AN ADAPTIVE ALGORITHM FOR DEPLOYING SELF-TUNING TRAFFIC CONTROL SYSTEMS

Anastasios Kouvelas*

Dynamic Systems and Simulation Laboratory, Technical University of Crete University Campus, GR-73100 Chania, Greece Phone: +30-28210-37307, fax: +30-28210-37584, e-mail: tasos@dssl.tuc.gr

Markos Papageorgiou

Dynamic Systems and Simulation Laboratory, Technical University of Crete University Campus, GR-73100 Chania, Greece Phone: +30-28210-37289, e-mail: markos@dssl.tuc.gr

Elias B. Kosmatopoulos Department of Electrical and Computer Engineering, Democritus University of Thrace University Campus, GR-67100 Xanthi, Greece Phone: +30-25410-79533, e-mail: kosmatop@ee.duth.gr

*Corresponding Author Number of words: 4.693 + 750 (3 Figures) + 500 (2 Tables) = 5.943

ABSTRACT

Currently, a considerable amount of human effort and time is spent for initialization or calibration of operational traffic control systems. Typically, this optimization (fine-tuning) procedure is conducted manually, via trial-and-error, relying on expertise and human judgment and does not always lead to a desirable outcome. This paper presents a new learn-ing/adaptive algorithm that enables automatic fine-tuning of general traffic control systems. The efficiency and online feasibility of the algorithm is investigated through extensive simulation experiments. The fine-tuning problem of three mutually-interacting control modules – each one with its distinct design parameters – of an urban traffic signal control strategy is thoroughly investigated. Simulation results indicate that the learning algorithm can provide efficient automatic fine-tuning, guaranteeing safe and convergent behavior.

Keywords: Traffic control, parameter optimization, self-optimizing control, adaptive control, large-scale systems.

INTRODUCTION

Despite the continuous advances in the fields of control and computing, the design and deployment of an efficient Traffic Control System (TCS) remains a significant objective, mainly because of the involved complexity and the strong nonlinearities. The ultimate performance of a designed or operational TCS (e.g. urban signal control or ramp metering or variable speed limit (VSL) control) depends on two main factors: (a) the exogenous influences, e.g. demand, weather conditions, incidents; and (b) the values of some design parameters included in the TCS. As a matter of fact, when a new control algorithm is implemented (or an operational but "aged" control algorithm needs to be updated), there is a period of, sometimes tedious, fine-tuning activity that is needed in order to elevate the control algorithm to its best achievable performance. Fine-tuning concerns the selection of appropriate (or even optimal) values for a number of design parameters included in the control strategy. Typically, this fine-tuning procedure is conducted manually, via trial-and-error, relying on expertise and human judgment and without the use of a systematic approach. Experienced engineers (in cooperation with practitioners or system operators) experiment with different sets of design parameters, trying to achieve an acceptable (close to optimal) system performance, according to some pre-specified performance metrics.

To address this problem, a research activity has been ongoing recently ((1), (2), (3)), aiming at developing learning/adaptive algorithms that will enable an automatic fine-tuning of TCSs, so as to reach the maximum performance that is achievable with the utilized control strategy. The algorithm AFT (Adaptive Fine-Tuning) is the latest development in this research direction and is based on rigorous but simple adaptive optimization and stochastic approximation principles. The proposed learning algorithm is aiming at replacing the conventional manual optimization practice with a fully automated online procedure.

BACKGROUND

This section presents the problem formulation and a short review of stochastic approximation algorithms and methodologies which attempt to solve the same problem.

Problem formulation

Consider a general discrete-time control system where the underlying dynamics are described according to the following nonlinear first-order difference equation

$$z(t+1) = F(z(t), u_i(t), d(t), t), \quad z(0) = z_0$$
(1)

where $z(t), u_i(t), d(t)$ are the vectors of system states, control inputs, and exogenous (possibly measurable) signals, respectively, t denotes the discrete time-index, i denotes the regulator-index and $F(\cdot)$ is a sufficiently smooth nonlinear vector function. Note, that the proposed methodology can be applied to a system even if the function F is unknown.

Consider also, that one or more control laws are applied to the system (1), which are described as follows:

$$u_i(t) = \varpi_i\left(\theta_i, z(t)\right) \tag{2}$$

where $\varpi_i(\cdot)$ are known smooth vector functions and θ_i is the vector of the tunable parameters for the *i*-th regulator. Note, that we do not impose any restriction neither on the form of the equation (2), nor on the number of the applied control laws. Also, the discrete time-index t may be different for each control law *i*.

