congested networks: an analytic queueing network approach, *Technical Report 090825*, Transport and Mobility Laboratory, ENAC, Ecole Polytechnique Fédérale de Lausanne.
- Powell, M. J. D. (2003). On trust region methods for unconstrained minimization without derivatives, *Mathematical Programming* **97**(3): 605–623.
- Robbins and Monro (1951). A stochastic approximation method, *The Annals of Mathematical Statistics* **22**(3): 400–407.
- Serafini, D. B. (1998). *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*, PhD thesis, Rice University.
- Søndergaard, J. (2003). *Optimization using surrogate models - by the Space Mapping technique*, PhD thesis, Technical University of Denmark.
- Spall, J. C. (2003). *Introduction to stochastic search and optimization: estimation, simulation, and control*, Wiley-Interscience series in discrete mathematics and optimization, John Wiley & Sons, New Jersey, USA.
- Spall, J. C. and Chin, D. C. (1997). Traffic-responsive signal timing for system-wide traffic control, *Transportation Research Part C: Emerging Technologies* **5**(3-4): 153–163.
- Stafford, R. (2006). *The Theory Behind the 'randfixedsum' Function*. <http://www.mathworks.com/matlabcentral/fileexchange/9700>.
- The Mathworks, I. (2008). *Optimization Toolbox Version 4. User's Guide Matlab*, Natick, MA, USA.
- TSS (2008a). *AIMSUN 5.1 Microsimulator Users Manual*, Transport Simulation Systems.
- TSS (2008b). *AIMSUN NG and AIMSUN Micro Version 5.1*, 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, S. M., Regis, R. G. and Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions, *SIAM Journal on Scientific Computing* **30**: 3197–3219.