SAFFRON: Store-And-Forward model toolbox For urban ROad Network signal control in MATLAB

Leonardo Pedroso[1], Pedro Batista[1], Markos Papageorgiou[2,3], and Elias Kosmatopoulos[4]

[1] *Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal*
[2] *Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece*
[3] *Faculty of Maritime and Transportation, Ningbo University, Ningbo, China*
[4] *Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece*

Accepted version

Accepted on 16 Jun 2022

Cite as

L. Pedroso, P. Batista, M. Papageorgiou, and E. Kosmatopoulos, 'SAFFRON: Store-And-Forward model toolbox For urban ROad Network signal control in MATLAB', in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 3698–3703. doi: 10.1109/ITSC55140.2022.9922508.

SAFFRON: Store-And-Forward model toolbox For urban ROad Network signal control in MATLAB

Leonardo Pedroso[1], Pedro Batista[1], Markos Papageorgiou[2], and Elias Kosmatopoulos[3]

***Abstract*— The SAFFRON toolbox is introduced to synthesize, analyze, and simulate store-and-forward based strategies for the signal control problem in congested urban road networks in MATLAB. It features: i) well-documented tools to manipulate and simulate store-and-forward macroscopic traffic network models; ii) the parameters of the model of the urban road network of the city center of Chania, Greece; and iii) the implementation of state-of-the-art signal control strategies.**

I. INTRODUCTION

The simulation of traffic dynamics is of paramount importance to evaluate the performance of a signal control strategy before it is implemented in a real traffic network. Moreover, it enables the assessment of the viability of innovative solutions. There are some open-source toolboxes for the microscopic simulation of large-scale traffic networks, such as SUMO [1], MATSim-T [2], and Open Traffic [3]. Nevertheless, emerging traffic control solutions are often synthesized for macroscopic models, albeit significantly simpler and not as accurate, and their performance is evaluated, in a first instance, resorting to macroscopic simulations.

Nevertheless, there are two aspects hampering the research of traffic signal control strategies in a macroscopic setting. First, the models used by each research group are not open-source, and thus not available to the community. Second, the source-code of the simulations reported in scientific results is rarely made publicly available. For these reasons: i) it is not possible to reproduce the results reported for innovative control strategies; ii) it is challenging to compare the performance of different solutions; and iii) it is difficult for young researchers to enter the field of Intelligent Transportation Systems. In this paper, we aim at mitigating these undesirable effects by introducing the SAFFRON toolbox for simulating signal control strategies in MATLAB according to the store-and-forward macroscopic model, initially proposed in [4]. This model was used over the past decade to develop high-impact methods, namely the well-known TUC strategy [5].

In this context, the contribution of the SAFFRON toolbox is threefold:

The work of Leonardo Pedroso and Pedro Batista was supported by the Fundação para a Ciência e a Tecnologia (FCT) through LARSyS - FCT Project UIDB/50009/2020 and through the FCT project DECENTER [LISBOA-01-0145-FEDER-029605], funded by the Programa Operacional Regional de Lisboa 2020 and PIDDAC programs.

[1]Leonardo Pedroso and Pedro Batista are with the Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal

[2]Markos Papageorgiou is with the Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece; and with the Faculty of Maritime and Transportation, Ningbo University, Ningbo, China.

[3]Elias Kosmatopoulos is with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece

1) Providing open-source and well documented tools for the synthesis, analysis, and simulation of store-and-forward models of urban road networks;
2) Making a model of the traffic network of the city center of Chania, Greece, available, which has already been used to validate high-impact strategies proposed by the DSSLab [6], [5], [7], [8], [9];
3) Sharing the source-code of an implementation of the TUC strategy, as well as two novel decentralized strategies proposed in [10], using the tools provided in the SAFFRON toolbox.

The aim of the toolbox is, thus, to provide the communities of Control and of Intelligent Transportation Systems with open-source tools, a model, and the implementation of state-of-the-art methods. Therefore, future strategies based on the store-and-forward model: i) can be seamlessly simulated; ii) can be applied to a meaningful model that is publicly available and, hence, can be reproduced; and iii) can be compared with other strategies with ease. The effort required by new researchers entering the field is decreased, especially in computational development terms, and the development and evaluation of novel solutions becomes more transparent.