The overall system performance is evaluated through the following objective function (performance index)

$$J(\theta; z(0), D_T) = \pi_T(z(T)) + \sum_{i=1}^{I} \sum_{t=0}^{T-1} \pi_{i,t} (z(t), u_i(t))$$

= $\pi_T(z(T)) + \sum_{i=1}^{I} \sum_{t=0}^{T-1} \pi_{i,t} (z(t), \varpi (\theta_i, z(t)))$ (3)

where $\theta = \operatorname{vec}(\theta_1, \theta_2, \dots, \theta_I)$, π_T and $\pi_{i,t}$ are known non-negative functions, I is the number of the fine-tuned regulators, T the finite time-horizon over which the control laws (2) are applied and $D_T \stackrel{\triangle}{=} [d(0), d(1), \dots, d(T-1)]$ denotes the time-history of the exogenous signals over the optimization horizon T. By defining $x = \operatorname{vec}(z(0), D_T)$, equation (3) may be rewritten as

$$J(\theta; z(0), D_T) = J(\theta, x).$$
(4)

Equation (4) indicates that the system performance is affected by the vector of the system's tunable parameters θ and the exogenous vector x. The problem in hand is to develop an appropriate iterative algorithm, which will be applied every T and will update the current control system parameters vector θ , so as to achieve better performance but also provide safe and efficient behavior. This means, that the algorithm should guarantee the stable and sustainable system performance during the fine-tuning process. This requirement is crucial in most practical TCS fine-tuning applications,

since its violation may cause serious performance, safety, etc., problems. For instance, in the case of traffic control systems fine-tuning, the violation this requirement may lead to serious problems (e.g., complaints, dangerous driving, etc.) that may force the traffic operators to cancel the fine-tuning process.

Related methodologies

The fine-tuning problem described in the previous section is closely related to the parameter estimation problem. Let Θ denote the domain of allowable values for the vector θ of dimension n_{θ} . The main problem of interest is to find the values of a vector $\theta^* \in \Theta$ that maximize a scalar-valued performance function $J(\theta, x)$. The vector θ represents a collection of tunable (or "adjustable") parameters that one is aiming to pick in the best way. The nonlinear function $J(\theta, x)$ is a scalar measure that summarizes the performance of the system for a given set of values of the tunable parameters. The domain Θ reflects allowable values (constraints) on the elements of θ and θ^* represents the optimal solution.

Many stochastic approximation algorithms have been developed for the solution of this problem. Robbins and Monro (4) were the first to propose an adaptive technique for parameter estimation. Important generalizations and extensions of this algorithm followed close behind by Kiefer and Wolfowitz in (5) where the FDSA (Finite Difference Stochastic Approximation) algorithm was introduced. FDSA has provided the basis for many learning or "parameter tuning" algorithms in control engineering problems. An extension of FDSA is the RDSA (Random Directions Stochastic Approximation) algorithm which was firstly introduced in (6) and makes use of many random perturbations of the tunable parameters θ in order to come up with a "good" set (according to the performance criterion $J(\theta, x)$).

Finally, Spall in (7) introduced the SPSA (Simultaneous Perturbation Stochastic Approximation) algorithm for stochastic optimization of multivariate systems. In this paper we compare the performance of SPSA algorithm with the AFT algorithm to an online large-scale fine-tuning problem of the urban signal control strategy TUC ((8), (9)) through simulation experiments.

THE PROPOSED ALGORITHM

Introduction

Figure 1 illustrates the working principle of the AFT algorithm. The basic functioning procedure for AO methods may be summarized as follows:

- The traffic flow process (e.g. urban road network) is controlled in real time by a control strategy (of any kind) which includes a number of parameters to fine-tune.
- At the end of appropriately defined periods (e.g. at the end of each day), the AFT algorithm receives the value of the real (measured) performance index (e.g. average speed over space and time for traffic networks, total number of containers loaded/unloaded for seaport container terminals, etc.), as well as some aggregated values of the most significant external factors (e.g. demand). Note, that the performance index $J(\theta, x)$ is a (generally unknown) function of the external factors x and the tunable parameters to be adjusted θ .
- Using the measured quantities (the number of which increases iteration by iteration), the AFT algorithm calculates new tunable parameter values to be applied at the next period (e.g. the next day) in an attempt to improve the system performance.
- This (iterative) procedure is continued over many periods (e.g. days) until a maximum in performance is reached; then, the AFT algorithm may remain active for continuous adaptation or can be switched off and re-activated at a later stage (e.g. after few months).

