

Traffic Lights Strategy Adaptation

Michal Janošek*, Radim Farana

Centre of Excellence IT4Innovations, Division of the University of Ostrava,
Institute for Research and Applications of Fuzzy Modeling, Ostrava, CZ

*Corresponding author: michal.janosek@osu.cz

Received October 25, 2013; Revised November 14, 2013; Accepted November 26, 2013

Abstract This paper deals with a problem of a traffic lights strategy adaptation based on a traffic flow. The aim of this paper is to show an advantage of an adaptive traffic lights strategy. For the adaptive traffic lights strategy switching the author's methodology for adaptation of parameters of complex system for complex system behaviour adaptation is used [3,4,6]. Traffic simulation is done using simulation software NetLogo. As an example, a model of a traffic grid with traffic lights on its crossings is used.

Keywords: traffic lights, adaptation, strategy, methodology, complex system, pattern recognition

Cite This Article: Michal Janošek, and Radim Farana, "Traffic Lights Strategy Adaptation." *American Journal of Mechanical Engineering* 1, no. 7 (2013): 226-230. doi: 10.12691/ajme-1-7-15.

1. Introduction

Nowadays we are experiencing constantly increasing traffic density. Moreover, traffic density is not constant but it changes related to the daytime. It will be different during morning peaks, weekends, in a time of cultural events, closures, or after a football match. This leads to frequent traffic jams, especially in places where roads intersect [1]. This problem is very important especially in city mass transport density [2].

There are generally two ways how to deal with traffic jams and roads capacity. One of them is to implement minor modifications and change the traffic priority. This can be done using traffic signs based on the traffic observation and calculation just by placing these sign at correct places. Other way is to use traffic lights on the roads intersects. Usually, there is only one strategy implemented on each traffic intersection used and in the case of low traffic during nights traffic lights are turned off. Rarely it is taken into an account current situation on the road. So there is no traffic lights adaptation based on the traffic flow. In this article there is presented an adaptation strategy which switches the strategy of traffic lights based on current traffic situation.

The original idea of a more advanced algorithm for mediating the traffic by traffic lights strategy adaptation comes from Carlos Gershenson and his work on Self Organizing Traffic Lights [7]. He has presented three self-organizing strategies for traffic light control which outperform traditional methods due to the fact that they are "aware" of changes in their environment, and therefore are able to adapt to new situations. The strategies are very simple: they give preference to cars that have been waiting longer, and to larger groups of cars. Still, they achieve self-organization by the probabilistic formation of car platoons. In turn, platoons affect the behaviour of traffic lights, prompting them to turn green even before they have

reached an intersection. Traffic lights coordinate stigmatically via platoons, and they minimize waiting times and maximize average speeds of cars. Under simplified circumstances, two methods can achieve robust full synchronization, in which cars do not stop at all. Indeed, his methodology still does not solve major changes in the traffic system environment. Each of his strategies has its advantages in different traffic situations.

This article focuses on the traffic lights system adaptation where is an effort to find such means of mediating the system's behaviour that would make it possible to adapt to the current state of the system and the environment and react to the changes (by switching the strategies) so that the desired behaviour of the system is kept in specified limits or patterns of behaviour. Various behaviour patterns are searched in the system behaviour to recognise a change in the traffic situation. Based on the recognised patterns, a corresponding response is chosen.

2. Methodology

In this article there is an effort to find such means of mediating the system's behaviour that would make it possible to adapt to the current state of the system and the environment and react to the changes so that the desired behaviour of the system is kept in specified limits or patterns of behaviour. The instruments of regulating the system's behaviour are its parameters [5]. The adaptation of the system's behaviour itself consists of recognizing these characteristic patterns using neural networks and the subsequent mediation of the system's behaviour through selected parameters and their action ranges based on pre-prepared expectations of what will happen if the system's behaviour exhibits a known characteristic pattern.

Since the NetLogo software uses the bottom up approach to simulate different scenarios our methodology follows this approach as well. Each individual vehicle so as traffic lights is seen as an individual agent fulfilling its goals.

The methodology consists of several main phases (Figure 1). Agents and Parameters Identification, Parameters, Hierarchy, Simulation, Behaviour Pattern Recognition, Adaptation of Parameters. It is possible to return back to any previous phase if next phases are altered accordingly. In this article, only last three phases are used.

