SCENARIO-BASED APPROACH TO AGENT'S EVALUATION

Richard Lipka[1], Pavel Herout[1]

[1]University of West Bohemia, Faculty of Applied Sciences,
Plzeň, Univerzitní 8, Czech Republic

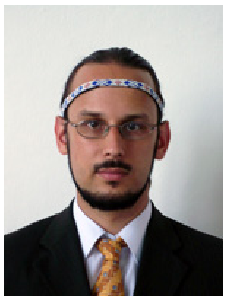
lipka@kiv.zcu.cz (Richard Lipka)

Abstract

Software agents allow solving problems, which cannot be easily described in the pure analytical way. Control of the traffic light signalization in city is an example of such problem, and many different agent-based solutions were proposed in the last decade. However, it is difficult to compare proposed solutions and determine which agent will provide best results. We have created a system that allows comparing different traffic control agents. We are using microsimulation of the real traffic network and the agents are operating in the same conditions, described in the detailed scenarios. Scenarios allows us to model non-linear character of the traffic flow and embrace different situations, which may occurs in the real traffic – such as accidents or sudden changes of weather conditions. Comparison of agents in the different scenarios may show their strong and weak parts, and help with selection of agent which provides the best results.

Keywords: Traffic control, Software agent, Evaluation, Microsimulation.

Presenting Author's biography



Richard Lipka was born in Plzeň, Czech Republic and went to University of West Bohemia, where he obtained master degree in 2007. Since then, he is a PhD student at Department of Computer Sciences and Engineering of UWB. His main scientific interests are road traffic simulation and agent control of large networks. His e-mail address is: lipka@kiv.zcu.cz .

1 Introduction

As intensity of traffic in cities grows, it seems more important to find a solution, which allows a smooth traffic and minimizes the traffic congestions. Within cities, it is very expensive or even impossible to change existing traffic infrastructure, so better traffic control is important way to improve the traffic situation. Since the first utilization and controlling of the traffic lights, several approaches to their settings are explored ([1]). One of the recent explored options is using of the software agents.

We have studied thoroughly a lot of different publications describing traffic control based on using software agents ([2], [3], [4], [5] or [6]), we have not found a tool for their comparison. In papers, usually some basic experiments and results are shown, but they don't provide sufficient information to decide, which of described agents is better. It also cannot provide any information about agent's behavior in different situations. Moreover, tests are performed by using completely different tools, in macro and microsimulators, which makes comparison of results in papers difficult, if not impossible. Because of this, we have decided to create a system that allows evaluation of different traffic control agents. This system will allow comparing agents in the same environment, with the same traffic condition. In order to test agents thoroughly, we have decided to equip our tool with detailed adjustable scenarios.

1.1 Agents in traffic control

One of most general definitions of software agent is given in ([7]). It claims that any software capable of perceiving its environment and acting upon it may be considered as a software agent. In some publications, intelligent agents are discussed. One of the widely accepted definitions of intelligent agent is given in a [8]. According to it, software agent is an encapsulated computer system, deployed in environment, which is capable of flexible and autonomous activities, in order to achieve a specific goal. This definition implies three basic conditions imposed on agent. They are autonomy, sociability and an initiative. Agent is autonomous, if it is able to act without being controlled by any other system or human. The sociability is the ability of agent to cooperate with other agents. And finally the initiative is the ability of agent to act in order to achieve a long-term goal. An agent fulfilling these three conditions may be called "intelligent".

Any dynamic traffic control fulfils at least two of these conditions – it has to have sensors to observe traffic situation at the crossroad and adjacent roads and it can interact with traffic flow by changing light signals on semaphores. These are also conditions required by ([7]). Additionally, many proposals for dynamic control (such as [3] or [4]) are using also a cooperation of agents on different crossroads, thus fulfill also third part of definition. For us, it seems natural to use paradigm of software agents to describe a dynamic traffic control, even if it is not explicitly mentioned in its description.

In proposed control systems, agents are placed on several important positions in traffic network. Basically, agents are used to control crossroad, usually one agent is controlling one crossroad. Some proposals are also containing a supervising agent, superior to several crossroad agents and allowing some type of centralized cooperation ([4]). It is also possible, that agents cooperate as peer-to-peer ([3]). Other agents may be used to control variable traffic signs ([6]).

