MATIJA GULIĆ M.Sc.[1]
(Corresponding author)
E-mail: matija.gulic@fer.hr
RICARDO OLIVARES, B.Sc.[2]
E-mail: ricardo.olivares@alumnos.uc3m.es
DANIEL BORRAJO, Professor, Ph.D.[2]
E-mail: dborrajo@ia.uc3m.es
[1] University of Zagreb,
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
[2] University Carlos III of Madrid
Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain

Intelligent Transport Systems (ITS)
Preliminary Communication
Submitted: Sep. 22, 2015
Accepted: May 17, 2016

USING AUTOMATED PLANNING FOR TRAFFIC SIGNALS CONTROL

ABSTRACT

Solving traffic congestions represents a high priority issue in many big cities. Traditional traffic control systems are mainly based on pre-programmed, reactive and local techniques. This paper presents an autonomic system that uses automated planning techniques instead. These techniques are easily configurable and modified, and can reason about the future implications of actions that change the default traffic lights behaviour. The proposed implemented system includes some autonomic properties, since it monitors the current traffic state, detects if the system is degrading its performance, sets up new sets of goals to be achieved by the planner, triggers the planner that generates plans with control actions, and executes the selected courses of actions. The obtained results in several artificial and real world data-based simulation scenarios show that the proposed system can efficiently solve traffic congestion.

KEY WORDS

road management system; traffic signal control; autonomic system; automated planning;

1. INTRODUCTION

With economy growth, the number of vehicles is increasing in many countries. Newly built road networks increase the capacity, but in some cities (and in the future in many more cities), building new roads will no longer be possible. Hence, road networks have difficulties when trying to adapt to needed demands (number of vehicles). One of the most popular ways to partly alleviate this problem consists in increasing capacity by better control of road flow; that is, using traffic lights and similar actuators. Traffic control consists of methods for monitoring traffic (sensors, vehicle detection on images, communication among vehicles), methods for processing available data and generating commands that can change the traffic state, and finally using traffic lights or other methods to control the traffic flow.

In order to reduce traffic congestions many researchers have proposed different approaches to address traffic lights control [1]. Early systems implemented fixed-time plans that allow for the generation of "green waves" by coordinating a subset of related traffic lights. Their main disadvantage was that their behaviour was static, so they could not react to incidents. New techniques can react to traffic incidents. Traffic-responsive control techniques are probably among the most used and studied methods for adaptive traffic signal control [2, 3, 4]. There are two main alternatives: centralized approaches, as SCOOT [5]; and distributed approaches, as UTOPIA [6]. In these classical approaches, in case new sensors/actuators are incorporated into the control system, or new metrics (or their combinations) have to be considered, the control programs have to be carefully modified to account for the new components, requiring a big engineering effort. Other approaches consider the use of genetic algorithms [7], classical optimization techniques [8, 9, 10, 11, 12, 13, 14] or Stackelberg-Nash and similar game theory approaches for traffic control [15, 16, 17, 18]. Due to the nature of traffic, fluctuations and unpredicted events are always present, which could lead to decreased control accuracy. Some approaches dealt with uncertainty by combining Model Predictive Control (MPC) with manually defined uncertain terms in the control equations [19].

The goal of this research is to provide an intelligent and autonomic system based on automated planning techniques [20]. The system can autonomously: monitor the current state of the traffic; detect if the system is degrading its performance by checking the traffic density at each street section; in case the density increased in one or more sections, set up a new set of goals to decrease those densities; call the automated planner that will generate a sequence of actions to control traffic lights in a given time frame; and execute the selected actions. The execution of the generated plans can expand over a period of time.

An automated planner is used as the core problem solver. A planner takes as input two descriptions: a domain file that mainly specifies a set of actions an agent can take; and a problem file that mainly specifies an initial state and a set of goals to be achieved. Both files are usually provided in the PDDL (Planning Domain Description Language), a standard language used by the planning community [21]. The planner should generate a plan as output that consists of a sequence of actions that allow the system to transit from the initial state to a state where the goals have been achieved. One key characteristic of automated planners is that they are mostly domain-independent. Therefore, the same code (planner) can solve logistics [22], tourism [23] or rovers [24] tasks given the declarative input provided in the domain and problem files. Automated planning can also tackle tasks which include reasoning with time, uncertainty, costs, and expressive action representations.

By using automated planning, the proposed system can anticipate future high densities due to longer-than-usual green times generated by previous control actions. Furthermore, the method is global in the sense that it considers the city (or a city section) as a whole. So, all traffic signals are controlled in a centralized way. In contrast to most previous works, the automated planning control program is defined declaratively in a high-level programming language, PDDL. Thus, it is relatively simple to include new control actions, sensoring information, or new optimization metrics in the PDDL descriptions, according to new requirements by authorities. To the best of the authors' knowledge there is no prior work that addresses this task using classical Automated planning and a full planning-execution cycle.

Section 2 describes the architecture of the autonomic system that includes the planner and the simulator used in the experiments, SUMO. It also describes in some detail each of the modules that compose the architecture. Section 3 presents experiments that show the benefits of using the proposed system. Finally, Section 4 provides conclusions and suggests the future work.