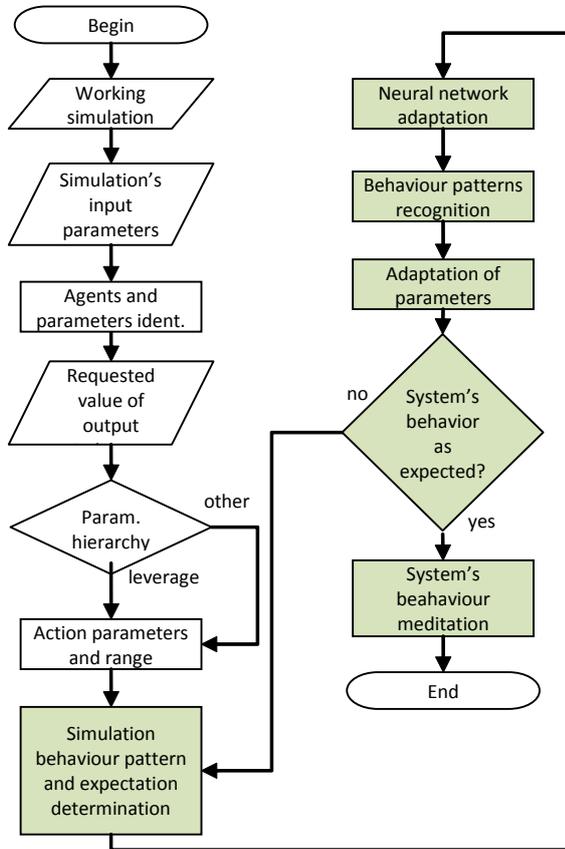


Figure 1. Methodology flowchart

Simulation

In this phase we have to conduct a necessary amount of experiments when a behaviour of the system is observed. Characteristic patterns of the system behaviour are searched in its output indicators. These patterns are saved and described. Particular patterns are determined. These patterns are split into several categories, each of which has similar properties. A training set is created by this procedure. For every category, it is necessary to define what will happen if any pattern from any category emerges in the system's behaviour.

Behaviour Pattern Recognition

In this phase the neural network is adapted by the prepared training set. This task makes use of our own developed tool called EDU Sandbox (<http://sourceforge.net/projects/esbox/>). Then the neural network is able to generalise the learned training set. After that, when the simulation is running, the neural network can recognize similar behaviour pattern it already knows and returns that information back to the simulation software.

Adaptation of Parameters

Based on the recognised patterns in the system's behaviour it is possible to adapt action parameters' values according to the prepared expectations. Adaptation logic has to be programmed in the simulation software.

The flowchart of the methodology (Figure 1) represents a more detailed step-by-step procedure with each above-mentioned phase.

3. Experiment

For our experiment we use a simulation prepared by Carlos Gershenson [7] which is available at <http://turing.iimas.unam.mx>. We have modified this simulation to able to connect it to our pattern recognition utility. Adaptation logic was added as well. The model simulates city traffic where traffic lights try to "self-organise" to improve the traffic flow in the city. It consists of an abstract traffic grid with intersections between cyclic single-lane arteries of two types: vertical or horizontal. Cars only flow in a straight line, either eastbound or southbound. Each crossroad has traffic lights that allow traffic flow in only one of the intersecting arteries with a green light.

The experiment used the following tools: NetLogo simulation software [8], MySQL database, EDU Sandbox. Communication between these tools is following (Figure 2).



Figure 2. Communication between NetLogo and EDU Sandbox

The simulation runs in the NetLogo simulation software. The simulation consists of discrete steps. Information about each simulation step is stored into the MySQL database. This information is read by the EDU Sandbox utility. This utility is able to learn patterns beforehand. EDU Sandbox reads in real-time data from the database and reports back to NetLogo if any known patterns have been recognized. Based on the recognised pattern, the adaptation logic in NetLogo adapts the chosen action parameters to mediate simulation's behaviour.

As we can see (Figure 3) Gershenson developed different strategies of self-organising traffic lights. Each strategy has its advantage and disadvantage. The most significant difference is with about 1000 vehicles. The *soil-platoon* strategy begins to be ineffective, while the *soil-phase* strategy tends to be better for more than about 1000 vehicles. Gershenson's work does not implement any adaptation strategy where it would be possible to switch between strategies based on the current traffic status. Our experiment tries to do that.