We have sorted traffic control systems based on software agents into three groups – centralized, social and non-interacting agents. This grouping is based on way of agent collaboration; it doesn't specify anything about their internal mechanism. We have decided to focus on communication. This approach allows us to impose requirements on used simulation and also requirements on communication hardware if agents are planned to be used in real world. The simplest are non-interacting agents. They control only one crossroad, without any direct cooperation. They have no information about traffic situation on other crossroad. Thus, they are able only to provide a local optimization ([2], [5]). Social agents are able to communicate, either with all agents in the network or with agents on adjacent crossroads ([3]). The last group contains centralized agents. Agent systems listed in this group are using hierarchy of agents, where agents on crossroads are controlled by one supervising agent ([4]).

1.2 Overview of tested agents

In this section, two existing agents are described, as an example of possible variations. We have selected these agents, because they are representing two distinct families of software agents – non-interacting agents and social agents. Further, mechanism of vehicle actuated signal control is described, as it is basic way of dynamic traffic control ([9]). Result of comparison of these three agents is the last part of this paper.

1.2.1 KOSU

Paper [2] shows three simple control agents. All types are observing only one crossroad and there is no communication amongst them. They observe number of passing vehicles and calculate difference in amount of cars in last two periods. They are also able to determine length of queues in upstream lanes. First type is called even agent. It has prepared set of static time plans and it can only choose amongst them, according to actual state of traffic. Second agent is using method similar to vehicle actuated signal control. If there are vehicles in the lane, agent is trying to keep green signal as long as possible. If there are no vehicles in the lane, agent is lowering duration of green signal. With change of green signal duration, length of cycle is also changed, so the change doesn't affect other phases. The third type is very similar, but the length of cycle is constant, so rising of duration of green signal in one phase leads to lowering duration of green in other phases. At the end of article, it is shown, that the third agent has best result in terms of average waiting time of cars. KOSU is an example of very simple, non-interacting reactive agent.

1.2.2 FERR

In [3], agent based on modifying time plan according to current situation on crossroad is described. There is only one type of agent, designed to control one crossroad. It is able to observe number of vehicles in upstream of each lane. This information is used to create "opinion", coefficient shared with neighbor agents and together with opinion of neighbor agent to choose control strategy. Agent has prepared signal phases and is able to choose their order and duration. Order of phases is chosen according to optimization function. At the end of each phase, new phase is selected by using "score" gained from performance evaluation function. Shared opinion and measurement from last cycle phase are used as inputs. Agents expect that the state of traffic in next phase will be the same as in previous one. They calculate score for all possible phases, to determine which would have been the best in the last cycle. The phase with biggest score will be used for the next cycle – if the traffic in two following phases is similar, then the phase with the biggest score will be the best one. Because character of traffic is changing in time, one coefficient of opinion is changed by learning mechanism. It can be replaced by using database with the pre-calculated coefficients for the different time periods.

FERR is a good example of simple, social learning agent (but with using database of pre-calculated coefficients, learning mechanism can be evaded and agent changed into reactive one). Agents are not using any complex representation of their environment, they only changes coefficient in performance evaluation function.

1.2.3 VASC

Vehicle actuated signal control is not usually denoted as agent, but we need describe it, because it was used as basis for comparison of dynamic methods of traffic control ([9]). It is the simplest form of dynamic traffic lights control. In VASC, a set of signal phases is given, with fixed order. Instead of exact timing of phases, there are minimal and maximal durations for each phase. When signal phase is activated, it continues at least till end of minimal duration. Then, traffic controller starts to observe lanes with a green signal. If there are still some vehicles, duration of phase is prolonged for some time. This can continue until maximal duration of phase is reached, or no vehicles are detected in lanes with green signal. Lights are consequently switched to next phase.