2. ARCHITECTURE OF THE PROPOSED SYSTEM

Architecture of the proposed system is composed of the following modules: Simulator, Planner, and Intelligent Autonomic System (IAS), composed of Monitor and Execution (*Figure 1*). The central module of the system is the Intelligent Autonomic System (IAS), which cooperates with two supporting systems (Simulator and Planner). IAS starts the simulation, and simultaneously triggers the monitoring. The Monitor communicates directly with the *SUMO* simulator. Based on the current simulation state, it detects possible traffic incidents. Once an incident is recognized, a problem file for the planner is automatically created, and the planner is called to solve the problem for the specified goals. After the planner generates a plan, the system converts the plan to a set of actions which are then used to control traffic simulation. Due to the modularity of the architecture, any component can be easily substituted for another one. For instance, given that a standard PDDL input is used, it is possible to substitute the planner used in the current system for any other planner. Each component will be now described in more detail.

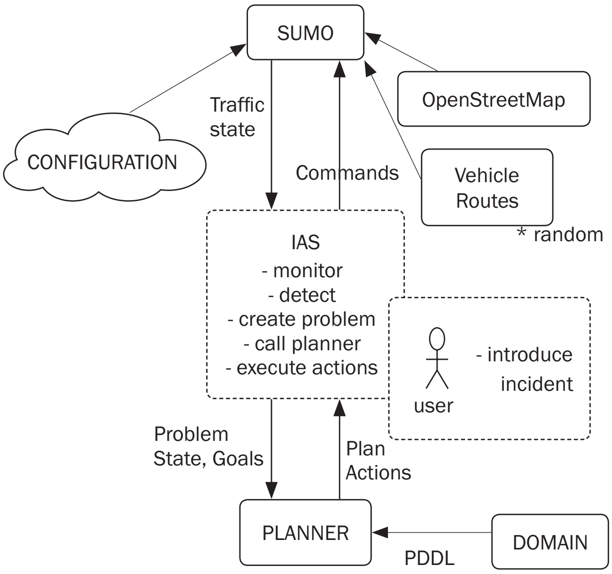


Figure 1 – Architecture of the proposed system

2.1 Simulator

The selected traffic simulator for this research is Simulation of Urban Mobility (*SUMO*) [25] for traffic simulation, given that it is an open source and a powerful software [26, 27]. *SUMO* includes vehicular communication [28], route choice and navigation [29], traffic lights algorithms [30], as well as emission and noise modelling [31].

The main inputs to the simulator are the road network file (*net*) (streets, traffic lights, speed limits), the routes file (each vehicle has a defined route) and the configuration file (gui/no-gui version, other needed files paths). The *net* file can be built using the tools and scripts provided by *SUMO*. The initial step is to create the *net* file for any city or area. One useful alternative is to download an OpenStreetMap [32] (*osm*) file (possibly using the import *osm* tool provided by *SUMO*). Once the *osm* file is available, a set of flags (options) is used to prepare the needed *net* file.

There are two main options to generate traffic (vehicles). The first one is to provide an origin-destination matrix and then the best possible routes will be calculated by *SUMO*. The second approach uses random routes, which means vehicles can start their route anywhere and finish at any random point.

Controlling simulations in *SUMO* is possible using the Traffic Control Interface (TraCI) [33]. TraCI uses a TCP based client/server architecture to provide access to simulations, online retrieving values for multiple simulated objects (vehicles, lanes, traffic lights) and manipulating their behaviour. It is available for several programming languages, such as Python, Java, .NET family languages and Matlab.

2.2 Planner

As explained before, the automated planner requires two PDDL [34] files in order to generate a set of actions: the domain and the problem. The domain file consists of predicates and actions. The predicates defined in developed application are related to the network structure, traffic lights state and traffic conditions. Actions were designed in line with the broad range of situations encountered on an intersection of two streets. They manage crossings with two and three exit streets. Every action has a set of preconditions (conditions that should hold to execute the action) and effects (expected changes in the state after applying the action). *Figure 2* shows an example of an action model with its set of preconditions and effects.

Figure 2 shows an action model for setting green light in traffic entering intersections from one of the street directions. The parameters specify the variables that will be used in the action description. The preconditions represent the conditions that should hold in order to be able to select the action for execution. In this case, they show what the intersection model looks like: some streets enter the intersection; some only exit; traffic lights are on all streets; and some checks on the density on the input and output street sections. In this action all output streets need to have low density, and the input street should have high density. The action will open the green light for a longer period, so the expected effect is that traffic entering the intersection will have lower density, and streets exiting the intersection will then have an increased density (standard traffic flow).

The problem file consists of objects, which could be found in the traffic simulator model, the description of the initial state, and the definition of the goals. In the current system, the goals are to reduce traffic density of streets from high or very-high density to low density. In the case of reducing density, the system is expected to reduce traffic jams, total travel time and CO2 emissions. This is not always true. For instance, generation of emissions is a highly complex process that depends on many other variables [35]. Since adding goals is an easy modelling task, one could define new types of goals such as having low density in a set of predefined street sections as when the goal is to free a given route from traffic, or lower the traffic density of some area due to noise levels. *Figure 3* shows a simplified example of a problem file, with the objects involved, the initial state and the goals. It defines three types of objects (street, crossing and traffic light). These objects are used, together with defined predicates, to define the traffic initial state. The predicates are used to define the relations among objects. For instance, the goes-into predicate is used to define which street is entering each crossing. Similar predicates are used to define other relations (opposite direction or density). Finally, there is a list of goals which need to be achieved. In this example, the goals are to decrease the density levels for the streets with high density.