The developed approach appropriately combines the nice features of adaptive optimization algorithms with those of approximation theory. The resultant adaptive optimization methodology is capable of rapidly and efficiently optimizing systems of arbitrary complexity and scale, such as general TCS and, most importantly, guaranteeing robust and safe performance while the maintenance operation is on. The main components of the employed algorithm are summarized as follows:

- An approximator $\widehat{J}(\theta, x)$ is used (e.g., a neural network or a polynomial-like approximator) in order to obtain an approximation of the nonlinear mapping $\widehat{J}(\theta, x) = J(\theta, x)$.
- An on-line adaptive/learning mechanism is employed for "training" the above approximator. Globally convergent learning algorithms (see e.g., (10), (11)) are required for such a purpose.



Figure 1: Working principle of AFT for automatic calibration of TCSs.

• At each algorithm iteration k, many randomly chosen candidate perturbations (of dimension n_{θ}) of vector $\theta(k)$ are generated. The effect of each of these perturbations to the LNTCS performance is estimated by use of the approximator mentioned above. The perturbation that corresponds to the "best" estimate (i.e., the one that leads to the best value for \hat{J}) is picked to depict the new values for the tunable parameters $\theta(k + 1)$. That is, $\theta(k + 1)$ corresponds to the best estimate of \hat{J} , selected to be applied at the next period (e.g. the next day).

Linear-in-the-weights Universal Approximator

The universal approximator used in the simulation experiments in order to approximate the objective function $J(\theta, x)$, is a linear-in-the-weights polynomial-like approximator with L_q regressor terms, which takes the form

$$\widehat{J}(\theta, x) = \vartheta^T \phi(\theta, x) \tag{5}$$

where ϑ denotes the vector of the approximator parameter estimates and

$$\phi(\theta, x) = \left[\phi_1(\theta, x), \phi_2(\theta, x), \dots, \phi_{L_g}(\theta, x)\right]^T.$$
(6)

The non-linear functions $\phi_i(\theta, x)$ are given by

$$\phi_i(\theta, x) = S^{d_1}(\theta_{m_1}) \cdot \bar{S}^{d_2}(x_{m_2}) \cdot S^{d_3}(\theta_{m_3}), \ d_i \in \{0, 1\}$$
(7)

where d_i, m_i are randomly chosen at each iteration of AFT (with $m_1, m_3 \in \{1, 2, ..., n_\theta\}$, $m_2 \in \{1, 2, ..., n_x\}$ and $\sum_i m_i \in \{2, 3\}$) and $S(\cdot), \overline{S}(\cdot)$ are smooth monotone nonlinear functions. In the neural networks literature (12), (13) these functions are usually chosen to be "sigmoidal". In our simulations we choose

$$S(\theta) = \tanh\left(\lambda_1\theta + \lambda_2\right), \quad \bar{S}(x) = \tanh\left(\lambda_3 x + \lambda_4\right) \tag{8}$$

where λ_i are non-negative real numbers initially defined by the user; after 4–5 iterations of the algorithm the values of λ_i are optimized so as to minimize

$$\min\sum_{\ell=1}^{k-1} \left(J_{\ell} - \vartheta^T \phi_{\ell}^{(k)} \right)^2.$$
(9)

The factors ℓ_i are relevant to the normalization of the values of the approximator inputs θ, x .

AFT algorithm description

For the implementation of AFT, it is assumed that an estimate, or prediction, $\bar{x}(k)$ of the vector x(k) is available (referring to the estimate of next iteration). In many applications such an assumption is quite realistic, since the entries of x(k)correspond to system states and exogenous inputs, which are available or measurable. However, there may be cases where such an assumption is not realistic; in this case $\bar{x}(k)$ can be estimated/predicted using appropriate estimation algorithms. It should be noted, that the dimension n_x of the estimate vector $\bar{x}(k)$ should be comparative to the dimension n_θ of the vector of tunable parameters θ . Below, we discuss in details the application steps of the algorithm displayed in Table 1, which are performed at every iteration k:

Step 1: $\Delta \theta_i^{(j)}(k+1) = \alpha(k)\Delta_i^{(j)}(k) - \theta_i(k) + \theta_i^*(k), j \in \{1,, K\}$				
Step 2: $L_g(k) = \min \{2(k-1), \bar{L}_g\}$				
Step 3: $\ell(k) = \max\{k - T_h, 1\}$				
Step 4: $\phi_{\ell}(k) = \phi\left(\theta_{\ell}(k), \bar{x}_{\ell}(k)\right)$				
$\vartheta(k) \mapsto rg \min_{\vartheta} \frac{1}{2} \sum_{\ell=\ell(k)}^{k} \left(J_{\ell} - \vartheta^{\tau} \phi_{\ell}(k) \right)^2$				
$\text{Step 6:} \widehat{J}\left(\pm\Delta\theta^{(j)}(k+1) + \theta^*(k), \bar{x}(k+1)\right) = \vartheta(k)^\tau \phi\left(\pm\Delta\theta^{(j)}(k+1) + \theta^*(k), \bar{x}(k+1)\right)$				
Step 7: $\Delta \theta(k+1) = \arg \max_{\Delta \theta^{(\pm j)}(k+1)} \widehat{J}\left(\pm \Delta \theta^{(j)}(k+1) + \theta^*(k), \bar{x}(k+1)\right)$				
$\Delta \theta_i(k+1) = \theta_i(k+1) - \theta_i(k)$				
$\alpha(k)$ is a user-defined positive sequence (e.g. constant stepsize $\alpha(k) \equiv \alpha \in (0, 1)$)				
T_h, L_g, K are user-defined positive integers				
$\theta^*(k) + \Delta \theta(k+1)$ denotes the vector of tunable parameters picked to be applied at				
the next experiment $k + 1$				

Table 1: AFT	algorithm	mathematical	description.

- Step 1: Calculate K random perturbations. In this step K random perturbations are calculated (according to e.g. Gaussian distribution). The resulting candidate vectors $\theta_i(k+1) = \theta_i^*(k) + \Delta \theta_i(k)$ are then projected in Θ , in order to satisfy the problem constraints (with $\theta_i^*(k)$ the best set of tunable parameters found so far).
- Step 2: Calculate the number of approximator regressor terms. The number of the approximator's regressor terms $L_g(k)$ to be used for this iteration is calculated.
- Step 3: *Calculate the number of past measurements*. The algorithm keeps a window of past measurements which moves along with the iterations. In this step the starting point of the window in the past is calculated. The end point of the window is always *k*.
- Step 4: *Produce the polynomial-like approximator*. After steps 2, 3 the structure of the universal approximator may be formed and applied to the window of the past measurements.
- Step 5: Calculate the optimal approximator parameter estimates. The optimal values of the approximetor's parameters ϑ are calculated according to the solution of a least squares estimation method.
- Step 6: Apply the 2K random perturbations $\pm \Delta \theta(k)^{(j)}$ to the $\widehat{J}(k)$. The 2K candidate vectors $\theta^*(k) \pm \Delta \theta(k)^{(j)}$ are applied to the approximator $\widehat{J}(k)$ for evaluation.
- Step 7: *Pick the "best" random perturbation (according to the* \hat{J}_k). The vector $\theta(k+1)$ with the best estimated performance is selected for application to the next simulation experiment.

It is worth noting, that similarly to RDSA, the proposed algorithm introduces random perturbations to the control design parameter vector θ . Besides, the use of random perturbations is crucial for the efficiency of the proposed algorithm as it provides the so-called persistence of excitation (PE) property, which is a sufficient and necessary condition for the neural approximator \hat{J} to be able to efficiently learn the unknown function J. However, due to the use of Step 6 (see Table 1) the proposed methodology avoids poor performance or instability problems, and guarantees safe and efficient performance is fulfilled.

BRIEF INTRODUCTION TO TUC STRATEGY

TUC ((8), (9)) is a recently developed, efficient real-time urban traffic control strategy, whose design principles are based on feedback control theory as opposed to most of the existing strategies employing model-based optimization techniques. TUC consists of four distinct interconnected control modules that allow for real-time control of the following: (a) green times (split); (b) cycle time; (c) offset (green wave along an arterial), as well as (d) provision of public transport priority. These four control modules are complemented by a fifth data processing module. All control modules are based on feedback concepts of various types, which leads to TUC's computational simplicity as compared to model-based optimization approaches, without actually sacrificing efficiency. In this paper, we will concentrate on the fine-tuning of the TUC split, cycle and offset control module parameters.

Split control regulator

The basic methodology used by TUC for split control, is the formulation of the urban traffic control problem as a Linear Quadratic (LQ) optimal control problem, based on a store-and-forward type of mathematical modeling. This modeling philosophy circumvents the inclusion of discrete variables in the signal control problem formulation, thus allowing the application of polynomial-complexity solution methods of optimization and control. This part is been described in details elsewhere ((8), (9), (14)), hence, a short summary is provided here.