2 Simulation

We are using our own simulation software JUTS ([10]). It is microsimulator, based on using cellular automaton for modeling roads. It allows creating arbitrary maps, corresponding to real situation of the city. Simulation is time-stepped, one step in the model is equivalent of one second of real time. Simulated environment is composed of one directional traffic lanes (1 dimensional array of cells), which may be composed into roads. Vehicles may pass from one lane to another, but only in the same direction. Lanes are connected by crossroads, also composed from cells (so movement within the crossroad is also simulated). Vehicles generators represent the surroundings of simulated network by injecting new vehicles into simulation. Terminators are used to dispose vehicles from the simulation, when they are leaving simulated area, and also to collect characteristic taken by vehicles (such as time spend in simulation, time spend by waiting in queues etc.).

Vehicles in JUTS are implemented without any ability to search path, their way of movement in traffic network is described below (section 3.1). They always follows traffic rules, there are no rules violations implemented.

In each step, state of all simulated units is updated. Because vehicles and vehicle generators are only active elements which requires computation time, speed of simulation is dependent on amount of simulated vehicles, size of simulated map has no important influence. We have tested, that our simulation can handle about 5000 vehicles, when it is running in real time (to simulate 1 second takes approximately 1 second in reality), with activated visualization module, which needs approximately 40% of computation power. These tests were performed on common PC (specification – CPU: Intel Core2 2.66 GHz, 2GB RAM).

3 Scenarios

Core of our approach to agent's evaluation are scenarios. Traffic flow can be highly unstable and difficult to predict ([11]). We cannot expect, that it remain unchanged during long period of time, or that its changes will be slow and regular. Dynamic traffic control, such as using of software control agents, may allow controlling traffic effectively, by immediate reaction on changes in traffic flow. Thus, it is important to test proposed control agents in changing environment, comparable with real situation. Simulation scenarios are designed to allow this type of testing.

Each scenario describes behavior of whole traffic network during time. When the scenario is loaded into simulator, it provides initial settings of simulated network and also provides changes during run of simulation. Each scenario is described by set of xml files.

3.1 Vehicle settings

The most important part of scenario is settings of vehicles/drivers behavior. Two basic ways how to simulate vehicles behavior exists. If you don't want to use artificial intelligence to drive vehicles (which requires higher computing power and significantly reduces the amount of vehicles which could be simulated).

3.1.1 Prepared paths

The first way is to use previously prepared paths. Each vehicle has assigned path, possibly with some kind of time schedule for stops, and follows it. If this approach should be used, it is necessary to prepare large amount of paths before simulation is started. This allows modeling vehicles which really mimics behavior of real drivers, but there is a problem with collecting data for path generating.

The easiest thing is to prepare path for vehicles of public transportation. Their routes and time schedules are usually publicly known and can be easily found. Obtain data for a privately owned vehicle is more difficult. The most convincing way is to use sociological survey. Information obtained from real people allows reconstructing their habits and used paths ([12]). Modern technologies offers more accurate ways of data collecting, such as tracking of cellular phones moving in the cars . But for now, these are just theoretical proposals, and there is huge concern about privacy protection ([13]).

Another approach is to use some form of artificial intelligence to generate paths of each vehicle ([14]). This is similar like using of artificial intelligence to drive vehicles in simulation itself, but it requires less computing power. Paths are prepared by some kind of optimization process before simulation itself. During simulation, each vehicle obtains prepared paths and follows it "blindly", without any changes – thus being much easier to simulate, than if it is driven by artificial intelligence. Results of simulation (especially information about congestions) may be used in next iteration of path generating (reference). This way of path generating is easier, than using of sociological survey, but less conclusive. Path generator may create paths optimal by some criterion, but it is difficult to prove, that such paths corresponds to paths used by real drivers.

3.1.2 Randomly generated paths (Monte Carlo method)

Another approach may be used, if information about turning on crossroads is available. Instead of preparing paths trough whole network, vehicles are deciding randomly at each crossroad. When a vehicle arrives to a crossroad, where it may choose from several future directions, the choice is made by using a random numbers generator. To set the generator, it is necessary to know probability of each direction applicable at the crossroad.

Obtaining data for this method is much easier; it requires only sensors on each used crossroad. Many cities is now using some kind of dynamic traffic control, so such sensor are widely spread. Even if they are only detecting passing vehicles, information from them may be used to obtain information about vehicles turning.