Now, any PDDL complaint planner can be executed (there are many available from the International Planning Competition, IPC [36]) and it will generate a plan as the one shown in *Figure 4*. More specifically, we have used the Fast-Downward code, since planners built from it, as the LAMA planner [37], have shown

```
(:action m-green-to-all-ways
        :parameters (?t - traffic-light ?c - crossing ?sin - street ?sout1 - street ?sout2 - street ?sout3 - street)
        :precondition (and (goes-into ?sin ?c) (goes-out ?sout1 ?c) (goes-out ?sout2 ?c) (goes-out ?sout3 ?c)
                           (not (= ?sout2 ?sout1)) (not (= ?sout2 ?sout3)) (not (= ?sout3 ?sout1))
                           (traffic-lights-from-street ?t ?c ?sin) (traffic-lights-to-street ?t ?c ?sout1)
                           (traffic-lights-to-street ?t ?c ?sout2) (traffic-lights-to-street ?t ?c ?sout3)
                           (not (densityLevel ?sout1 very-high)) (not (densityLevel ?sout1 high))
                           (not (densityLevel ?sout2 very-high)) (not (densityLevel ?sout2 high))
                           (not (densityLevel ?sout3 very-high)) (not (densityLevel ?sout3 high))
                           (densityLevel ?sin moderate))
        :effect (and (not (state-to-street ?t ?sout1 red)) (not (state-to-street ?t ?sout2 red))
                     (not (state-to-street ?t ?sout3 red)) (state-to-street ?t ?sout1 green)
                     (state-to-street ?t ?sout2 green) (state-to-street ?t ?sout3 green)
                     (not (densityLevel ?sin moderate)) (densityLevel ?sin low)
                     (not (densityLevel ?sout1 moderate)) (not (densityLevel ?sout1 very-low))
                     (densityLevel ?sout1 low) (not (densityLevel ?sout2 moderate))
                     (not (densityLevel ?sout2 very-low)) (densityLevel ?sout2 low)
                     (not (densityLevel ?sout3 moderate)) (not (densityLevel ?sout3 very-low ))
                     (densityLevel ?sout3 low)))
```

Figure 2 – Example of an action model with its preconditions and effects

```
(define (problem traffic1) (:domain traffic)
(:objects
            s_103371696#0 s_103371696#1 ... s_54044824#4 - street
            c_1182103183 c_1193695620 ... c_153378749 - crossing
            tl_152121796 tl_152121799 ... tl_152736084 tl_152736088 - traffic-light)
(:init
            (goes-into s_103371696#0 c_152736084)
            (goes-into s_103371696#1 c_152736088)
            ...
            (goes-out s_103371696#0 c_1193695620)
            ...
            (opposite-direction s_:152121796_0 s_:152121796_1)
            (traffic-lights-from-street tl_152736084 c_152736084 s_103371696#0)
            ...
            (traffic-lights-to-street tl_152700480 c_152700480 s_103371696#4)
            ...
            (state-from-street tl_152736084 s_103371696#0 red)
            (state-to-street tl_152419819 s_54044824#4 green)
            (densityLevel s_103371696#0 high)
            ...)
(:goal
    (and
            (densityLevel s_103371696#0 low)
            (densityLevel s_54044824#3 low) ...)))
```

Figure 3 – Example of a problem file

```
Actions executed in step 503:
h-green-to-two tl_152121796 c_152121796 s_105234280#6 s_105234280#7 s_297982417#2 s_103371696#0
l-green-and-red-to-one tl_152736088 c_152736088 s_297810848#0 s_297810848#5 s_103371696#2
                              s_103371696#0
h-green-to-two tl_152121799 c_152121799 s_105234280#5 s_105234280#6 s_124875319 s_103371696#0
h-green-to-two tl_152121802 c_152121802 s_105234280#4 s_105234280#5 s_124875327#3 s_103371696#0
h-green-to-two tl_152121805 c_152121805 s_298579938#8 s_105234280#4 s_298579938#9 s_103371696#0
h-green-to-two tl_152700480 c_152700480 s_103371696#3 s_103371696#4 s_298579938#8 s_103371696#0
l-green-and-red-to-one tl_152419816 c_152419816 s_124875327#1 s_124875327#2 s_54044824#3
                              s_103371696#0
h-green-to-two tl_152370738 c_152370738 s_297810849#3 s_124875327#1 s_297810849#4 s_103371696#0
Actions to be executed in the future:
hm-green-to-two tl_152736088 c_152736088 s_103371696#1 s_103371696#2 s_297810848#5 s_103371696#0
h-green-to-two tl_152419816 c_152419816 s_54044824#2 s_124875327#2 s_54044824#3 s_103371696#0
```

Figure 4 – Example of a generated plan

a remarkable performance in the past IPCs (2008, 2011). *Figure 4* shows a plan that contains the actions *h-green-to-two and l-green-and-red-to-one*, among others. Some actions will be executed at the time when the planner is executed (step 503), while the last two actions will be executed in the later time steps. Each action is related to a specific object (traffic light). The actions in this plan are the ones that the planner selected to achieve all goals (reducing traffic density). More precisely, actions have parameters that identify the particular instances of traffic lights, crossings and streets. Executing these actions will control the traffic lights.

