University of West Bohemia
Department of Computer Science and Engineering
Univerzitní 8
30614 Plzeň
Czech Republic

System for comparison of traffic control agents

Richard Lipka

Supervisor: Pavel Herout

Technical Report No. DCSE/TR-2009-01
April, 2009

Distribution: Public

Abstract

With increasing of traffic density on roads and especially in cities, optimization traffic control is becoming more important. Several methods were created, to calculate static traffic plans for specific area, or to allow dynamic traffic control on each crossroad. Spreading of computers allowed using advanced methods of artificial intelligence directly in controllers of traffic lights. Using of software agents represents one of newer approaches to traffic control.

In this work, traffic control agents proposed in last ten years are discussed. Their basic properties are described and agents are divided into groups. Because it is difficult to estimate agents' abilities from their description, we are presenting system for their comparison. This system allows performing comparison of different traffic control agents in the same environment, and with the same traffic conditions. The tests are performed on Java Urban Traffic Simulator, tool for traffic simulation developed on our department.

Keywords: urban traffic simulation, software agents, agents' comparison, traffic optimization, JUTS

Table of contents

1 Introduction

Road traffic control is one of typical problems of optimization. With development of road traffic optimization, paradigm of software agents is becoming popular. Every year, several new traffic control agents are proposed. They are used in traffic lights controllers, embedded in intelligent transportation systems, or in decision support systems. In some cases, they are replacing classical AI approach. But no ultimate solution has been found. Traffic is by its nature to dependent on many variables, which may not be easily modelled. It is basically behaviour of drivers, which is highly individual and may differ in various areas. It has been shown, that methods of traffic control, effective in one country or even city, were failing in another one ([KRAT]). Because of this, it is still important to attempt to find search for new ways of traffic control and test the old ones in different conditions.

Concept of software agents has first appeared in seventies. Carl Hewitt used it to describe self-contained, communicating and concurrently running processes, used in distributed artificial intelligence system [NWAN].

Very general definition of software agent is proposed in [RUSS]:

> *Anything that can be viewed as perceiving its environment through sensors and acting upon that environment trough actuators.(p. 32)*

According to this, agent has to be able observe its environment and it has to have ability to influence it. More specific description of agent is in [WOOL]. According to it, agent is encapsulated computer system deployed in environment, which is capable of flexible and autonomous activities, in order to reach specific goal. Four basic properties of software agent are mentioned there – autonomy (the ability to work without control of other systems), reactivity (the ability to react on impulses from environment), sociability (the ability to collaborate with other agents in the environment) and initiative (the ability to perform actions in order to achieve long-term goal). Using these properties, software agents may be classified to different groups, agents fulfilling all of them are called intelligent agents.

Agents represents natural way how to model how to model traffic control. In last ten years, many ways how to use agents to achieve at least semi-optimal traffic control were proposed. They use large scale of methods to find plans for traffic lights, which will assure the best possible traffic conditions. Very simple, reactive agents are used aside from complex, deliberative agents equipped with learning abilities or internal simulation models. In order to search global optimum, centralized agent systems are proposed along with decentralized and even with systems with agents without any form of communication.

During search for road the traffic control agents, we have noticed, that there is no reliable method or set of tools for comparison of the different agents. Usually, in the article where an agent is described, there is only its comparison with some type of basic traffic control (such as static time plans) or comparison of several types or settings of the same agent. Used criteria are different, used traffic networks are different and of course, used simulators are different. There is no simple way how to find if one agent is really better than another, according to chosen criteria. That is why we have decided to design system for comparison of different traffic control agents. We hope that it will be possible to used described method for comparison of all types of control agents, when simulator of environment they are controlling will be available.

This work consists of four main sections. In section 2, a very brief introduction in problem of traffic modelling and simulation is given. It is focused on cellular automaton and agent-based simulations, which are used in our work. Section 3 is showing basic issues of traffic control. Section 4 contains overview of existing agent technologies, used in traffic simulation. Part of this section is description of 7 agent-based traffic control systems, we want to compare. Section 5 is description of our comparison system. It also overview of experiments we want to perform.

