Toivio, Tuomas; Kosonen, Iisakki; Roncoli, Claudio

A multilayer optimisation framework for policy-based traffic signal control

Published in:
2020 Forum on Integrated and Sustainable Transportation Systems, FISTS 2020

DOI:
10.1109/FISTS46898.2020.9264870

Published: 03/11/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Toivio, T., Kosonen, I., & Roncoli, C. (2020). A multilayer optimisation framework for policy-based traffic signal control. In *2020 Forum on Integrated and Sustainable Transportation Systems, FISTS 2020* (pp. 347-352). Article 9264870 IEEE. https://doi.org/10.1109/FISTS46898.2020.9264870

A multilayer optimisation framework for policy-based traffic signal control

Tuomas Toivio, Iisakki Kosonen and Claudio Roncoli
Spatial Planning and Transportation Engineering,
Department of Built Environment, Aalto University, Finland

Abstract— **Traffic performance has many positive and negative consequences to the environment and society. These external effects are ever more often considered in the traffic system planning and administration. Desired effects of traffic can be thought as traffic performance policies. It is also possible to support these policies through traffic management and traffic signal controllers. In this study we introduce a general framework for a process flow which allows signalised junction controllers to adapt into desired policy. Also, we present an example implementation of the processes of the framework, and experiment with it by optimising a signal controller in a microscopic traffic simulation environment.**

I. Introduction

Traffic and transportation problems in cities are nowadays complex. When cities in 20th century generally started to grow rapidly together with increasing number of vehicles in roads, the traffic congestions started to emerge causing problems of several types. Negative impacts do not only concern solely mobility but also economical [1] and environmental [2] problems are induced by congestion.

There are diverse measures for adverse effects of traffic. Besides traditional travel time delay, it is quantitatively possible to consider e.g. air pollutants and noise caused by traffic. However, we still do not have any established way to incorporate and formulate traffic management policies out of diverse desires. As performance desires might also vary depending on dynamic circumstances in the traffic environment, we have a need for a procedure to formulate suitable management policies and embed them in the traffic management system.

There has been recent promising results that individual junction control strategies can have a reasonable influence in a network level [3]. As all the modern technology allows individual signal controllers to be highly dynamic and adaptive, in this paper we end up developing a framework for policy-based optimisation of a traffic signal controller (TSC).

For any control system the transparency of the controller logic is important for general acceptance [4]. When we want to unite policies and traffic management, it is good to notice that policy decisions come often from non-experts of the traffic engineering field. This emphasises the need to have understandable mapping between policies and their respective controller inputs.

*The research leading to these results has received funding from Henry Ford Foundation Finland (Henry Fordin Säätiö).

Schmöcker et al. [5] have experimented with policy-based signal control optimisation by uniting three entities: Bellman-Zadeh decision making [6] for policy setting, fuzzy logic based signal controller and genetic algorithms. They use the Bellman-Zadeh decision making (BmZ) as multi-criteria evaluator of the traffic performance during the optimisation. BmZ contains a user-defined mapping between a policy and measurable variables representing it.

This paper is tackling the question of having a general traffic signal control optimisation process that can adapt to arbitrary traffic performance policies. For that, the paper proposes an abstract framework, which could work as a hardware independent methodology for policy-based traffic signal control optimisation. To support the proposed methodology, we also construct an implementation of the abstract framework and study its performance through simulations. The implementation uses the ideas of policy optimisation process in [5] to adjust control parameters of FUSICO FUzzy SIgnal COntroller [7]. Our construction expands the frame of [5] by considering policy adjustment procedure. Thus, as a side product we introduce a policy-optimized adaptive fuzzy logic signal controller for an isolated junction. This controller will be optimised and evaluated by traffic simulations using real traffic data collected from the Tasman-Zanker intersection in San Jose, California, USA.