One of the main advantages of using a planner as part of an autonomic road system is the possibility to abstractly and declaratively model complex actions. Complex actions in PDDL can be easily described to control a set of crossings. And, given that PDDL is a declarative language, this file can be autonomically tuned by a learning system, once its behaviour is observed in the real world [38].

2.3 Intelligent Autonomic System

The Intelligent Autonomic System (IAS) performs plans monitoring and execution. The Monitoring module task is to detect incidents and notify the Execution module to proceed with appropriate actions. An incident is defined when one or more vehicles are stopped for a long period of time. The Execution module task consists of compiling the current traffic state in a problem file for the planner only if no previously computed plan exists. The problem file is an abstraction of the current traffic state and the definition of desired goals (change traffic density of streets with high density into low density). The Execution module sets as goals not only to lower the traffic density of the congested streets, but also to lower the traffic density of the streets to which it predicts that the traffic is going to be diverted from the congested ones. Therefore, the generated plan (by the planning module) is divided into a plan to be executed at the same step as the planner (current plan) and another plan to be executed later (future plan). Finally, it translates the actions from the current plan to a set of control actions which are sent back to the *SUMO* simulator. The actions in the current plan are performed and the rest of them are kept in case they may be executed in the next interaction (after a 50-step period). If so, it means that the prediction made by the system about the traffic behaviour after executing the actions is accurate. If an action or several actions from an existing plan can be executed, the IAS does not trigger a new planning episode. Otherwise, the planner will be called again until the incident is solved.

An additional ability of the IAS module is to introduce incidents into the system. Incidents can be generated based on three options: stopping a vehicle for a selected period of time; changing the speed limit for some street; and turning red lights on a selected crossing. These possibilities have been implemented primarily to create scenarios which could be used for testing the proposed system. Secondarily, this functionality could be used for simulations of the current state of a road network. Many "what-if" critical scenarios could be generated and results can be used for enhancing the current road network. These simulations are not part of this research but they are a good point for leading future research in many different directions.

2.4 Monitoring module

The Monitoring module is the basic component which overviews the current simulation (or real world data in case the system would be fed with such data). It can recognize situations which could lead to creation of traffic congestions. The definition of "problematic" traffic state could be expressed in several ways, such as: duration of a stopped vehicle, number of vehicles in one traffic lane, or high traffic density. In this paper, vehicle stop duration is used as a method to predict potential problematic states.

Once the simulation starts, the Monitoring module "computes" every 50 simulation steps (seconds) how long it has taken since the last notification that the traffic system is in a state which requires a control action. If an incident is recognized or there is a future plan previously computed, the system starts the control action process.

2.5 Execution module

When a problem is detected in the traffic, the current traffic state from *SUMO* has to be abstracted and represented in a format that can be used by the planner. Therefore, the module has to create literals that relate to the network structure and some traffic related conditions. Also, it has to generate the goals that the planner will try to achieve. In the current implementation, they are related to reducing traffic density on the streets with the highest density and on the streets to which the traffic is going to be diverted according to the traffic predicted by the system. Then, the planner is called with the fixed domain file and the recently generated problem file. After the planner generates the plan, it consists of actions that must be converted into actions that can be understood and further controlled by the simulation. Only the actions in the current plan will be executed. Additional time is provided to the simulator to achieve the results from the applied actions, and it monitors again the traffic state, checking if the actions from the future plan can be executed or if a new cycle of planning-execution has to be triggered.

3. CASE STUDY

In order to carry out the experiments, several different configurations were defined varying the road network and relevant parameters. First the experimental setup will be defined and then some results and the discussion presented.

3.1 Experimental setup

First, a road network has to be selected which will be used for simulation. Artificial scenarios and a small area in the Downtown Houston have been set as a real scenario. This area was selected using the *SUMO* tool *server.py* for importing OSM maps [39]. The two artificial scenarios are grill-shaped with 35 and 130 crossings, respectively, each crossing with several traffic lights. The number of vehicles used in the real scenario simulation was 917. In the artificial scenarios 2,250 and 5,000 vehicles were used. Random routes were generated for all vehicles. And vehicles start their travel at random times during the simulation. The selected road network can be seen in *Figure 5*.

The simulation requires other configuration parameters, such as Simulation time (5,000 steps) and the *lanechange.allow-swap* option that is set to True (in order to allow vehicles to change the lane according to traffic conditions and their destination).

Additional parameters for the IAS module are the threshold for stop duration of a vehicle to consider it an incident (120 steps), and the threshold of the number of vehicles stopped for long before generating an incident (1 vehicle). Several scenarios were used, and the total duration until all vehicles reach their destination has been measured. The second reported variable is the amount of CO_2 emissions all the vehicles emit during the simulation.

The following section compares running the simulation without using monitoring (no interventions) against IAS (using monitoring and planning to solve incidents).