The SAFFRON toolbox is hosted on a open-source repository at `https://github.com/decenter2021/SAFFRON`. Thorough low-level documentation is provided in the repository webpage, as well as in the source files. The community is encouraged to contribute to SAFFRON with suggestions, additions, and the implementation of signal control strategies.

This paper is organized as follows. In Section II, the store-and-forward macroscopic traffic dynamics model is briefly described. In Section III, the features of the SAFFRON toolbox are described. Finally, Section IV presents the main conclusions of this paper.

A. Notation

The identity and null matrices, both of appropriate dimensions, are denoted by $\mathbf{I}$ and $\mathbf{0}$, respectively. Alternatively, $\mathbf{I}_n$ and $\mathbf{0}_{n\times m}$ are also used to represent the $n \times m$ identity matrix and the $n \times m$ null matrix, respectively. The i-th component of a vector $\mathbf{v} \in \mathbb{R}^n$ is denoted by $[\mathbf{v}]_i$, and the entry (i, j) of a matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{ij}$. The column wise concatenation of vectors $\mathbf{x_1}, \dots, \mathbf{x_N}$ is denoted by $\mathrm{col}(\mathbf{x_1}, \dots, \mathbf{x_N})$. The block diagonal matrix whose diagonal blocks are given by matrices $\mathbf{A_1}, ..., \mathbf{A_N}$ is denoted by $\mathrm{diag}(\mathbf{A_1}, ..., \mathbf{A_N})$. Moreover, $\mathrm{diag}(\mathbf{v}) \in \mathbb{R}^{n\times n}$, where $\mathbf{v} \in \mathbb{R}^n$ is a vector, denotes the diagonal matrix whose diagonal entries correspond to the entries of $\mathbf{v}$. The cardinality of a set $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

II. Store-and-forward model

The store-and-forward model was initially proposed in [4]. The brief description presented herein closely follows the one in [6] and [10]. The topology of a traffic network, which is assumed to be time invariant, can be defined by the interconnection of the junctions via directional links. Such topology may be represented by a directed graph $\mathcal{G} := (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, composed of a set $\mathcal{V}_\mathcal{G}$ of vertices and a set $\mathcal{E}_\mathcal{G}$ of directed edges. For a vertex i, its in-degree, ν_i^-, is the number of edges directed towards it, and $\mathcal{E}_i^-$ is the set of such edges. Conversely, for a vertex i, its out-degree, ν_i^+, is the number of edges directed from it, and $\mathcal{E}_i^+$ is the set of such edges. A directed walk of length n is an ordered sequence of edges denoted by $p = (e_1, \ldots, e_{n-1})$ for which there exists a sequence of vertices $(v_1, \ldots, v_n)$ such that $e_i = (v_i, v_{i+1})$.

Consider a traffic network with links $z \in \{1, \ldots, Z, Z+1 \ldots, \tilde{Z}\}$ and signalized junctions $j \in \{1, \ldots, J\}$. In this framework, each junction is represented by a vertex and if there is a directional link z from junction i towards junction j, then this link is represented by an edge directed from vertex i towards vertex j, *i.e.*, edge $e_z = (i, j)$ with $z \in \{1, ..., Z\}$. If there is a link z from outside of the network towards a vertex j then it represented by $e_z = (0, j)$, with $z \in \{1, ..., Z\}$. Conversely, a link z directed from a vertex j towards outside of the network is represented by $e_z = (j, 0)$, with $z \in \{Z+1, ..., \tilde{Z}\}$. Links $z \in \{Z+1, ..., \tilde{Z}\}$ are not considered in the traffic network control, since their flow is not controlled by any of the junctions. According to the configuration of the network, a vehicle in link z has the possibility of turning to link $w \in O_z$, where

$$O_z := \left\{ w \in \mathbb{N} : \exists j \in \{1, \ldots, J\} : e_z \in \mathcal{E}_j^- \wedge e_w \in \mathcal{E}_j^+ \right\} .$$