The content of the rest of this paper is as follows: Section II is the walkthrough of the abstract framework. Section III is about the implemented framework, which yields an example of policy-based signal control optimisation system. Section IV provides results of simulation experiments showing how the implemented system performs, and finally we make conclusions of this study in Section V.

II. Abstract framework for TSC optimisation

Here we describe our proposal for methodological structure for policy-based traffic signal control optimisation. It contains the principal structure behind many of the state-of-art signal control optimisation processes discussed in literature, but emphasises the existence of policies, which is not present in many advanced systems. The abstract framework does not exclude any particular paradigm of how the signal controller behaves, nor imply any certain optimisation method.

Fig. 1 presents our abstract framework: layers, acting blocks within layers, and interactions between all of them.

Next we present the layers and explain their interactions. We also discuss about the reasons to refer the presented parts as abstract and useful for any signal control optimisation system.

We identify three hierarchical loops of interaction in the abstract framework. These loops have inner cyclic interactions, and they interact also as a block of their surrounding loop. The innermost one is Control loop, which handles the experimenting of signal control parameters. Control loop acts as a part of Optimisation loop, which seeks to optimise the signal controller. The outermost is Policy loop, which guides the operations of the Optimisation loop and defines or re-defines policy mappings that are used in controller optimisations.

This system sketch is designed more for offline optimisation. The idea can still be utilised in online cases with running parallel offline optimisation processes and update the controller in action.

A. *Control loop*

The essence of the Control loop is the interaction between the signal controller and the traffic. Traffic observations summon requests for traffic control, and the signal controller responses, which in order affects the traffic behaviour. Nowadays a suitable simulation software is a standard way to experiment with the real case and practically every study developing signal control optimisations present simulation results in order to prove their concept to be plausible.

When the Control loop is simulated, the signal controller can be part of the simulation system or external. A capability for the simulator to work with external hardware enriches the flexibility of the full system [8]. In order to truly see the effects of the controller in variating traffic environments, traffic should be microscopic [8]. With macroscopic traffic rather simple scenarios can be evaluated and more advanced signal controllers using detailed detections cannot even be experimented with such traffic models. Also, in order to test a TSC in a real environment, we have to model the junction, its signal controller, and its traffic patterns in high precision, so that we could make statements about functionality of some proposed TSC. There are hardly any place one could test a new control paradigm without a priori showing some positive simulation results.

For the optimisation, the Control loop has to have an ability to measure various key performance indicators that might be used in evaluation of the controller or control policy and give them as an output.

B. *Optimisation loop*

Structure of the Optimisation loop is based on the assumption of learning the traffic behaviour and correcting the parameters incrementally. This works very well with controller paradigms that use artificial intelligence methods, such as Q-learning, neural networks or fuzzy logic [9]. Even though there are myriads of analytical models to obtain optimal signal controller without traffic simulations, they are not always so flexible with different policies, as their formulation is designed for a certain measurement, many of the time minimising the travel time or maximizing the throughput. In addition they usually advance macroscopic flow models, which makes them hard to adapt exceptional situations. Their advantage is the predictive power of the situation in the networks, but that still usually comes with cost of non-practical online computability [10][11].

Loop contains the block which reforms the parameter proposal. This is tested in Control loop after giving the control parameters to the signal controller. Control loop returns performance indicators that are used by a cost evaluation block, which states the acceptance of the trial parameters. This acceptance affects the further step taken by the parameter optimisation block. Number of iterations in the loop is dependent of a particular optimisation method.

C. *Policy loop*

The Policy loop of Fig. 1 is a top cycle, where policy tuning block interacts with the full optimisation process. The idea is that Optimisation loop system needs a certain policy input to produce an outcome for any given policy. Then these outcomes of different (simple) policies can be used to reformulate more complicated policies and to find balance between different policy criteria.