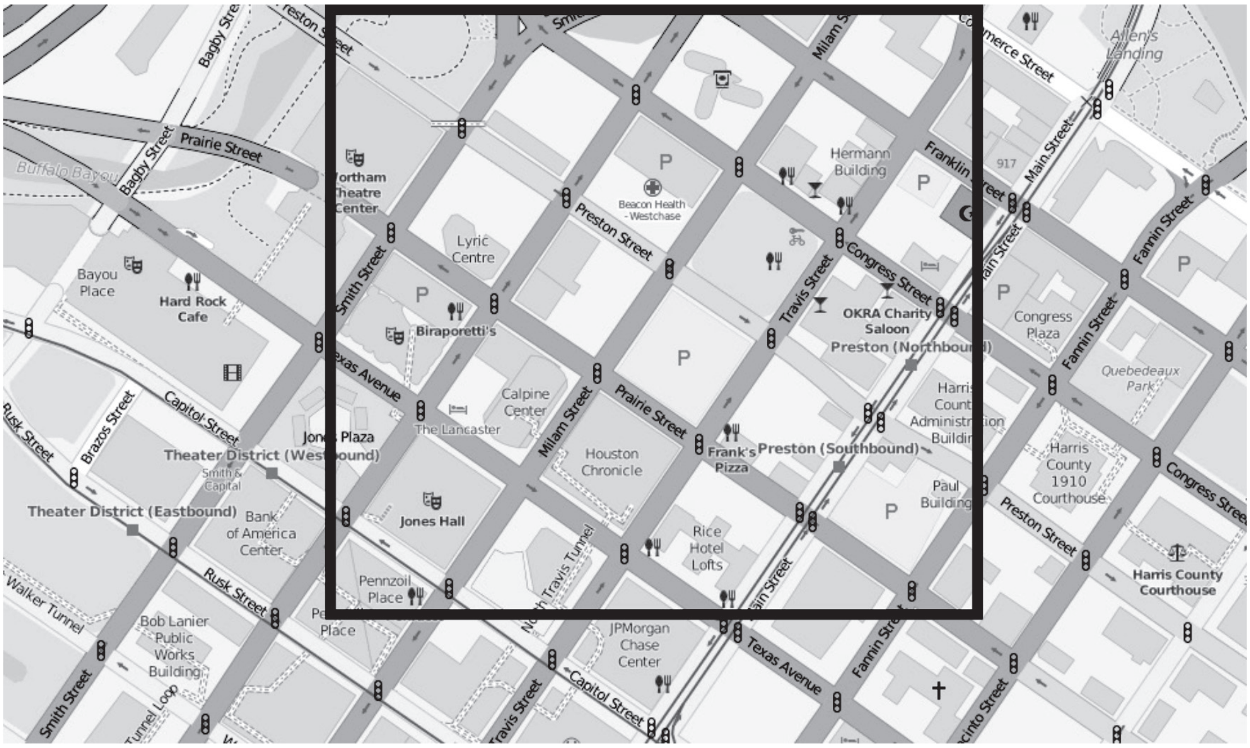


Figure 5 – Downtown Houston

3.2 Results and discussion

In the case of running the *simulation with artificial urban networks*, IAS outperforms the non-monitored system. When the traffic becomes more intensive than the normal one, traffic jams appear and IAS is able to manage the traffic and solve incidents, whereas when we do not use IAS the road network is not able to spread out the traffic, causing a congested net and some vehicles never reach their destination (cases 1, 2, 3, 4 and 8 in *Tables 1* and *2*). On the other hand, when traffic is not so intense, both systems solved the incidents approximately with the same number of steps (cases 5, 6 and 7). More detailed results for artificial urban networks are provided in *Table 1* and *Table 2*. *Table 1* shows that there is a significant number of vehicles which were unable to finish their route. However, in case IAS is used, all vehicles are able to finish their journey under 5,000 steps. *Figure 6* shows the detailed CO2 emissions for each artificial scenario when using or not the proposed system (IAS).

Figure 7 shows the average travel time with and without using IAS. In five out of eight scenarios IAS significantly outperforms the default system and IAS performance is very similar to the default system in the remaining cases. *Table 3* compares the vehicle waiting time for the two approaches. In cases with higher travel time, the IAS system outperforms the default system. As IAS is a deliberative system, it recognizes potential incidents and executes actions to mitigate future incidents and that leads to reduced waiting time.

Table 1 – Results on the artificial urban network without IAS

Case	Traffic lights	Vehicles	Vehicle generation time	Total steps	Remaining vehicles	CO_2 emission (Kg)
1	35	2,250	900	5,000	413	3,179.76
2	35	2,250	450	5,000	585	4,537.98
3	35	5,000	900	5,000	1,090	7,770.96
4	35	5,000	450	5,000	307	2,761.08
5	130	2,250	900	1,269	0	524.34
6	130	2,250	450	1,270	0	573.43
7	130	5,000	900	2,222	0	1,467.63
8	130	5,000	450	5,000	1,177	9,239.55