We use one output indicator called *Average Speed of Cars* which tells us what the average speed of all vehicles in the simulation is. The goal will be to allow the vehicles to pass through this grid system as fast as possible regardless the traffic density.

Desired value of output indicator

For our purpose, there is one key indicator which is *Average Speed of Cars*. We require that its value would be as high as possible.

Action parameters and their range

For the choice of action parameters (parameters which can effectively change the system's behaviour; traffic lights strategy parameters, etc.) it is crucial to define their

range where behaviour of the output indicator is desirable (Table 1). It includes the minimum and maximum (or

enumeration) values of the output indicator given its stability and possible oscillations as well.

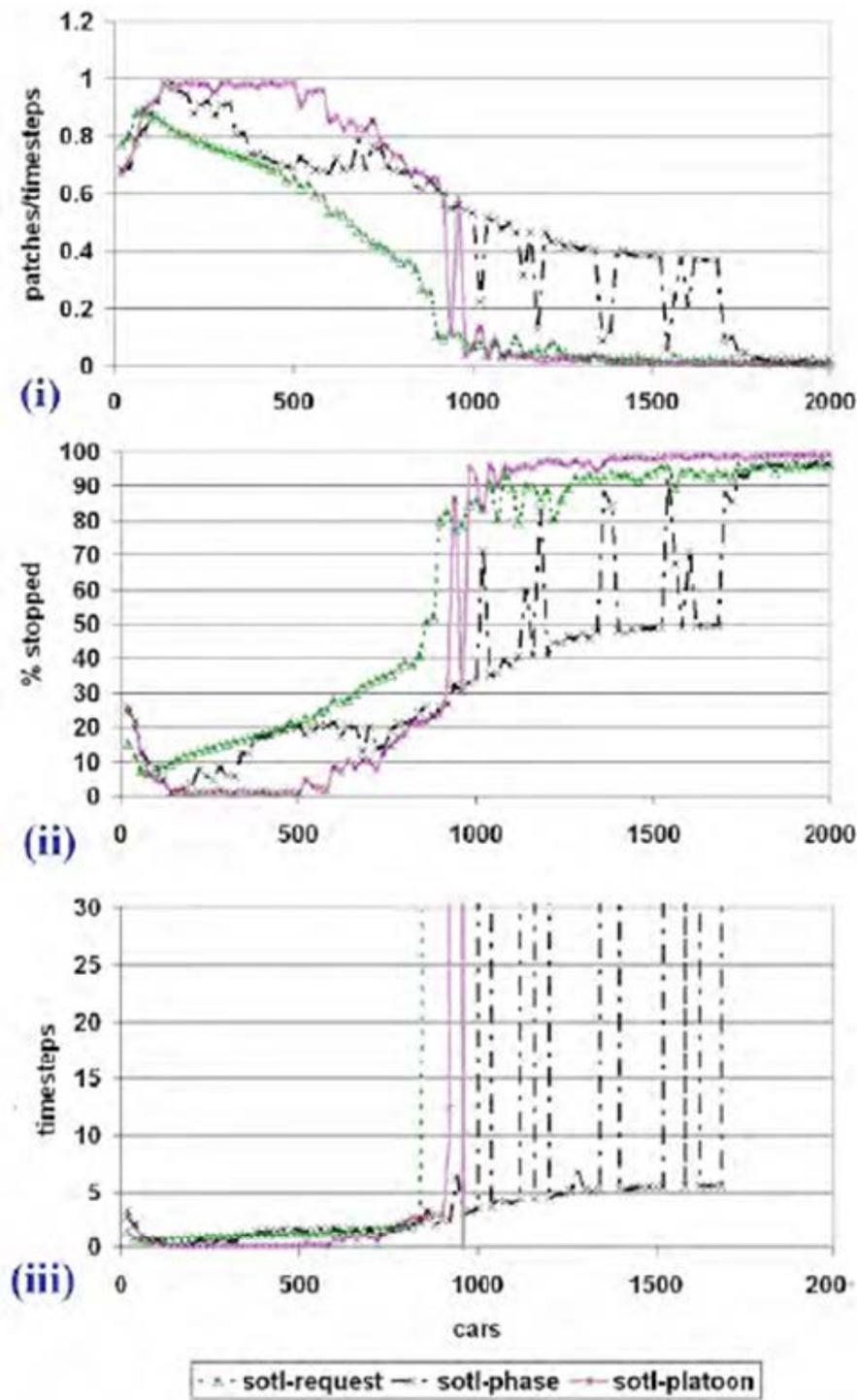


Figure 3. Traffic lights SOTL strategies comparison [7]

Simulation

In this phase we have to conduct a necessary amount of experiments with action parameters' candidates and their range to observe characteristic system's behaviour observable on the output indicator.