III. Implementation of the abstract framework

The abstract framework introduced in the previous section allows much freedom in its implementation. In Fig. 2 the blocks of Fig. 1 are substituted with suitable tools and methods that are revised in this Section. The Control loop is implemented with FUSICO software and the Optimisation loop uses genetic algorithm (GA) with BmZ in the cost function evaluation. Next we explain shortly these parts of the implementation and a simple method for policy tuning.

A. *FUSICO signal control and simulation software*

FUSICO [7] is a signal group oriented signal controller equipped with microscopic traffic simulation software. Signal group oriented means that there is an order of which the green phases should start, but the next green phase is always allowed to start whenever there are no conflicting active green phases. Extension of the green phase for individual signal group can utilise conventional actuated logic or a fuzzy logic extension module. When the latter is utilised, the extension of the green phase after minimum green time uses fuzzy inference. Length of the extension is given by fuzzy rules that use the demand from the approaching direction and the queue count from conflicting directions as inputs, and gives the length of the next extension as an output. In our case FUSICO fully covers the role of the Control Loop.

Parameters that will be optimised in the Optimisation Loop are tested and evaluated in the FUSICO simulation system in every round. In this implementation we only change the membership functions of the fuzzy extension modules, and take other signal controller parameters granted i.e. determined by other means. As fuzzy modules decide about the extension of green phases, the change of membership

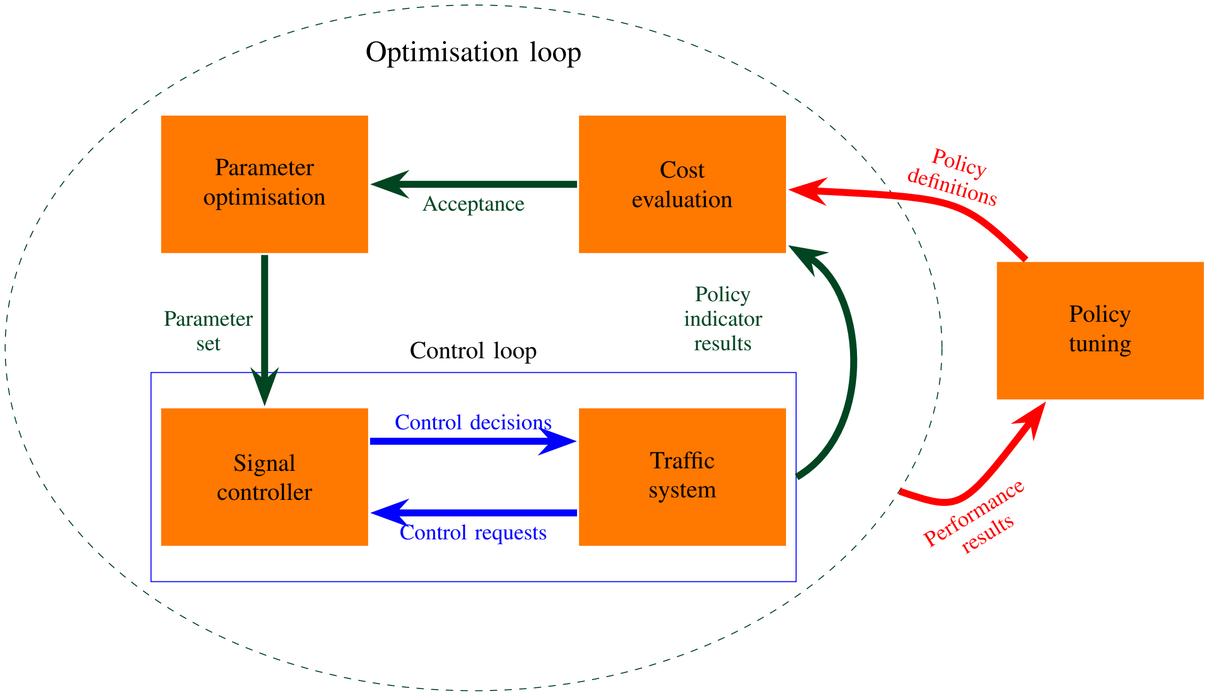

Fig. 1. Abstract optimisation framework for generic traffic signal controller. A comprehensive process can be modelled by Control, Optimisation, and Policy loops and their inner and outer interactions.