Conversely, the set of links with the possibility of turning to link z is defined as

$$I_z := \left\{ w \in \mathbb{N} : \exists j \in \{1, \ldots, J\} : e_w \in \mathcal{E}_j^- \wedge e_z \in \mathcal{E}_j^+ \right\} .$$

Each link z is characterized by: i) a saturation flow $S_z \in \mathbb{R}^+$, expressed in vehicles per unit of time; ii) turning rates $t_{w,z} \in [0, 1]$, where $w \in I_z$; and iii) the link exit rate $t_{z,0} \in [0, 1[$. Define the turning rate matrix $\mathbf{T} \in \mathbb{R}^{Z \times Z}$ as

$$[\mathbf{T}]_{zw} := \begin{cases} t_{w,z} \,, & w \in I_z \\ 0 \,, & w \notin I_z \end{cases} , \quad z, w \in \{1, \ldots, Z\} ,$$

and the exit rates vector $\mathbf{t_0} := [t_{1,0} \ \ldots \ t_{Z,0}]^T \in \mathbb{R}^Z$.

A traffic network is defined by the triplet $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$. Note that there are configurations of $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$ that are not physically meaningful. For that reason, a subset of traffic networks of finite dimension and for which vehicles are not permanently trapped inside it was defined in [10], as detailed in the following definitions, without any loss of generality in the context of traffic network control.

Definition 2.1 (Open traffic network): A traffic network characterized by $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$ is said to be open if, for every edge of the network $e_z \in \mathcal{E}_\mathcal{G}$, there is a directed walk starting at e_z which a vehicle may follow to exit the network with non-zero probability.

Definition 2.2 (Feasible traffic network): A traffic network characterized by $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$ is said to be feasible if

1) $\mathcal{E}_\mathcal{G}$ and $\mathcal{V}_\mathcal{G}$ are finite sets;
2) $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$ is open.

The signal control strategy for each junction j is based on cycles of a given duration C_j, which for the sake of simplicity is considered to be constant and equal to C across all junctions. For each cycle of junction j, there is a fixed number of stages, which belong to the set $\mathcal{F}_j$, each defined by an unique integer $s \in \{1, \ldots, S\}$ network-wise. Each stage s has an associated green time g_s, that is the control variable, which must satisfy the constraint

$$g_s \geq g_{s,\text{min}} \,, s \in \{1, \ldots, S\}, \tag{1}$$

where $g_{s,\text{min}} \in \mathbb{R}$ is the minimum permissible green time for stage s, necessary to guarantee sufficient green time allocated to the pedestrian phases that are allowed during stage s. Then, each cycle has to satisfy the constraint

$$\sum\nolimits_{s \in \mathcal{F}_j} g_s + L_j = C \,, \quad j \in \{1, \ldots, J\} \,, \tag{2}$$

where L_j is the lost time per cycle at junction j, also designated as intergreen time. For each stage, there is a set of links which have right of way. Define the stage matrix $\mathbf{S} \in \mathbb{R}^{Z \times S}$ as

$$[\mathbf{S}]_{zs} := \begin{cases} 1 \,, & \text{if link } z \text{ has r.o.w. at stage } s \\ 0 \,, & \text{otherwise} \end{cases} .$$

Definition 2.3 (Minimum complete stage strategy): A stage strategy characterized by the stage matrix $\mathbf{S}$ is said to be a minimum complete stage strategy if

1) $\forall s \in \{1, \ldots, S\} \ \exists z : [\mathbf{S}]_{zs} = 1$;
2) $\forall z \in \{1, \ldots, Z\} \ \exists s : [\mathbf{S}]_{zs} = 1$;
3) $\forall j \in \{1, \ldots, J\} \, \forall s \in \mathcal{F}_j \ \forall z \in \{1, \ldots, Z\}$ $[\mathbf{S}]_{zs} = 1 \implies e_z \in \mathcal{E}_j^-$;
4) $\forall j \in \{1, \ldots, J\} \forall s_1, s_2 \in \mathcal{F}_j \, s_1 \neq s_2 \implies \nexists k \in \mathbb{R} :$ $[\mathbf{S}]_{s_1} = k [\mathbf{S}]_{s_2}$, where $[\mathbf{S}]_s$ denotes the s-th column of $\mathbf{S}$.