The LQ approach leads in a straightforward way to the following multivariable regulator:

$$g(t_s) = g^{\mathsf{N}} - Lx(t_s) \tag{10}$$

where:

- $t_s = 0, 1, 2, \dots$ is the discrete time index reflecting corresponding signal cycles.
- g is the vector of the green times of all stages and all junctions in the network.
- g^{N} is the vector of nominal green times for all network stages; these nominal green times correspond to a prespecified fixed signal plan for the network.
- x is the vector of the vehicle-numbers in all network links.
- *L* is a constant gain matrix (of appropriate dimensions) the values of which depend on the network geometry, the turning rates and the saturation flows, but were found to be little sensitive to moderate variations of these values (14), (15).

Calculation of L is the straightforward outcome of the LQ problem formulation and is carried out off-line (at the lab) once per application network, while the online real-time calculations are limited to the execution of equation (10) with a given constant control matrix L and state measurements $x(t_s)$. After the application of equation (10), a simple low-cost algorithm ((14)) applies any existing constraints (e.g. cycle constraints and minimum admissible green times) to the obtained values of $g(t_s)$. Given the split decisions of this part of the TUC strategy, as well as the input this part has received from the cycle and offset control parts, complete network-wide signal settings including cycle, split, and built-in offsets are available for implementation at the end of split control.

Cycle control regulator

One way to influence traffic conditions via traffic lights is by modifying cycle time. Note that one single cycle time is considered here for the whole network, in order to enable junction coordination via suitable offsets. Fundamentally, a longer cycle time typically increases the junction capacity, because the proportion of the constant lost (intergreen) times becomes accordingly smaller; on the other hand, a longer cycle time may increase vehicle delays in undersaturated junctions due to longer waiting times during the red phase.

The network cycle $C(t_c)$ is calculated from the feedback P-type brunch regulator

$$C(t_c) = \begin{cases} C^N + K_1 \left[\sigma(t_c) - \sigma_{N_1} \right], & \text{if } \sigma(t_c) \le \sigma_{cr} \\ C^N - K_2 \left[\sigma(t_c) - \sigma_{N_2} \right], & \text{if } \sigma(t_c) > \sigma_{cr} \end{cases}$$
(11)

where:

- $t_c = 0, 1, 2, ...$ is the discrete time index for applying cycle control.
- C^{N} a nominal network cycle time (e.g. equal to the minimum admissible cycle C_{\min}).
- σ^{N} a nominal average load (e.g. equal to zero).
- K_1, K_2 are tunable control parameters, the value of which affect the intensity of the control reactions.

A prespecified percentage of network links with currently maximum load (link load $\sigma_z(t_c)$ for link z derives from $\sigma_z(t_c) = x_z(t_c)/x_{z,\max}$, with $x_z(t_c)$ the vehicle-number in the link and $x_{z,\max}$ the capacity of the link) are identified and the corresponding loads are averaged to provide the average maximum load $\sigma(t_c)$. After applying (11), the calculated cycle time is constrained within the range $[C_{\min}, C_{\max}]$, if necessary, to become feasible, where C_{\min} and C_{\max} are the minimum and maximum admissible network cycle times, respectively.

Offset control regulator

The feedback control law that calculates the offset between two successive junctions j_1 and j_2 reads

$$t_{j_1,j_2}(t_c) = \frac{l_z}{\nu_z} - l_z K_z^o \frac{x_z(t_c)}{x_{z,\max}}$$
(12)

where ν_z denotes the free-flow mean speed on link z, l_z the length of link z and $K_z^o = (\nu^c - \nu_z)/\nu_z \nu^c$ is a tunable control parameter (with ν^c the speed of the kinematic wave moving upstream, along the queue of the link). Generalization of the previously described logic leads to the relevant formulas which are used at each successive couple of junctions (j_i, j_{i+1}) along a predefined arterial, in order to specify the offset between the two junctions and consequently along the arterial. In order to implement the new offset specified in (12), a transient cycle time C_{j_2} is temporarily implemented in junction j_2 . The transient cycle is implemented one single time, after which, the junctions are coordinated according to the new offset.

APPLICATION OF AFT TO TUC STRATEGY

In order to evaluate the efficiency of AFT algorithm to the problem of optimizing the design parameters of the split, cycle and offset control modules of TUC, extensive simulation experiments have been carried out, comparing the performance of AFT to the base-case (no AFT case). In the base-case, the aforementioned design parameters of the original TUC system were manually fine-tuned to virtual perfection by the system operators (16).

Network and simulation setup

Chania, located at the north-western part of Crete, is the capital of the prefecture of Chania and covers 12.5km^2 . Chania is the second biggest prefecture of Crete in size, population and development. Figure 2(a) exhibits a satellite view of the trial urban road network (red bullets correspond to the controlled junctions), which has a total length of approximately 8km and consists of 16 controlled junctions.