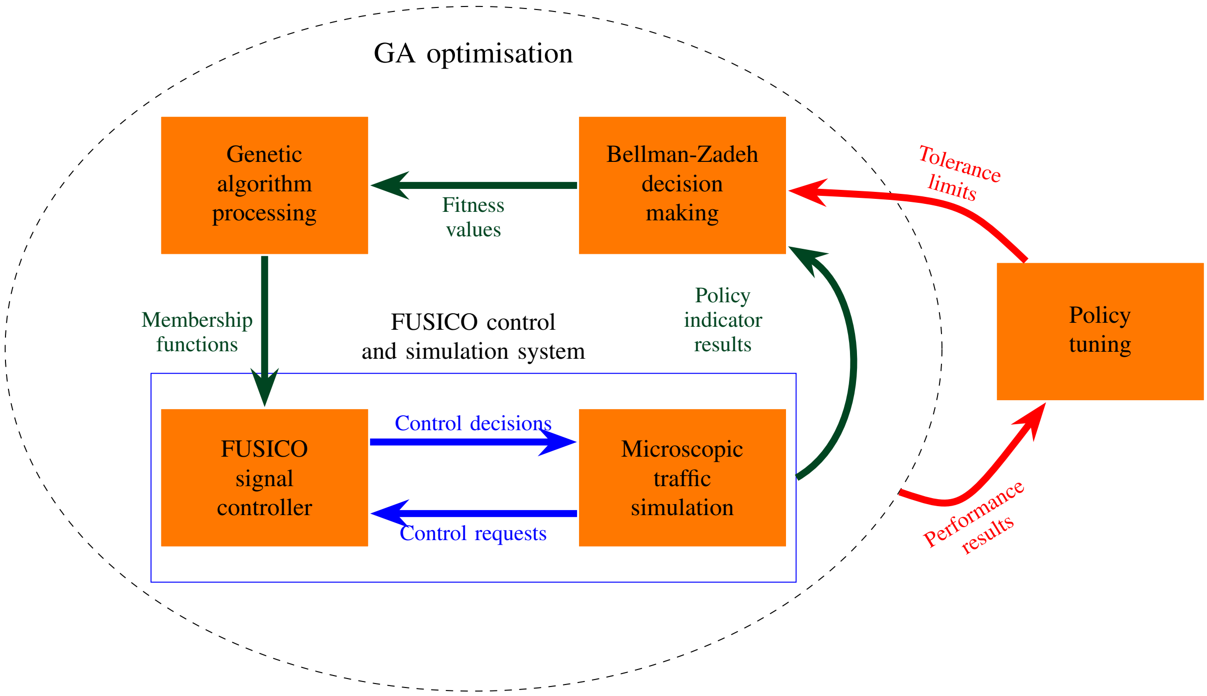

Fig. 2. Implementation of the blocks and interactions of the abstract optimisation framework for signal controller. This implementation is used in the experiments of this paper.

functions has a major effect in phase changing dynamics. Membership functions for all the fuzzy sets that are used in FUSICO rulebase are found in [12].

FUSICO produces many applicable performance indicator values for controller with a given set of parameters. These are transformed into policy indicators and handed to the Bellman-Zadeh decision making block to evaluate the parameter set (membership functions).

B. Genetic algorithm processing

The core part of a genetic algorithm is processed in this block, which means the formulation of new solution generation based on the fitness values of the previous generation of solutions. In our case a solution means a set of control parameters, which in turn refers to the membership function definitions used in the fuzzy inference of the green signal phase extension of FUSICO.