Consider now a link z and denote the number of vehicles in link z as $x_z(k)$ at time kC, where k is the discrete time instant and the cycle time C is the chosen sampling time. The dynamics are modeled by the vehicle conservation equation

$$x_z(k+1) = x_z(k) + C \left(q_z(k) - u_z(k) + d_z(k) - s_z(k) \right) ,$$

where $u_z(k)$ is the outflow of link z; $q_z(k)$ is the inflow given by $q_z(k) = \sum_{w \in I_z} t_{wz} u_w(k)$; $d_z(k)$ is the demand of the link; and $s_z(k)$ is the exit flow of the link, set to $s_z(k) = t_{z,0} q_z(k)$. Additionally, links are subject to constraints

$$0 \leq x_z(k) \leq x_{z,\text{max}} \,, \quad z \in Z \,, \tag{3}$$

where $x_{z,\text{max}} \in \mathbb{R}$ denotes the maximum admissible number of vehicles in link z. To satisfy this constraint, an upstream gating may be put in place in order to avoid overloading any links during periods of high demand (see [6] for more details). After some algebraic manipulations, detailed in [10,

Appendix A], it is possible to write the dynamics as an LTI system with a time-varying disturbance

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B_u}\mathbf{u}(k) + C\mathbf{d}(k)\;, \tag{4}$$

where $\mathbf{x}(k) := \mathrm{col}(x_1(k), ..., x_Z(k)) \in \mathbb{R}^Z, \mathbf{u}(k) := \mathrm{col}(u_1(k), ..., u_Z(k)) \in \mathbb{R}^Z, \mathbf{d}(k) := \mathrm{col}(d_1(k), ..., d_Z(k)) \in \mathbb{R}^Z$, $\mathbf{A} = \mathbf{I}_Z$, and

$$\mathbf{B_u} = C\left((\mathbf{I}_Z - \mathrm{diag}(\mathbf{t_0}))\,\mathbf{T} - \mathbf{I}_Z\right).$$

However, note that the components of $\mathbf{u}(k)$ in (4) cannot be independently selected, as they depend on the different admissible stages at each junction. The store-and-forward model is mainly characterized by the following simplification of the traffic flow, which models green-red switchings within a whole cycle as a continuous flow of vehicles

$$u_z(k) = S_z G_z(k)/C\;, \quad z \in \{1, \ldots, Z\}\;, \tag{5}$$

where k is the discrete time instant, and $G_z(k)$ is the total green time of link z, given by the summation of the green times of each stage for which link z has r.o.w., *i.e.*,

$$G_z(k) = \sum\nolimits_{s:[\mathbf{S}]_{zs}\neq 0} g_s(k)\;. \tag{6}$$

Substituting (5) in (4), according to the store-and-forward model, one obtains the following LTI system with a time-varying disturbance

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B_G}\mathbf{G}(k) + C\mathbf{d}(k)\;, \tag{7}$$

where $\mathbf{G}(k) := \mathrm{col}(G_1(k), ..., G_Z(k)) \in \mathbb{R}^Z$ and

$$\mathbf{B_G} = \mathbf{B}_u \mathrm{diag}(S_1, \ldots, S_Z)/C.$$

Similarly to (4), the components of the command action in (7), $\mathbf{G}(k)$, cannot be independently selected, since they depend on the different admissible stages at each junction. However, the QPC and D2TUC control strategies, proposed in [6] and [10], respectively, make use of this LTI system to find a suitable command action $\mathbf{G}(k)$, and then apply a post-processing algorithm to allocate the green times among the stages. Substituting (6) in (4), after some algebraic manipulation, as detailed in [10, Appendix A], one obtains the LTI system with a time-varying disturbance

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B_g}\mathbf{g}(k) + C\mathbf{d}(k)\;, \tag{8}$$

where $\mathbf{g}(k) := \mathrm{col}(g_1(k), ..., g_S(k)) \in \mathbb{R}^S$ and $\mathbf{B_g} = \mathbf{B_G}\mathbf{S}$. The components of the command action in (8), $\mathbf{g}(k)$, are the green times of each stage, thus can be independently selected. Oftentimes, traffic responsive control strategies rely on a known historic green time of each stage $\mathbf{g_N} \in \mathbb{R}^S$. The TUC and DTUC control strategy, proposed in [5] and [10], respectively, make use of this LTI system to find a suitable command action $\mathbf{g}(k)$. Consider the following results regarding the controllability of the store-and-forward model.