Pattern and expectation determination

In this step, we should have characteristic behaviour patterns split into several categories (Figure 4). Each category represents one type of behaviour and an anticipation what would happen after that behaviour emerges. In (Table 1) we have two categories with corresponding anticipations.

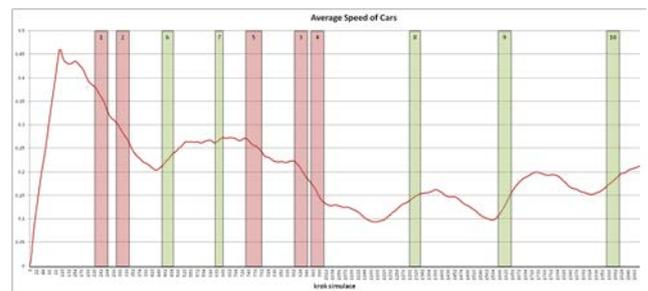


Figure 4. Patterns determination

Table 1. Font Sizes for Papers

Pattern	Anticipation	Action parameter
category 1	ongoing downtrend	control: 1) sotl-platoon to sotl-phase 2) sotl-phase to sotl-platoon
category 2	ongoing uptrend	control: current value – no change

Neural network adaptation

In this step, all patterns are presented to the neural network to learn them. Here are some examples of training sets (Figure 5).

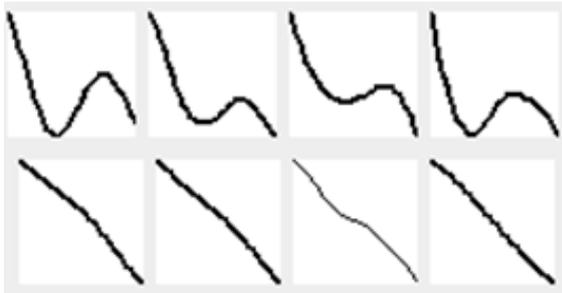


Figure 5. Category 1 patterns example

Behaviour Pattern Recognition

When the neural network has been adapted to recognize all training patterns, we can launch the simulation again and see how patterns are recognised. In our case, the recognized patterns appear in our EDU Sandbox utility and are transferred via MySQL to NetLogo according to adaptation logic.

Adaptation of Parameters

In case we had chosen correct patterns and their adaptation had been held successfully, we can adapt the chosen input action parameters of the simulation using application logic implemented in NetLogo. The system should behave according to our expectation. In our case, it should switch between two strategies in order to allow the vehicles to pas as fast as possible (Table 1).

Figure 6 shows the experiment’s result. First, original experiments for *sotl-phase* and *sotl-platoon* control mechanism were repeated. Two other lines, *pattern3vzory4_02* and *grid_2def_3pattern* represent our adaptation strategies, each with slightly different patterns. These two strategies adaptively combine both original strategies *sotl-phase* and *sotl-platoon* based on the traffic density.