It is well-known that there are multitude application independent parameters in the genetic algorithm itself. In the conceptual level it is not necessary to specify them: they should be picked according to the complexity of the target parameters of the manipulation and sensitivity of the solutions and their performance. The parameters can even be dynamic. There is a long tradition of techniques to identify optimal parameters for GA itself e.g. [13]

Our implementation encodes the membership functions into binary strings, where each binary string represents the full set of membership functions representing 14 different fuzzy sets used by FUSICO rulebase. Detailed description of the encoding process can be found in [14]. The chosen encoding assures that linguistical partial order relations between fuzzy sets of the system make sense even after genetic manipulations are done inside this block. We conventionally call each membership function encoding as a chromosome during the GA loop process.

C. Bellman-Zadeh decision making

Bellman-Zadeh decision making (BmZ) evaluates the fitness of control solutions during GA optimisation loop. This means that it determines how supportive the FUSICO controller with certain membership functions is with respect to the pre-defined policy.

BmZ has two kinds of inputs, policy inputs and performance inputs:

1) Policy input: A policy input is given directly by the user or the policy tuning block. The policy input defines the aim of the GA optimisation loop, i.e. it determines the fitness function form for GA. The policy input is determined by two threshold values for each performance criterion used in the definition of a certain policy. The number of different criteria is unlimited in theory.

2) Performance input: A performance input results from the FUSICO simulation. It is a transformation of one or more output metrics of FUSICO into values of same performance criteria that are used to determine the optimisation policy. For example, we will later use average delay of vehicles and share of vehicles that has to stop as policy criteria.

Every chromosome of the GA loop process yields a performance input from FUSICO. It is used together with policy input thresholds to calculate the fitness value for the corresponding set of membership functions. The interpretation of thresholds values is that they determine 1) desired, and 2) tolerated value (denoted by y^F and y^0, respectively) for performance criteria whose actualisations come in as performance input. In Fig. 3 the connection between threshold values and performance inputs is illustrated.

Next we present carefully the fitness evaluation of BmZ block. BmZ first determines individual criterion values based on results provided by FUSICO. For every $x \in X$, where X is the set of chromosomes available, we define functions $y_i \colon X \to \mathbb{R}$, where $y_i(x)$ is the individual performance value of criterion i. Then for every criterion i, we denote policy input thresholds as y_i^0 (tolerated) and y_i^F (desired).

We define a continuous function

$$C_i \colon \mathbb{R} \to [0,1]$$

to be such that $C_i(y_i^0) = 0$ and $C_i(y_i^F) = 1$ and that it grows or decends linearly between the thershold values, and is constant otherwise (see Fig. 3). We call C_i as the *fitness function* of criterion i. We can now write a fitness formula for a chromosome x and criterion i as

$$C_i(y_i(x)) = \max\left\{0, \min\left\{1, \frac{y_i^0 - y_i(x)}{y_i^0 - y_i^F}\right\}\right\}. \qquad (1)$$

Finally, the total *fitness* function $D \colon X \to \mathbb{R}$ defined as

$$D(x) = \min\{C_1(y_1(x)), \ldots, C_n(y_n(x))\}, \qquad (2)$$

gives the fitness of chromosome x, where n is the number of criteria used in definition of the optimisation policy.

These fitness values are assigned to the corresponding chromosome and handed to the GA algorithm processing block in order to formulate the next generation of chromosomes. The point of BmZ is to balance between the fitness function values of each criterion by maximising the minimum of criterion fitness values. Hence, GA optimisation tries to maximise D.

Anderson et al. [15] have investigated the suitability of multi-objective GA to tweak multi-criteria fuzzy system applied with signal control. They concluded the approach to be appropriate. However, a normal multi-objective GA optimisation produces a curve of solutions known as Pareto-optimal front, and the final result has to be choosed by some other means.