Figure 2(b) represents the model of the network developed for the simulation investigations. It consists of 16 signalized junctions (nodes) and 60 links (arrows). Each network link corresponds to a particular junction phase. Typical loop-detector locations within Chania urban network links are either around the middle of the link or some 40m upstream of the stop-line. The traffic network characteristics (turning rates, lost times, staging, and saturation flows) and the fixed plan g^N of equation (10) used in AIMSUN and in TUC were provided by the system operators of the Traffic Control Centre (TCC) of the city. Note that the fixed plan g^N is one of the six fixed predefined network signal plans used by the TCC. Split, cycle and offset control modules of TUC strategy are applied to the network for all simulation investigations. Finally, a simulation step of 0.25s is considered for the microscopic simulation model.

For the application of the TUC strategy the following typical design values were used: $k_s = C$, $k_c = 600s$, $C_{\min} = 60s$, $C_{\max} = 120s$, $C^N = C_{\min}$. Also, for the implementation of AFT algorithm the following design values were used: $T_h = 90$, $\bar{L}_g = 150$, K = 100 and initial values to λ_i according to $\lambda_1 = 100$, $\lambda_3 = 0.1$, $\lambda_2 = \lambda_4 = 0$. The exogenous vector x(k) in this application corresponds to the traffic demand (time-history of the number of vehicles entering the network in each of the network origins). Following the procedure presented in the description of AFT algorithm, an estimate of the demand $\bar{x}(k)$ every 1h was constructed (i.e., four different estimates for the demand of the whole day). This low-dimension, noisy estimate $\bar{x}(k)$ of the traffic demand was again based on the average number of vehicles entering every particular origin over the 1-h period and resulted in vectors of dimension $n_x = 88$ (4(hours)x22(network origins)).

Tunable parameters

The next three paragraphs describe the reasoning for the definition of the set of tunable parameters for the split, cycle and offset control respectively.

The aim of regulator (10) is to balance the relative space occupancies $x_z/x_{z,max}$ in the network links, so as to minimize the risk of queue spillovers which may lead to a waste of green time and even to gridlocks. For this reason, a non-negative design parameter b_z is introduced for each link z (so-called "importance factor"), such that the $x_z(t_s)$ -values can be affected. These design parameters are critical for the successful deployment and operation of the signal control strategy TUC, and hence were selected for automated fine-tuning by AFT algorithm.

Equation (11) describes the feedback P-type regulator of the cycle control module of TUC. In this equation there are 5 network-wide design parameters, the selection of which affects the intensity of the cycle control module reactions, and hence may cause a degradation in the overall performance of TUC strategy if not suitably configured. For this reason, the design parameters $K_1, K_2, \sigma_{N_1}, \sigma_{N_2}, \sigma_{cr}$ were selected for automated fine-tuning by AFT algorithm.



Figure 2: The Chania urban road network (a) satellite view of the network; (b) simulation model.

Finally, the feedback regulator (12) is applied for offset coordination in all simulation experiments. Equation (12) is applied to all network links that connect successive junctions that belong to the main arterials of the network. The parameters K_z^o and $1/\nu_z$ (two for each link z), which affect the efficiency of the offset control of TUC were selected for automated fine-tuning by AFT algorithm.

The initial values $\theta(0)$ were chosen so as to correspond to values that have been manually fine-tuned in past field implementations of TUC in Chania network (16). In order to assess the overall system performance the criterion was set to the actual daily network mean speed ($J \equiv ms$). Finally, a projection to the nearest valid point is applied whenever a vector θ violates the imposed constraints.

Demand scenarios and integration with AIMSUN simulator

In order to investigate the performance of AFT algorithm under different traffic conditions, two basic traffic demand scenarios (time-history of vehicles entering the network in the network origins during the day) were designed based on actual measurements, each with a simulation horizon of 4 hours. Scenario 1 comprises medium demand in all network origins, while scenario 2 comprises high demand and the network faces serious congestion for some 2 hours, with some link queues spilling back into upstream links. For simplicity, we assume that a demand scenario with a time horizon of 4 hours corresponds to a day. Each day (iteration of the AFT algorithm) a randomly perturbed 5%-width version of the basic demand scenarios is produced and the assessment criterion is gathered from the AIMSUN simulator. Then, the design parameters of TUC strategy are updated by AFT algorithm according to the calculated assessment criterion.