Table 2 – Results of the artificial urban network with IAS

Case	Traffic lights	Vehicles	Vehicle generation time	Total steps	Current plans	Future plans	CO_2 emission (Kg)
1	35	2,250	900	1,848	6	18	535.96
2	35	2,250	450	1,741	5	20	636.84
3	35	5,000	900	3,163	7	39	1,049.37
4	35	5,000	450	1,586	7	12	710.14
5	130	2,250	900	1,269	0	0	524.34
6	130	2,250	450	1,179	1	0	569.41
7	130	5,000	900	2,169	8	19	1,528.21
8	130	5,000	450	2,936	8	32	2,226.97

Table 3 – Comparison of vehicle waiting time

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
IAS (h)	153.03	212.33	349.83	241.20	63.98	87.77	373.21	806.92
Without IAS (h)	541.68	787.79	1,311.26	485.28	63.98	90.36	337.24	1,717.49

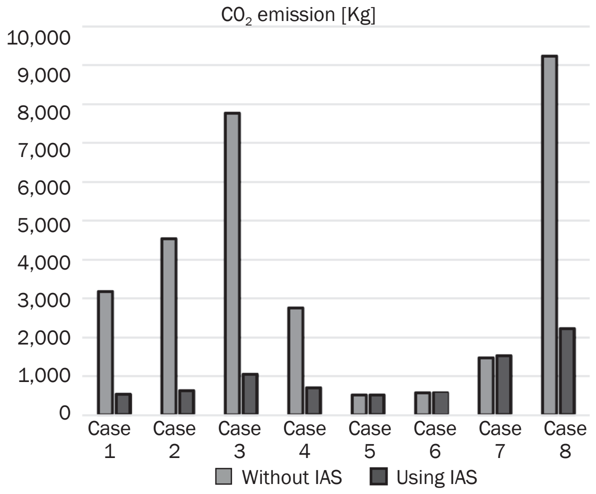
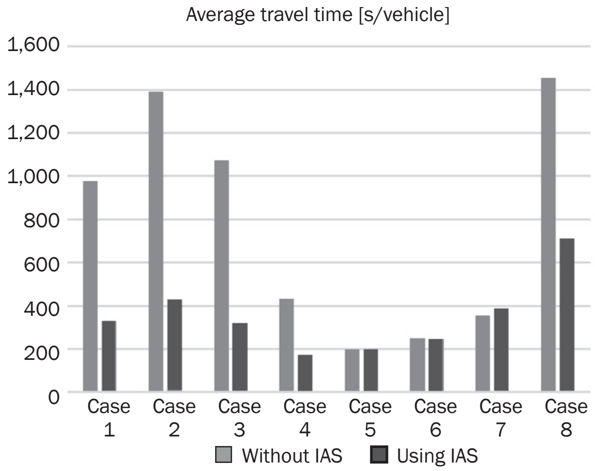


Figure 6 – CO_2 emissions in the artificial scenarios



.

Figure 7 – Average travel time in the artificial scenarios

When running the simulation in the *real scenario* without IAS, the urban net gets congested and the standard traffic lights control system is not able to solve the congestion. Thus, the vehicles cannot reach their destinations.

Using IAS, all vehicles reach their destination after 1,343 steps (while without IAS 247 vehicles did not finish their travel in 5,000 steps). During the simulation, the planner was called four times, and on one occasion there was no need to call it, given that the future plans worked properly. It resulted in 292.74 kg of CO2 emitted (versus 2,062.67 kg without IAS). The waiting time with IAS is 31.27 h, versus 309.20 h without it. Vehicle average travel time was 233.08 s, and 1,383.99 s without IAS. Thus, IAS also significantly outperforms the default system in the simulated real city scenario. In order to verify how robust the proposed approach is, its behaviour has been tested on other road networks with variable number of vehicles, and similar results were obtained

4. CONCLUSION

In this work, a road management system is presented with the following autonomic properties: self-monitoring (continuously monitors its state), self-management (automatically generates and executes plans), and self-optimizing (those plans achieve specified goals taking into account optionally some metrics). The developed system uses an open source traffic simulator (*SUMO*), a planner (LAMA), and a module with the following functionalities: monitor the current state of the traffic; recognize an incident; transform the simulator traffic state into a state acceptable by the planner; automatically generate goals for the planner; execute the planner; translate actions provided by the planner; and execute control actions in the simulator.

The contributions of this work can be summarized as follows:

- an architecture that monitors an environment for traffic incidents, generates plans for solving them

when detected, executes the actions in the plan, monitors the execution of plans, and replans if needed;
- a declarative model of traffic signals control actions;
- the integration with a traffic simulator, *SUMO*; and
- some experiments with map of real city, as well as some artificial scenarios.

The proposed approach was tested against artificial and real world data based simulation scenarios and showed a significant ability to solve traffic congestion.

One direction for future work could be to incorporate richer action models (including temporal or numeric aspects), and new types of goals. A second research direction would be to include other types of control actions, as diverting traffic by appropriate traffic signals and integrating them with traffic lights control, similarly to [10, 40, 41]. Also, it would be adequate to add some adaptation capabilities by using machine learning to automatically change the action models.

ACKNOWLEDGMENT

This work has been partially supported by the ARTS COST TU1102 action and the Spanish MICINN projects TIN2011-27652-C03-02 and TIN2014-55637-C2-1-R.