In our case BmZ allows the GA algorithm core to be single-criterion as BmZ produces one single fitness value

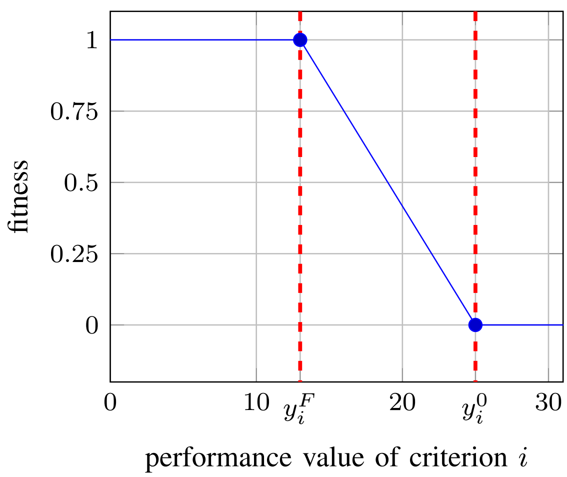

Fig. 3. Illustration of how fitness value function for individual performance criterion is constructed by given threshold values. Blue line gives the fitness function, and red dashed lines indicate threshold values given by policy.

based on given policy, which can still result from multiple criteria. Therefore GA with fitness evaluation by BmZ aims towards an unique point in the Pareto-front, which is gained by finding a solution x that maximises the value of $D(x)$.

D. Policy tuning

With one policy criterion it is fairly easy to set the thresholds y_i^0 and y_i^F based on desired outcome. Finding a magnitude of feasible values can be obtained by running some example simulations with arbitrary controller.

When creating multi-criteria policy in this system, one needs thresholds for every criterion. First step in tuning is to discover optimal values for single-criterion policies and at the same time observing how they hinder the performance values used by other criteria. The initial mixed experiment can be formed by using optimal values as the desired thresholds, and the most hindered values as tolerated.

Detailed tuning process demands several optimisation rounds. After the initial round the process can be continued e.g. by incremental changes to the thresholds in order to change the balance point between the criteria.

IV. Experiments and Results

A. Experimental scenario

Performance of the implemented optimisation framework described earlier is evaluated in the real world simulation scenario. We have the junction model of Tasman Drive and Zanker Road from San Jose, California, USA. Detections of approaching traffic were collected lane-by-lane from every junction leg by microwave radars. Installation was part of the project to investigate how to make traffic in the area more efficient with the FUSICO controller.

For the experiments we defined three different policies for BmZ block of the optimisation loop (GA process). Goal of the optimisation for every policy was to find optimal membership functions for fuzzy logic based green extension of the TSC. Training rounds of the optimisation used all the traffic between 6 – 22 of a representative weekday. Results presented here are average performance metrics of the traffic of four separate representative weekdays. More detailed description of the scenario can be found in [14].

First, we have made simulation experiments with two simple policies. Level of Service (LoS) policy seeks to minimise average delays caused by the intersection controller, where as Environmental (Env) policy tries to minimise amount of vehicles that has to stop during their journey through the junction. Second, we introduce a simple way to implement Mixed policy out of the two simple ones and compare its results against the simple policy outcomes. This gives an example of the Policy tuning block process.

Experiments aim to show that this implementation of the abstract framework can improve a TSC performance corresponding to the given policy. This in turn supports our goal to see that the abstract framework introduced in section II provides a prominent skeleton for a methodology to optimise signal control performance under arbitrary policies.

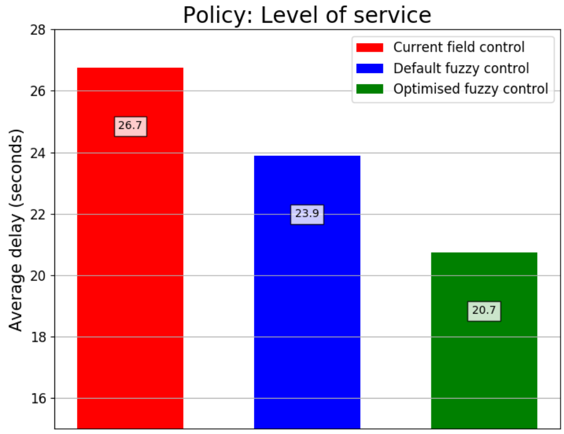

Fig. 4. Average delay comparison with Tasman-Zanker field controller (Vehicle actuated), default FUSICO controller and FUSICO optimised with LoS policy. Result shows that controller can well adapt to LoS policy.