We use combination of both ways described above. In order to simulate vehicles, scenario description contains settings of vehicles generators and generators on the crossroads. Common traffic is simulated by Monte Carlo method; generators are set to inject vehicles, a time between two vehicles is generated randomly, with using of Poisson distribution. Parameters of generators may change in time during scenario, thus emulating irregularities in real traffic flow. In similar way, random generators on crossroads are created, according to observed probabilities of turning. These probabilities may also be changed during one scenario. Changes on crossroads may be used to simulate change in drivers behavior in time (such as preference of different targets in different period of time), but also to simulate accidents, detours and similar phenomena. As was mentioned previously, vehicles do not have any form of intelligence or ability to choose their paths, so simulation doesn't allows inserting obstacles into traffic network. Instead of this, some direction may be closed by using zero probability in appropriate generators, so vehicles then won't try to use this direction.

Vehicle generators also get information about amount of public transportation vehicles, their characteristics and their time schedules. Vehicles with prepared paths are injected according to their time schedules and follows prepared paths. If scenario contains accident or some other change in traffic network, path reflecting the new situation has also to be prepared. Such path is automatically submitted to the path-following vehicles in front of the accident, to simulate their reaction.

An optional part of settings are seeds of random numbers generators, which allows running exactly the same simulation several times.

3.2 Network settings

Network settings contain all settings which are influencing traffic infrastructure and not vehicles directly. At first, it is description of network topology, positions of traffic lights, generators ant terminators. Network topology consists from full description of roads and crossroads, and it cannot be changed during run of scenario. If reaction of agent to change of topology is tested, it has to be handled through settings of vehicles, as described higher. Settings of maximum allowed speed in each lane may be also part of network topology. This is only exception; it may vary in time, so it may be use for example to model worse weather conditions, when drivers are required to ride slowly.

3.3 Control settings

Important part of scenario is setting of selected method of a traffic control. Method of control and its parameters is specified in this part. The control is set separately for every crossroad equipped with light signalization. Because each agent requires set of its specific parameters, only a name of the selected method and a reference to file with agent's settings is given here.

Some of agents are able of learning and adjusting their parameters, but all of them require initial settings and usually several parameters, which are not changed during the simulation (for example minimal and maximal length of signal cycle, priority of main roads, type of phases).

3.4 Measuring

The last part of scenario defines points of measurement and measured parameters. JUTS allows to measure and store large variety of parameters at each point of simulation. Values may be measured in three distinct ways – at a cell (as we mentioned above, simulator is based on cellular automaton, so one cell is the smallest measurable object in the simulation), at a segment or from vehicles. Each value is measured by software "probe", with given position, type of measuring (point, segment and vehicle) and measured parameter.

The point measuring simulates work of devices such as induction loops, IR gates etc., which are able to observe only their immediate surroundings. This method is mainly used by control agents themselves, because it meets best the work of real measuring devices installed on crossroads.

The second method use information from whole segments. It allows to measure average speed at some part of simulation, with a length of queues or a density of traffic. This type of information seems to be useful for traffic control, but in a real world, it is difficult to obtain it. It is used mainly to evaluate results of simulation.

The last method is based on observation of vehicles. Now we use it only to get information from vehicles, which leave the simulated area. It is possible to measure values like time spend by vehicle in the simulation model, time spend by waiting, average speed during whole path through simulated network or the path which vehicle has followed. This may provide more general view on traffic control and gives information about experience of drivers in controlled network.

For whole measurement, sampling frequency may be set. Implicitly, we expect to take a sample in every step, e.g. each second. It is possible to take sample asynchronously, only after specified event. We use the sampling of a queue length after a change from the red signal to the green one.

It is advisable to choose only those probes, which will really be used during evaluation process. Each probe becomes active part of simulation, so using of too many probes slowing the run of simulation. Also, with one second sampling frequency, and many probes in simulation, considerable amount of data is generated by each simulation's run.

4 Evaluation

In order to compare two different agents, it is necessary to see their behavior in the same environment. Thus, the first step is preparation of map and scenario. We use real maps, based on the traffic network in Plzeň.