The overall closed-loop scheme consists of two main control loops as inner and outer loops. The inner loop is used by the TUC strategy to produce the traffic signal settings. More specifically at each cycle C, AIMSUN delivers the (emulated) occupancy measurements at the locations where detectors are placed (as in real conditions). These measurements are used by the control modules of TUC strategy to produce the traffic signal settings (splits, cycle, and offsets). These signal settings are then forwarded to the micro-simulator for application. The outer loop is used by AFT algorithm to update the design parameters of TUC strategy. More specifically, at each day, AIMSUN delivers the mean speed for the whole urban road network. The mean speed is used by AFT algorithm (together with the average demand estimation of next day $\bar{x}(k)$) in order to produce the new values for the design parameters of split, cycle and offset control modules of TUC strategy (the vector $\theta = \text{vec}(\theta_1, \theta_2, \theta_3)$). The new set of the design parameters is then forwarded to TUC strategy for application, and so forth.

SIMULATION RESULTS

In order to evaluate the efficiency of the presented AFT algorithm to the problem of optimizing the design parameters of split, cycle and offset control modules of TUC, extensive simulation experiments have been conducted. AIMSUN is based on stochastic distributions in order to calculate all the internal parameters of the simulations. As a result, two replications of the same simulation are not identical, unless they are fed with the same random seed. For our experiments, 10 simulation runs with different random seeds were carried out for each scenario for statistical justification.

Table 2 displays the average mean speed for the original TUC system and the average mean speed when using AFT or SPSA for the system fine-tuning, for 10 different replications. It can be seen that the use of AFT algorithm leads to an average improvement of the system performance of some 17% for demand scenario 1 and some 36% for demand scenario 2. For scenario 1, the average mean speed for the original TUC system is 16.29 km/h and after the convergence of AFT



Table 2: Comparison of the average mean speed (ms) for original TUC system, AFT and SPSA.

Figure 3: Mean speed trajectories with and without the use of AFT algorithm (a) for scenario 1 (4 different replications); (b) for scenario 2 (4 different replications).

this speed is increased to 19.13 km/h, whereas, for scenario 2 the average mean speed for the original TUC system is 9.67 km/h and is increased to 13.19 km/h.

For the same simulation experiments SPSA algorithm was applied in order to fine-tune TUC's design parameters. It is clear from Table 2 that SPSA cannot increase the mean speed of the network. The scale and complexity of the fine-tuning problem prevents SPSA from providing a good estimation of the system's performance gradient. Thus, the applied sets of parameters by SPSA algorithm are "random" and fail to achieve any improvement to the system performance. The average mean speed for the original TUC system is 16.29 km/h and for SPSA 16.47 km/h.

Results for demand scenario 1

Figure 3(a) compares the network-wide mean speed of the original TUC system (blue line) versus TUC system combined with AFT algorithm (red line) delivered for scenario 1 for 4 different simulation runs. In all diagrams, it can be seen that the application of AFT algorithm to the signal control strategy TUC leads to better performance than the original TUC for this demand scenario.

More precisely, AFT algorithm achieves to optimize the overall system performance within few days (iteration number in x-axis), by efficiently fine-tuning the design parameters for all TUC's control modules, while avoiding decreasing the daily mean speed lower than the initial point. The trajectory of the system performance (mean speed) is persistently increasing until it converges to a local maximum value. Note that the oscillations appearing in both blue and red lines of the diagrams are due to the $\pm 5\%$ daily random perturbations applied to the demand scenario.

Results for demand scenario 2

Figure 3(b) compares the network-wide mean speed of the original TUC system (blue line) versus TUC system combined with AFT algorithm (red line) delivered for 4 different simulation runs for this demand scenario. In all diagrams, it can be seen that the application of AFT algorithm leads to better performance than the original TUC system itself. Again, AFT algorithm achieves to optimize the overall system performance within few days, by efficiently fine-tuning the design parameters for all TUC's control modules, while avoiding decreasing the daily mean speed lower than the initial point.

The mean speed is persistently increasing until it converges to a local maximum value. Note that the oscillations appearing

in the blue lines of the diagrams (no AFT case), are due to the $\pm 5\%$ daily random perturbations applied to the demand scenario. The same perturbations are also applied to the AFT case; however, the oscillations of the red lines are clearly lower. Finally, the diagrams depict slight differences between the replications due to the stochasticity of the simulator and the random demand perturbations. However, all the simulations experiments demonstrate the superiority of AFT algorithm over the manually fine-tuned TUC system.