TABLE I
Mixed policy thresholds derived from LoS and Env policy results.

Threshold	Value
Stops not acceptable	70.0 %
Stops fully acceptable	67.6 %
Delays not acceptable	25.4 s
Delays fully acceptable	20.7 s

B. Simulation results: single criterion policies

In order to show the performance of different optimisations, every resulting figure compares the optimised FUSICO controller against the control program using ”default” membership functions, and against the vehicle actuated controller logic of the Tasman-Zanker junction. Here default refers to membership functions that were constructed for FUSICO in earlier research projects [7].

In Fig. 4 LoS policy has been used in optimisation. There is a clear improvement in average delays achieved by optimisation compared to the other control options. Fig. 5 shows similarly how the FUSICO optimised with Env policy lowers the share of vehicles stopping in the junction. We can notice that the controller with default fuzzy logic membership functions works better than the controller in the field, and the optimisation process outputs have clearly adapted both of the two policies.

C. Simulation results: Mixed policy

Following the process described in section III-D, we find the initial BmZ criterion thresholds for a Mixed policy. With these thresholds optimisation should result a compromise between LoS and Env policies. The thresholds applied here are given in Table I.

Fig. 6 shows how all the considered policies, LoS, Env and Mixed perform in the same circumstances after the fuzzy extension modules of the TSC are optimised according to

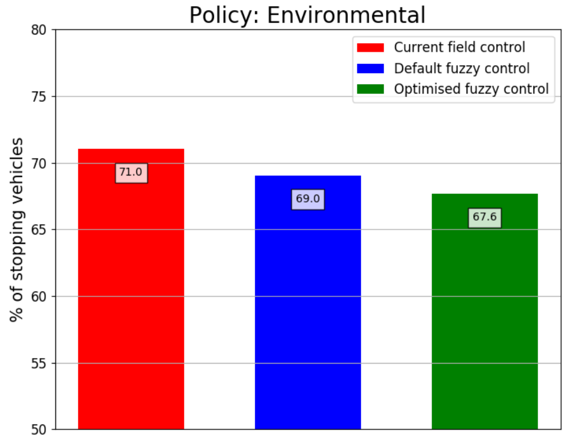

Fig. 5. Stopping share comparison with Tasman-Zanker field controller (Vehicle actuated), default FUSICO controller and FUSICO optimised with LoS policy. Result shows that the Env policy optimisation has an effect to the controller performance.

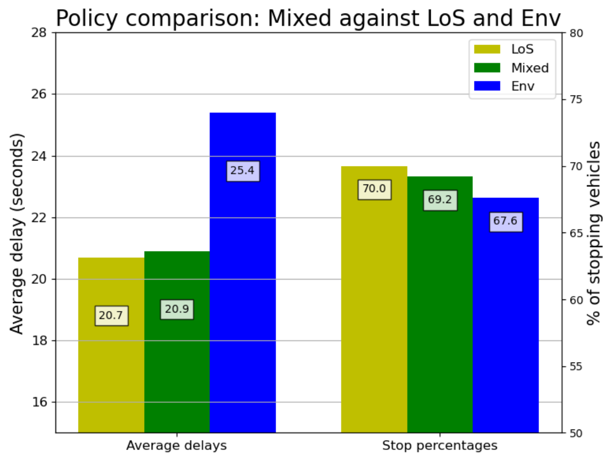

Fig. 6. Mixed policy performance is compared with LoS and Env policies. Bars on the left show average delays and bars on the right stopping share.

the respective policy. We see that our simply defined Mixed policy manages to balance between the achievement gained by single-criterion optimisations that clearly are competitive between each other. Obviously, we could have some other tailored way to define multi-criteria policies, which would yield a different kind of balance.