Before running the simulation itself, it is important to calibrate compared agents, so they may show their best performance. It means to set their invariable parameters especially. Unfortunately, original publications about agents often do not contain exact procedure of parameters adjustment, only some basic hints are included. Because of this, we run several simulations only to determine value of these parameters, before comparison itself. We cannot claim that our setting is optimal, but at least we choose from several options the one with the best performance.

When scenario and agents are ready, several (typically 10) simulation's run are performed for each agent. In each run, different seeds for random number generators are used. The same data are collected, according to setting of probes in scenario.

To evaluate quality of traffic control, we use two types of criteria – capacity criteria and queuing criteria. According to [1], both are the basic criteria to evaluate control of the crossroad. Capacity criteria are based on observing amount of vehicles, which were able to pass the crossroad during observed time. This may be compared with theoretically achievable maximum, computed as is shown in [1]. The queuing criteria are based on observation of queues in simulated network. Mainly average lengths of queues at the end of red signal (or sampled regularly) or overall time spent by vehicle in queues are measured. Further, information about traffic characteristics, such as an average speed in observed lanes might be used to evaluate an agent's performance.

Because of a large amount of observed values (many of them may be collected in each traffic lane or crossroad), we are also using artificial criteria, calculated from measured values.

5 Example of experiment

To show a usage of our system, we have prepared comparison of agents KOSU ([2]) and FERR ([3]) and VASC ([9]) method. Short description of selected agents is above. Because KOSU contains three possible modes, we use the third one (fixed length of signal cycle, length of green signal is set dynamically for each cycle at the end of the previous one).

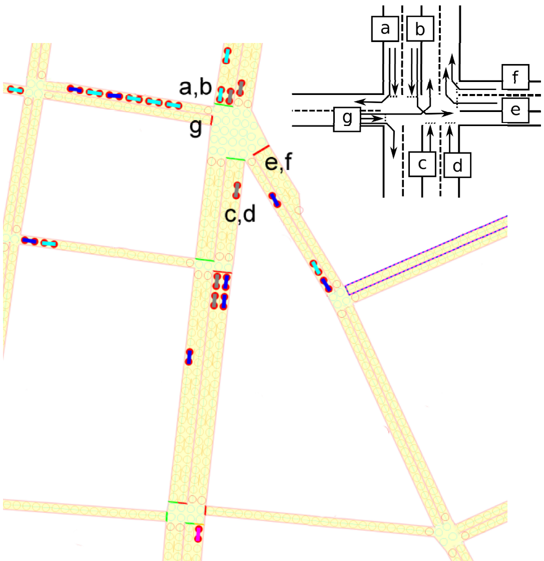


Fig. 1 Part of simulated map

We have prepared two scenarios, based on the real traffic network, around one of main roads in Plzeň (at Fig. 1 you may see screenshot from simulator and scheme of main crossroad). Both scenarios are 30 minutes of simulated time long. In the first scenario, a traffic density is varying from very dense (close to theoretical capacity of the road) to calm traffic. Four peaks of dense traffic are simulated. During third and fourth peak is part of traffic diverted from main road, as local drivers try to avoid the worst congestion by using minor roads. Second scenario doesn't contain significant changes in traffic flow, but after five minutes of simulation (enough time for vehicles to fill the network), speed on roads is reduced for next 15 minutes to 30 km/h to simulate bad weather conditions. During this period, vehicles also use a smaller acceleration. Agents were used on all controlled crossroads; all of them were initially set to give a priority to vehicles on main road.

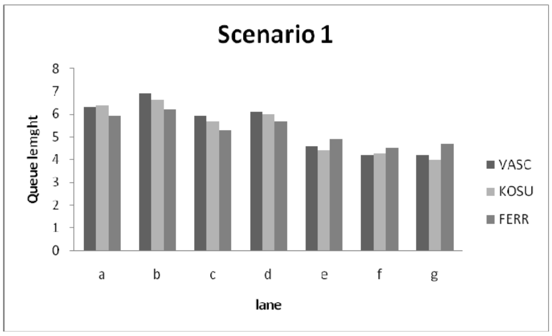


Fig. 2 Scenario 1 results

Results of both simulations are shown at Fig. 2 and Fig. 3. We have measured the length of queues at the most important crossroad, scheme of the crossroad is in Fig. 1. In both scenarios, we can see, that FERR achieved best results in lanes "a", "b", "c" and "d"(main road), but at the cost of longer queues on minor lanes "e", "f" and "g". This is caused by prioritization of lanes with higher traffic density.