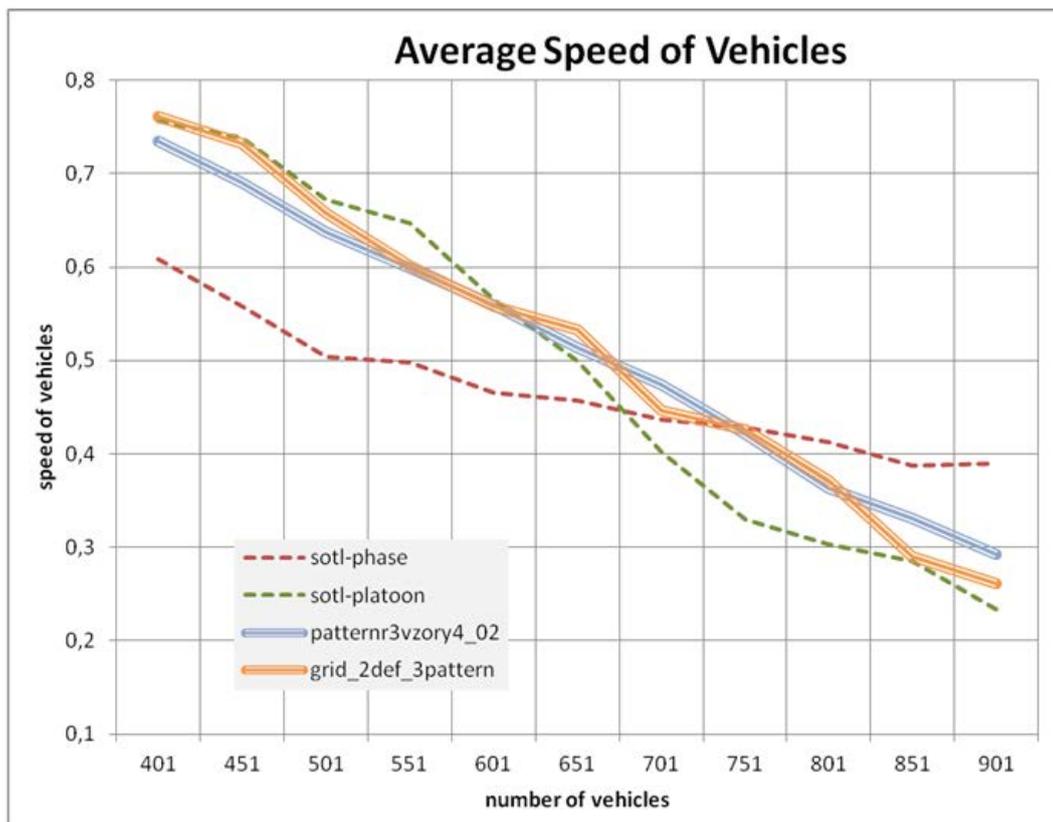


Figure 6. Results of adapting system’s behaviour

4. Conclusions

In this article, I have presented a methodology for system’s behaviour adaptation which was used for the traffic lights strategy adaptation based on the current traffic flow. In the Experiment chapter, I have verified that the methodology works as expected and is able to adapt system’s behaviour in the desired way. This experiment was only one of possible scenarios. If we compare the results without and with our methodology, our results are

better than the original experiment [7] because we were able to mediate the system’s behaviour by adapting the input parameters of the simulation and thus improve average speed of vehicles.

Acknowledgement

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and during

the completion of a Student Grant with student participation, supported by the Czech Ministry of Education, Youth and Sports.

References

- [1] Křivda, V. Simulation of traffic on intersection after change of traffic priority. *Transactions of the VŠB - Technical University of Ostrava, Mechanical Series*. 2011, LVII. Nr. 1, pp. 195-200.
- [2] Teichmann, D., Dorda, M., Fric, J. *An Extended Model for the Design of City Mass Transport Network*. Proceedings of the 29th International Conference on Mathematical Methods in Economics 2011. Praha: Professional Publishing, 2011, s. 704-708.
- [3] Janošek, M., Kocian, V., Volná, E. *Complex System Simulation Parameters Settings Methodology*. Nostradamus. International Publishing Switzerland 2013: Springer, 2013. s. 413-422.
- [4] Janošek, M., Kocian, V. *Simulation parameters settings methodology proposal based on leverage points*. Chaos and Complex Systems. Springer, 2013. s. 411-414.
- [5] Janošek, M., Kocian, V., Volná, E., Kotyrba, M., Habiballa, H. *Predator-Prey Simulation's Parameters and Leverage Points*. Proceedings of 27th European Conference on Modelling and Simulation ECMS 2013. Sbr.-Dudweiler, Germany: European Council for Modelling and Simulation, 2013. s. 367-371.
- [6] Volná, E., Janošek, M., Kocian, V., Kotyrba, M., Oplatková, Z. *Methodology for System Adaptation Based on Characteristic Patterns*. Robotic Systems - Applications, Control and Programming. Croatia: InTech, 2012. s. 611-628.
- [7] Gershenson, C.: *Design and Control of Self-organizing Systems*. Mexico: CopIt ArXives, 2007.
- [8] Wilensky, U. (1999). *NetLogo*. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.