MATIJA GULIĆ, M.Sc.[1]
E-mail: matija.gulic@fer.hr
RICARDO OLIVARES, B.Sc.[2]
E-mail: ricardo.olivares@alumnos.uc3m.es
DANIEL BORRAJO, Ph.D., Catedrático de Universidad[2]
E-mail: dborrajo@ia.uc3m.es
[1] Sveučilište u Zagrebu,
Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Croatia
[2] Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain

KORIŠTENJE AUTOMATIZIRANOG PLANIRANJA ZA UPRAVLJANJE PROMETNOM SIGNALIZACIJOM

SAŽETAK

Konstantno povećanje broja vozila uz zadržavanje postojećeg kapaciteta prometnih mreža dovodi do stvaranja sve većih prometnih gužvi. Samim time potrebna su nova rješenja pomoću kojih bi uspješno izbjegli stvaranje prometnih gužvi. U ovom članku opisan je prvi pokušaj upravljanja prometom korištenjem autonomnog sustava baziranog na automatiziranom planiranju. Automatizirano planiranje može pomoći kod odabira redoslijeda akcija. Autonomni sustav mora pratiti trenutno stanje prometa, prepoznati kada trenutno stanje sustava prelazi u problematično stanje, postaviti niz željenih ciljeva koje bi postigli korištenjem planera, pokrenuti planer, i izvršiti izabrani redoslijed akcija. Stoga, predloženi sustav se u potpunosti može smatrati autonomnim sustavom. Dobiveni rezultati prilikom rješavanja prometnih gužvi su ohrabrujući, i predloženi sustav pokazuje znatan potencijal.

KLJUČNE RIJEČI

sustav upravljanja prometom; signalizacija i regulacija prometa; autonomni sustav; automatizirano planiranje;

REFERENCES

[1] Papageorgiou M, Ben-Akiva M, Bottom J, Bovy PHL, Hoogendoorn SP, Hounsell NB, Kotsialos A, McDonald M. ITS and Traffic Management. In: Barnhart C, Laporte G, editors. Handbooks in Operations Research and Management Science, 2007; p 743-754.

[2] Papageorgiou M, Diakaki C, Dinopoulou V, Kotsialos A, Wang Y. Review of road traffic control strategies. Proc. IEEE. 2003 Dec;91(12):2043-2067.

[3] Diakaki C, Papageorgiou M, McLean T. Application and evaluation of the integrated traffic-responsive urban corridor control strategy IN-TUC in Glasgow. Proceedings of the 78th Annual Meeting of the Transportation Research Board; Washington, DC, USA; 1999. Paper 990310.

[4] De Oliveira LB, Camponogara E. Multi-agent model predictive control of signaling split in urban traffic networks. Transp. Res. C, Emerging Technol. 2010 Feb;18(1):120-139.

[5] Bretherton R, Wood K, Bowen GT. SCOOT version 4. Proceedings of the 9th International Conference on Road Transport Information and Control; 1998 Apr 21-23; London, UK; 1998.

[6] Donati F, Mauro V, Roncoloni G, Vallauri M. A hierarchical decentralized traffic light control system: the first realisation "Progetto Torino". Proceedings of the 9th World Congress of the International Federation of Automotive Control; Budapest, Hungary; 1984. p. 2853-2858.

[7] Lee C, Machemehl RB. Genetic algorithm, local and iterative searches for combining traffic assignment and signal control. Proceedings of the 1st Conference on Traffic and Transportation Studies (ICTTS '98); 1998 July 27-29; Beijing, China; 1998. p. 25-27.

[8] Cascetta E, Gallo M, Montella B. Models and algorithms for the optimization of signal settings on urban networks with stochastic assignment. Ann. Oper. Res. 2006;144(1):301-328.

[9] Heydecker B. A decomposition approach for signal optimisation in road networks. Transp. Res. B. 1996;30(2):99-114.

[10] Chiou SW. Optimization of area traffic control for equilibrium network flows. Transp. Sci. 1999;33(3): 279-289.

[11] Wey WM. Model formulation and solution algorithm of traffic signal control in an urban network. Comput. Environ. Urban Syst. 2000;24(4):355-378.

[12] Ziyou G, Yifan S. A reserve capacity model of optimal signal control with user-equilibrium route choice. Transp. Res. Part B. 2002;36(4):313-323.

[13] Gartner N, Little J, Gabby H. Simultaneous optimization of offsets, splits and cycle time. Transp. Res. Rec. 1976;(596):6-15.

[14] Smith MJ. Properties of a traffic control policy which ensure the existence of a traffic equilibrium

consistent with the policy. Transp. Res. Part B. 1981;15(6): 453-462.

[15] Alvarez I, Poznyak AS, Malo Tamayo A. Urban traffic control problem: a game theory approach. Proceedings of the 17th World Congress IFAC; 2008 July 6-11; Seoul, Korea; 2008. p. 7154-7159.

[16] Trejo KK, Clempner JB, Poznyak AS. A Stackelberg security game with random strategies based on the extraproximal theoretic approach. Eng. Appl. Artif. Intell. 2015;37:145-153.

[17] Clempner JB, Poznyak AS. Convergence method, properties and computational complexity for Lyapunov games. Int. J. Appl. Math. Comput. Sci. 2011;21(2):349-361.