Proposition 2.1: Consider a feasible traffic network characterized by $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$ and a minimum complete stage strategy characterized by a stage matrix $\mathbf{S}$. Let $\boldsymbol{\mathcal{C}}$ be the controllability matrix of the store-and-forward LTI system (8). Then, $\mathrm{rank}(\boldsymbol{\mathcal{C}}) = S \leq Z$.

Proof: See [10, Appendix B]. ∎

Proposition 2.2: Consider a feasible traffic network characterized by $(\mathcal{G}, \mathbf{T}, \mathbf{t_0})$. Then, the store-and-forward LTI system (7) is controllable.

Proof: See [10, Appendix D]. ∎

A. Nonlinear model

Although it is very convenient to work with a linear model for controller synthesis purposes, such as the store-and-forward model, it is insufficient to simulate the network dynamics and assess the performance of a control law, since the nonlinear constraint (3) is not enforced. As put forward in [6], adapting the store-and-forward model, a simple nonlinear discrete-time model can be employed to assess the performance of the store-and-forward based control laws. Considering a sampling time $T \ll C$ and assuming, for simplicity, that $C/T \in \mathbb{N}$, it is possible to write

$$\begin{cases}\mathbf{x}(k_T+1) = \mathbf{A}\mathbf{x}(k_T) + \frac{T}{C}\mathbf{B_u}\mathbf{u_{nl}}(k_T) + T\mathbf{d}(k_T)\\ \mathbf{y}(k_T+1) = \mathbf{C}\mathbf{x}(k_T)\end{cases}, \tag{9}$$

as put forward in [6], where k_T is the discrete time instant corresponding to time $k_T T$ and $\mathbf{u_{nl}}(k_T) := \mathrm{col}(u_{nl,1}(k_T), ..., u_{nl,Z}(k_T)) \in \mathbb{R}^Z$ with

$$u_{nl,z}(k_T) = \begin{cases}0\,, \exists w \in O_z : t_{z,w} \neq 0 \wedge x_w(k_T) > c_{ug} x_{w,\max}\\ \min\{x_z(k_T)/T, u_z(k = \lfloor k_T T/C \rfloor)\}, \text{otherwise}\end{cases}$$

as put forward in [6], in which $u_z(k)$ is the command action, which is updated every cycle C, whose synthesis is based on a linear model as defined in (5), and $c_{ug} \in]0,1[$ is a parameter to be tuned in order to adjust the sensitivity of upstream gating. Note that, in this model, constraint (3) is modeled.

III. SAFFRON Toolbox

The SAFFRON toolbox is open-source and can be accessed online at `https://github.com/decenter2021/SAFFRON`. It includes:

1) a collection of tools for the development and evaluation of store-and-forward based strategies;
2) the model of the urban traffic network of the city center of Chania, Greece;
3) the implementation of state-of-the-art signal control strategies.

In what follows the main features of SAFFRON are detailed. A thorough documentation of the functions and scripts of the toolbox is provided in the online repository.

A. Model Synthesis

The store-and-forward LTI models (4), (7), and (8) can be easily synthesized in SAFFRON by providing: i) the number of junctions, links, and stages, control cycle C, simulation cycle T, and the upstream gating parameter c_{ug}; ii) the lost time and number of stages in each junction; iii) the capacity, saturation flow, number of lanes, initial number of vehicles,

TABLE I: Most relevant fields of store-and-forward model *struct* object.