V. Conclusions

This paper aimed to formulate general traffic signal control optimisation framework that would allow policy-based thinking in determining any desired junction performance. We introduced a proposal for an abstract framework to act as a hardware-independent TSC optimisation methodology. We also made an implementation of it, which was tested in the real world based simulation scenario.

Results show that our example implementation of the abstract framework can optimise a TSC accordingly to the given policy definition. This gives evidence that the proposed abstract framework could be generally valid methodology for policy-based traffic signal control optimisation.

The abstract framework is a conceptual tool, and there are myriads of ways to construct an implementation of it. Those methods can be studied and developed independently. Particularly a more systematic and reasonable way to make policy tuning needs much further research. Also, the question of translating possibly non-numerically introduced policies into optimisation system is important from transparency perspective (in the implementation of this paper that is, how threshold values for BmZ are formulated and what metrics are used).

We conclude that our proposal for abstract TSC optimisation framework is a promising opening to incorporate policies more tightly into traffic management in the future.

References

[1] R. Arnott and K. Small, "The economics of traffic congestion," *American scientist*, vol. 82, no. 5, pp. 446–455, 1994.

[2] K. Zhang and S. Batterman, "Air pollution and health risks due to vehicle traffic," *Science of the total Environment*, vol. 450, pp. 307–316, 2013.

[3] B. Alonso, Ángel Ibeas, G. Musolino, C. Rindone, and A. Vitetta, "Effects of traffic control regulation on network macroscopic fundamental diagram: A statistical analysis of real data," *Transportation Research Part A: Policy and Practice*, vol. 126, pp. 136 – 151, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965856418303665

[4] M. N. Mladenovic, A. Stevanovic, I. Kosonen, and D. Glavic, "Adaptive traffic control systems: Guidelines for development of functional requirements," in *International Scientific Conference on Mobility and Transport (mobil. TUM)*, 2015.

[5] J.-D. Schmöcker, S. Ahuja, and M. G. Bell, "Multi-objective signal control of urban junctions–framework and a london case study," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 4, pp. 454–470, 2008.

[6] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management science*, vol. 17, no. 4, p. B141, 1970.

[7] J. Niittymäki, "Fuzzy traffic signal control: principles and applications," Doctoral dissertation, Publication no. 103 of the laboratory of transportation engineering, Helsinki University of Technology, 2002.

[8] D. Bullock, B. Johnson, R. B. Wells, M. Kyte, and Z. Li, "Hardware-in-the-loop simulation," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 1, pp. 73–89, 2004.

[9] S. Araghi, A. Khosravi, and D. Creighton, "A review on computational intelligence methods for controlling traffic signal timing," *Expert systems with applications*, vol. 42, no. 3, pp. 1538–1550, 2015.

[10] S. Göttlich, M. Herty, and U. Ziegler, "Modeling and optimizing traffic light settings in road networks," *Computers & Operations Research*, vol. 55, pp. 36–51, 2015.

[11] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, 2012.

[12] J. Niittymäki, "General fuzzy rule base for isolated traffic signal control-rule formulation," *Transportation Planning and Technology*, vol. 24, no. 3, pp. 227–247, 2001.

[13] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on systems, man, and cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.

[14] T. Toivio, "Adaptive framework with policy setting tool for fuzzy logic based traffic signal controller," Master's thesis, Aalto University, 2018.

[15] J. M. Anderson, T. M. Sayers, and M. Bell, "Optimisation of a fuzzy logic traffic signal controller by a multiobjective genetic algorithm," in *Ninth International Conference on Road Transport Information and Control*. IET, 1998, pp. 186–190.