[18] Clempner JB, Poznyak AS. Modeling the multi-traffic signal-control synchronization: A Markov chains game theory approach. Eng. Appl. of Artif. Intell. 2015;43:147-156.

[19] Tettamanti T, Luspay T, Kulcsár B, Péni T, Varga I. Robust Control for Urban Road Traffic Networks. IEEE Transactions of Intelligent Transportation Systems. 2014;15(1):385-398.

[20] Ghallab M, Nau D, Traverso P. Automated Planning. Theory & Practice. San Francisco: Morgan Kaufmann; 2004.

[21] Fox M, Long D. PDDL2.1: An extension to PDDL for expressing temporal planning domains. Journal of AI Research. 2003;20:61-124.

[22] Garcia J, Florez JE, Torralba A, Borrajo D, Linares-Lopez C, Garcia-Olaya A, Saenz J. Combining linear programming and automated planning to solve multimodal transportation problems. European Journal of Operations Research. 2013;227(1):216-226.

[23] Castillo L, Armengol E, Onaindia E, Sebastiá L, González-Boticario J, Rodríguez A, Fernández S, Arias JD, Borrajo D. SAMAP. A user-oriented adaptive system for planning tourist visits. Expert Systems with Applications. 2008 Feb;34(2):1318-1332.

[24] Ai-Chang M, Bresina J, Charest L, Chase A, Hsu JC-J, Jonsson A, Kanefsky B, Morris P, Rajan K, Yglesias J, Chafin BG, Dias WC, Maldague PF. MAPGEN: Mixed-initiative planning and scheduling for the Mars Exploration Rover mission. IEEE Intelligent Systems. 2004 Feb;19(1):8-12.

[25] Simulation of Urban MObility (SUMO) [homepage on Internet]. [cited 2015 June 06]. Available from: http://sumo.dlr.de/wiki/Main_Page

[26] Krajzewicz D, Erdmann J, Behrisch M, Bieker L. Recent development and applications of SUMO - Simulation of Urban MObility. International Journal On Advances in Systems and Measurements. 2012 Dec;5(3&4): 128-138.

[27] Behrisch M, Bieker L, Erdmann J, Krajzewicz D. Sumo - simulation of urban mobility: An overview. SIMUL 2011, The Third International Conference on Advances in Systems Simulation; 2011 October 23-28; Barcelona; 2011. p. 63-68.

[28] Piórkowski M, Raya M, Lugo A, Papadimitratos P, Grossglauser M, Hubaux J-P. TraNS: Realistic joint traffic and network simulator for VANETs. ACM SIGMOBILE Mobile Computing and Communications Review. 2008 Jan;12(1):31-33.

[29] Schäfer R-P. IQ routes and HD traffic: technology insights about tomtom's time-dynamic navigation concept. Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (ESEC/FSE '09); 2009 Aug 24-28; Amsterdam, The Netherlands; New York, USA: ACM New York; 2009. p. 171-172.

[30] Krajzewicz D, Brockfeld E, Mikat J, Ringel J, Rössel C, Tuchscheerer W, Wagner P, Wösler R. Simulation of modern traffic lights control systems using the open source traffic simulation SUMO. Proceedings of the 3rd Industrial Simulation Conference; 2005 June 9-11; Berlin, Germany; 2005. p. 299-302.

[31] Nota R, Barelds R, van Maercke D. Harmonoise WP 3 Engineering method for road traffic and railway noise after validation and fine-tuning. Technical Report Deliverable 18; Berlin, Germany; 2005.

[32] OpenStreetMap [homepage on Internet]. [cited 2015 June 06] Available from: http://www.openstreetmap.org

[33] TraCI [homepage on Internet]. [cited 2015 June 06] Available from: http://sumo.dlr.de/wiki/TraCI

[34] Ghallab M, Howe A, Knoblock C, McDermott D, Ram A, Veloso M, Weld D, Wilkins D. PDDL – The planning domain definition language. Yale Center for Computational Vision and Control. Tech Report CVC TR-98-003/DCS TR-1165; 1998 Oct.

[35] Németh B, Csikós A, Varga I, Gáspár P. Multicriteria cruise control design considering geographic and traffic conditions. Acta Polytechnica Hungarica. 2013;10(6):119-134.

[36] ICAPS [homepage on Internet]. [cited 2015 June 06] Available from: http://ipc.icaps-conference.org

[37] Richter S, Westphal M. The LAMA planner: Guiding cost-based anytime planning with landmarks. Journal of AI Research. 2010;39:127-177.

[38] Jiménez S, Fernández F, Borrajo D. Integrating planning, execution and learning to improve plan execution. Computational Intelligence Journal. 2013;29(1):1-36.

[39] Tools/Import/OSM [homepage on Internet]. [cited 2015 June 06] Available from: http://sumo.dlr.de/wiki/Tools/Import/OSM#server.py

[40] Allsop R. Some possibilities for using traffic control to influence trip distribution and route choice. Proceedings of the 6th International Symposium on Transportation and Traffic Theory; 1974 Aug 26-28; Sydney, Australia; New York, USA: Elsevier Publishing Company; 1974. p. 345-374.

[41] Yang H, Yagar S. Traffic assignment and signal control in saturated road networks. Transportation Research Part A: Policy and Practice. 1995;29(2):125-139.