Field	Description
`J`	J
`Z`	Z
`nStages`	S
`C`	C
`c`	c
`Tsim`	T
`lostTime`	$\text{col}(L_1, \ldots, L_J)$
`nStagesJunction`	$\text{col}(\lvert\mathcal{F}_1\rvert, \ldots, \lvert\mathcal{F}_S\rvert)$
`capacity`	$\text{col}(x_{1,\text{max}}, \ldots, x_{Z,\text{max}})$
`saturation`	$\text{col}(S_1, \ldots, S_Z)$
`x0`	$\mathbf{x}(0)$
`d`	$\text{col}(d_1, \ldots, d_Z)$
`gmin`	$\text{col}(g_{1,min}, \ldots, g_{S,min})$
`gN`	$\mathbf{g_N}$
`T`	$\mathbf{T}$
`t0`	$\mathbf{t_0}$
`S`	$\mathbf{S}$
`junctions`	Cell array indexed by junction number that contains an array of the indices of the stages associated with that junction
`links`	Ordered array of edges of the network graph
`inLinks`	Column vector of link indices that originate from outside the network
`notInLinks`	Column vector of link indices that do not originate from outside the network
`A`	$\mathbf{A}$
`Bu`	$\mathbf{B_u}$
`BG`	$\mathbf{B_G}$
`Bg`	$\mathbf{B_g}$

and demand flow for each link; iv) the minimum green time and historic green time of each stage; v) the stage matrix $\mathbf{S}$, *i.e.*, a table that indicates which links have right of way for each stage; and vi) the turning rates matrix $\mathbf{T}$, *i.e.*, a table that indicates the probability of turning into the links of the network on the exit of a certain link, and the exit rate of all links $\mathbf{t_0}$. These parameters are input into a custom spreadsheet provided in the toolbox. The model is loaded into MATLAB making use of the command

```
>> model = SFMSynthesis("directory")
```

where `directory` is the path of the folder that encloses the spreadsheet of the model one desires to load, and `model` is a *MATLAB struct* object that characterizes the urban road network. The most relevant fields of the *struct* are presented in Table I. The entirety of the fields are detailed in the documentation in the online repository.

B. Utilities

The SAFFRON toolbox also provides tools to check if a traffic network is open and if it has a minimum complete stage strategy, according to Definitions 2.1 and 2.3. The commands

```
>> flag = isOpen(model)
>> flag = isMinimumComplete(model)
```

output booleans that indicate whether a traffic network object `model` is open and whether it has a minimum complete stage strategy, respectively.

The performance metrics total time spent (TTS)

$$\text{TTS} := C \sum_k \sum_{z=1}^{Z} x_z(k)$$

and relative queue balance (RQB)

$$\text{RQB} := \sum_k \sum_{z=1}^{Z} \frac{x_z^2(k)}{x_{z,max}}$$

were introduced in [6, Section 5.2]. These criteria are applied to the average of the values of $x_z(k_T)$, simulated making use of the nonlinear model (9), over each cycle interval C. To compute the TTS and RQB for a simulation of a traffic network `model`, whose global occupancy at each time instant $t = (k_T - 1)T$, $k_T \in \mathbb{N}$ is stored in the k_T-th column of the array `xNL`, the command

```
>> [TTS,RQB] = SFMMetrics(model,xNL)
```

is provided.

The quadratic continuous knapsack problem often arises in a post-processing stage of a continuous traffic signal control policy to allocate the green times among the stages while following constraints (1) and (2). The traffic control solutions proposed in [5], [10] are two examples. The quadratic continuous knapsack optimization problem is given by

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2}\mathbf{x}^T \text{diag}(\mathbf{d})\mathbf{x} - \mathbf{a}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{0} \leq \mathbf{x} \leq \mathbf{b} \\ & \mathbf{1}^T\mathbf{x} = c \ , \end{aligned} \tag{10}$$

where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{b} \geq \mathbf{0} \in \mathbb{R}^n$, $c \in \mathbb{R}_0^+$, and $\mathbf{d} > \mathbf{0} \in \mathbb{R}^n$. The algorithm proposed in [11] is shown to find the solution to (10) in at most n iterations. It is explored in the context of traffic signal control in [5]. This algorithm is implemented in the command

```
>> x = knapsack(a,b,c,d)
```

of SAFFRON, which outputs the solution to (10) and whose arguments `a`, `b`, `c`, and `d`, have a direct correspondence with the variables $\mathbf{a}$, $\mathbf{b}$, c, and $\mathbf{d}$ of (10), respectively.