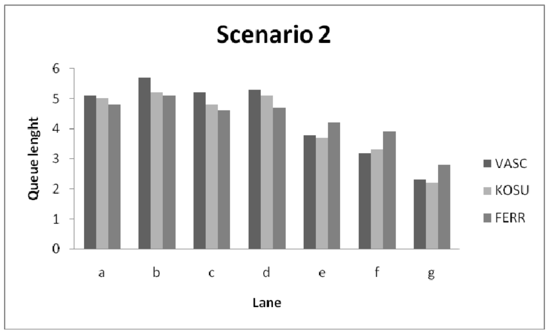


Fig. 3 Scenario 2 results

6 Future work

So far, we have created the simulation tool, equipped with support for implementation and comparison of different traffic control agents. It is possible to run prepared scenarios automatically and to aggregate results from several runs. Now the most difficult part is calibration of agents, preparation of their static parameters and limits of their dynamic parameters. We are now exploring possibilities of using of genetic algorithms in order to calibrate agents fully automatically.

7 Acknowledgement

This work is partially supported by Ministry of Education, Youth, and Sport of Czech Republic – University spec. research – 1311.

8 References and citations

[1] S. Guberinić, G. Šenborn, B. Lazić. Optimal Traffic Control. CRC Press. 2008.

[2] O. Kosuke, T. Hata, T. Maetani, M. Harao, K. Hirata. Development of a Multi-Agent BASED Generic Traffic Simulator. *Lecture notes in computer science, 4012:249-260,* 2006.

[3] D. E. Ferreira, E. Subrahmanian, D. Manstetten. Intelligent agents in decentralized traffic control. *IEEE Intelligent Transportation Systems Conference Proceedings*, page 705-710, Oakland, August 25-29 2001.

[4] J. France, A. A. Ghorbani. A Multiagent System for Optimization Urban Traffic. *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology* (IAT'03), page 411-414, Halifax, October 13-17 2003.

[5] V. Hirankitti, J. Krohkaew. An Agent Approach for Intelligent Traffic-Light Control. *Proceedings of the First Asia International Conference on Modelling and Simulation* (AMS'07), page 496-501, Phuket, 2007.

[6] L. Filippo, S.G. Ritchie. A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transportation Research Part C: Emerging technologies, 5:507-527*, 2002

[7] S. Russel, P. Norvig. Artificial Intelligence: A Modern Approach (2nd Edition). Prentice Hall. 2002.

[8] M. Wooldridge, N. Jennings. Theory and Practice. *The Knowledge Engineering Review, 10:115-152,* 1995

[9] H. Taale. Comparing methods to Optimise Vehicle Actuated Signal Control, *Proceedings of Eleventh International Conference on Road Transport Information and Control*, page 114-119, London, March 19-21, 2002.

[10] D. Hartman. Head Leading Algorithm for Urban Traffic Modeling. *Proceedings of the 16th International European Simulation Symposium (ESS-2004),* Budapest, October 17-20, 2004.

[11] S. Pengjian, W. Meng, K. Santi. Fractal Nature of Highway Traffic Data, *Computers and Mathematics with Applications, 54:107-116*, 2007.

[12] L. D'Acierno, A. Cartení, B. Montella. Estimation of urban traffic conditions using an Automatic Vehicle Location (AVL) System. *European Journal of Operational Research, 196:719-736,* 2009

[13] A. K. Cooper, H. W. Ittmann, T. Stylianides, P. M.U. Schmitz. Ethical issues in tracking cellular telephones at an event, *Omega 37:1063-1072*, 2009.

[14] D. Arnaoud, R. Mandiau, S. Piechowiak, S. Espié. A behavioral molti-agent model for road traffic simulation. *Engineering Applications of Artificial Inteligence, 21:1443-1454*, 2008.