CONCLUSIONS

The paper investigated the efficiency of the AFT algorithm for the problem of optimizing the design parameters of traffic control systems composed of distinct and mutually-interacting modules. This adaptive optimization methodology aims at replacing the conventional manually-based optimization with a fully-automated procedure. Extensive simulation experiments have been conducted for the signal control problem of the traffic network of Chania, where the design parameters of three distinct and mutually-interacting modules of TUC strategy were fine-tuned by the AFT algorithm.

The simulation results demonstrate that the application of AFT algorithm leads to better network performance (in terms of daily mean speed) compared to the original TUC system. This underlines the superiority of the fully-automated optimization procedure, pursued by the AFT algorithm, even in the case that the design parameters are already manually fine-tuned by field experts, indicating the algorithm's efficiency and real-time feasibility.

ACKNOWLEDGMENTS

The research leading to these results has been partially funded by the European Commission FP7-ICT-5-3.5, Engineering of Networked Monitoring and Control Systems, under the contract \#257806 AGILE. The research project AGILE is coordinated by the Informatics and Telematics Institute (http://www.iti.gr/), Centre for Research and Technology Hellas, 57001 Thessaloniki, Greece.

REFERENCES

- 1. Kosmatopoulos, E. B., Papageorgiou, M., Vakouli, A., and Kouvelas, A. Adaptive fine-tuning of non-linear control systems with application to the urban traffic control strategy TUC. *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 6, 2007, pp. 991–1002.
- Kosmatopoulos, E. B., and Kouvelas, A. Large-scale nonlinear control system fine-tuning through learning. *IEEE Transactions on Neural Networks*, Vol. 20, No. 6, 2009, pp. 1009–1023.
- 3. Kouvelas, A. Adaptive Fine-tuning for Large-scale Nonlinear Taffic control Systems. Ph.D. Thesis, Technical University of Crete, Chania, Greece, 2011.
- 4. Robbins, H., and Monro, S. A stochastic approximation method. Annals of Mathematical Statistics, Vol. 22, 1951, pp. 400-407.
- 5. Kiefer, J., and Wolfowitz, J. Stochastic estimation of a regression function. *Annals of Mathematical Statistics*, Vol. 23, 1952, pp. 462–466.
- 6. Ermoliev, Y. On the method of generalized stochastic gradients and quasi-fejer sequences. Cybernetics, Vol. 5, 1969, pp. 208-220.
- 7. Spall, J. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, Vol. 37, No. 3, 1992, pp. 332–341.
- Diakaki, C., Papageorgiou, M., and Aboudolas, K. A Multivariable Regulator Approach to Traffic-Responsive Network-Wide Signal Control. *Control Engineering Practice*, Vol. 10, 2002, pp. 183–195.
- Diakaki, C., Dinopoulou, V., Aboudolas, K., Papageorgiou, M., Ben-Shabat, E., Seider, E., and Leibov, A. Extensions and New Applications of the Traffic-Responsive Urban Control Strategy: Coordinated Signal Control for Urban Networks. *Transportation Research Record*, No. 1856, 2003, pp. 202–211.
- Kosmatopoulos, E., Polycarpou, M., Christodoulou, M., and Ioannou, P. High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks*, Vol. 6, No. 2, 1995, pp. 422–431.
- Kosmatopoulos, E., Christodoulou, M., and Ioannou, P. Dynamical neural networks that ensure exponential identification error convergence. *Neural Networks*, Vol. 10, No. 2, 1997, pp. 299–314.
- 12. Maiorov, V., and Meir, R. Approximation bounds for smooth functions in $C(\Re^d)$ by neural and mixture networks. *IEEE Transactions on Neural Networks*, Vol. 9, No. 5, 1998, pp. 969–978.
- 13. Huang, G.-B., Chen, L., and Siew, C.-K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, Vol. 17, No. 4, 2006, pp. 879–892.
- 14. Diakaki, C. Integrated Control of Traffic Flow in Corridor Road Networks. Ph.D. Thesis, Technical University of Crete, Chania, Greece, 1999.
- Aboudolas, K., Papageorgiou, M., Kouvelas, A., and Kosmatopoulos, E. A Rolling-Horizon Quadratic-Programming Approach to the Signal Control Problem in Large-Scale Congested Urban Road Networks. *Transportation Research*, Vol. 18C, No. 5, 2010, pp. 680–694.
- Kosmatopoulos, E., Papageorgiou, M., Bielefeldt, C., Dinopoulou, V., Morris, R., Mueck, J., Richards, A., and Weichenmeier, F. International Comparative Field Evaluation of a Traffic-Responsive Signal Control Strategy in Three Cities. *Transportation Research*, Vol. 40A, No. 5, 2006, pp. 399–413.