C. Simulation script

The SAFFRON toolbox also provides a template for the simulation of a traffic signal control policy in MATLAB making use of the nonlinear model (9), which simulates upstream gating. It is available as `simulation_template.m` in the online repository. In this MATLAB script: i) the model is loaded; ii) the initial conditions and demand are adjusted for the desired scenario; ii) it is verified that it the network is open and that it has a minimum complete stage strategy; iii) the signal control law is synthesized making use of the store-and-forward model; iv) the control policy is implemented; v) the nonlinear dynamics are simulated; vi) the performance metrics are computed; and v) the evolution of the occupancy of the links and of the green times is plotted. A novel control policy can be tested seamlessly in this script by

1) Setting the directory of the traffic network model;
2) Setting the initial occupancy and demand;
3) Synthesizing the novel control policy;
4) Implementing the novel control policy, *i.e.*, compute the green-times of the stages as a function of the link occupancy;

in the places indicated in the script. An example of the use of this template to the implementation of state-of-the signal control strategies is also available, as presented in Section III-E.

D. *Chania urban road network*

The SAFFRON toolbox also provides the community with a model of the urban traffic network of the city center of Chania, Greece. This model has been used over the past two decades to validate high-impact traffic signal control strategies [6], [5], [7], [8], [9] and for recently proposed innovative solutions [10], [12]. Nevertheless, this model has never been made publicly available to the community, inhibiting the reproduction of the results that were reported. Therefore, this model is now made available, as part of the toolbox, to ease the reproduction of previous results, the comparison of existing strategies, and to motivate the development of novel solutions.

The Chania urban traffic network, whose topology graph is depicted in Fig. 1, consists of $J = 16$ signalized junctions and $L = 60$ links. The model is provided in the directory

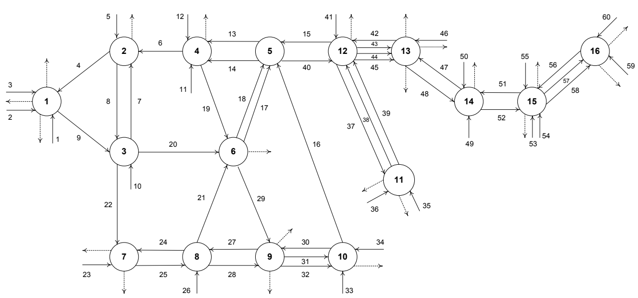

Fig. 1: Chania urban traffic network topology graph.

`ChaniaUrbanRoadModel` of the repository that includes: i) the model data in a spreadsheet that follows the template mentioned in Section III-A; ii) the model object stored in the MAT-file `data.mat`; and iii) an image of the topology graph in Fig. 1. To load the Chania urban road model into the workspace, run one of the commands

```
>> m = SFMSynthesis('ChaniaUrbanRoadModel');
>> m = load('ChaniaUrbanRoadModel/data.mat');
```

in the uppermost directory of the toolbox.

E. *Example*

Finally, the SAFFRON toolbox provides the full source code of an example of the application of its tools to: i) the well-known TUC strategy [5]; and ii) to two recent decentralized signal control strategies proposed in [10]. These can be used to promptly compare the performance of novel solutions with the performance of these methods. Figs. 2 and 3 show the evolution of the link occupancy and green times, respectively, for the simulation of the D2TUC [10] strategy with decentralized configuration Φ using the SAFFRON tools in the Chania urban road network.

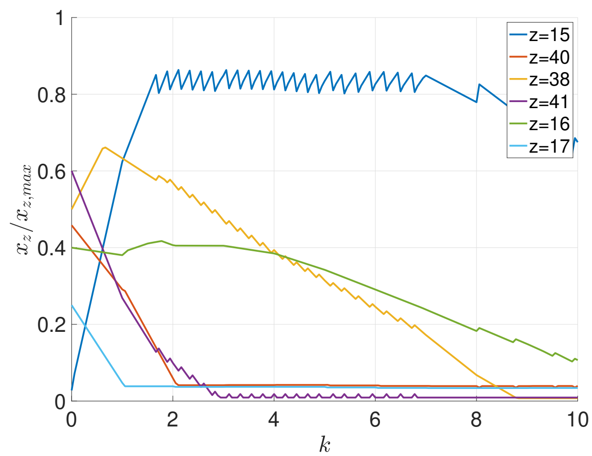

Fig. 2: Evolution of link occupancy of links related to junction 12.

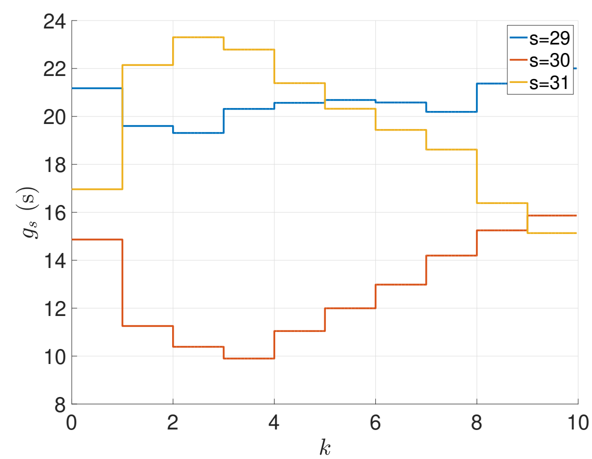

Fig. 3: Evolution of stage green times of junction 12.

The users of the toolbox are encouraged to share the implementation of their solutions, and make them available to the community. See heading *Contribute to SAFFRON* in the repository documentation for more information.

IV. Conclusion

In this paper the open-source SAFFRON toolbox was presented. The goal is to provide the community of Intelligent Transportation Systems with open-source tools, a model, and the implementation of state-of-the-art methods so that the effort required by new researchers in the field is decreased and the development of novel solutions becomes more transparent and easily reproducible. This toolbox will continue to be maintained and developed. The community is encouraged to contribute with suggestions, additions, and the implementation of innovative signal control strategies.

REFERENCES

[1] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package sumo," *RoboCup 2006*, 2006.

[2] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, and K. Nagel, "Matsim-t: Architecture and simulation times," in *Multi-agent systems for traffic and transportation engineering*. IGI Global, 2009, pp. 57–78.

[3] G. Tamminga, P. Knoppers, and J. Van Lint, "Open traffic: a toolbox for traffic research," *Procedia Computer Science*, vol. 32, pp. 788–795, 2014.

[4] D. C. Gazis and R. B. Potts, "The oversaturated intersection," Tech. Rep., 1963.

[5] C. Diakaki, "Integrated control of traffic flow in corridor networks," Ph.D. dissertation, Department of Production Engineering and Management, Technical University of Crete, 1999.

[6] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Store-and-forward based methods for the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 2, pp. 163–174, 2009.

[7] V. Dinopoulou, C. Diakaki, and M. Papageorgiou, "Applications of the urban traffic control strategy tuc," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1652–1665, 2006.

[8] K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos, "A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 680–694, 2010.

[9] A. Kouvelas, K. Aboudolas, M. Papageorgiou, and E. B. Kosmatopoulos, "A hybrid strategy for real-time traffic signal control of urban road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 884–894, 2011.

[10] L. Pedroso and P. Batista, "Decentralized store-and-forward based strategies for the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103412, 2021.

[11] R. Helgason, J. Kennington, and H. Lall, "A polynomially bounded algorithm for a singly constrained quadratic program," *Mathematical Programming*, vol. 18, no. 1, pp. 338–343, 1980.

[12] S. Baldi, I. Michailidis, V. Ntampasi, E. Kosmatopoulos, I. Papamichail, and M. Papageorgiou, "A simulation-based traffic signal control for congested urban traffic networks," *Transportation Science*, vol. 53, no. 1, pp. 6–20, 2019.