2 Road traffic modelling and simulation

Road traffic modelling and simulation became an important issue in fifties, with the development of road transportation. Crowded roads in cities led to efforts to design better traffic networks, to optimize traffic control, especially the automatic one (first automatic traffic lights were installed in Cleveland in August 1914) and to improvements of the Highway Code. The first models were created in order to predict movement of vehicles on highways and freeways [CHAN]. With increasing of computational capacity, models and simulations also become important in design of traffic lights.

There are two main approaches to traffic – macroscopic and microscopic. Macroscopic models are the oldest. They are designed to work with traffic flows, vehicles density and other macroscopic values. Usually, they are using equations, similar to the ones from fluid dynamics. They cannot provide any information about single vehicle. Microscopic models are dealing with single vehicles and describe their interaction. They can provide also macroscopic values, but generally, they are much more computational demanding. Their recent expansion is allowed by easy access to hardware powerful enough to compute them.

Using of modelling and simulations becomes part of design of roads and traffic lights. It can provide information about behaviour of vehicles in changed conditions of newly designed traffic network. Simulations can be also used in decision support systems for traffic control. Such systems are can be used in cities or on important, controlled freeways, where operators can adjust traffic control parameters (such as time plans or signals at variable signs) according to actual traffic situation [OSSO].

2.1 Numerical modelling

First traffic models were based on numerical analysis of fluid dynamics. In 1955 mathematicians Lighthill and Whitham [LIGH] created theory of kinematic waves, used for description of flood movement in long rivers. They showed the same theory may be used for description of vehicle movement in crowded roads. Fluid dynamics based model providing only macroscopic information. They are able to describe formation of congestion or propagation of wave of vehicles, but cannot be used to observation of single vehicle. The resulting equations were used in classical control theory. Originally, they were used only to description of one isolated road, or road with several road ramps (due to similarity with river and tributaries), because numerical description of complicated traffic network was too difficult. In last ten years even networks has been considered, due to advances in informatics.

Traffic model used in UTIA ([KRAT]) is another example of this approach. The state of crossroad according to [KRAT] is described by the queue lengths in all lanes and by a traffic intensity of the arriving vehicles in specified time. For each upstream lane (see Fig. 5, where upstream and downstream is explained), state equation is created, using data about traffic intensity, capacity of the lane and length of the green signal. State equation is showing how long will the queues be with given data. In similar way, an output equation is created. The output equation is used to calculate traffic intensity in downstream lanes and occupancy of the upstream lanes. Then, equations for modelled lanes are placed into matrices. The whole simulated network, its control and traffic flow is described as set of matrices and simulation is performed by operation with them.

2.2 Car-following model

Another approach to traffic modelling is based on observation of single vehicles. This is called a microsimulation. The oldest type of this model is car-following. First proposal of such model comes of 1953, but firs used model is from the year 1957. Chandler, Herman and Montroll [CHAN] described in high detail model, based on the idea that cars in traffic flow are following each other. The first car is trying to achieve maximal allowed speed and cars behind it are adjusting their speed to maintain safe distance. Cars in the one lane are represented as points, with some distance between them. Distance is calculated as safe distance between vehicles and length of vehicle (see on Fig. 1). The movement of each vehicle can be then described by one ordinary differential equation and simulation is reduced to problem of solving of set of equations. This model was tested in real condition ([CHAN]), with eight cars, connected by wire on reel. Since the fifties many enhancements of this model were done, such as dependability of safe distance on speed and road, collision avoidance model, models of human behaviour and decision and linearization [BRAC]. In some models, passing cars trough each other in one lane is allowed, to simulate overtaking [JAMI]. Till now, this is most common way of traffic simulation. It is used for example in widely spread Aimsun simulator.

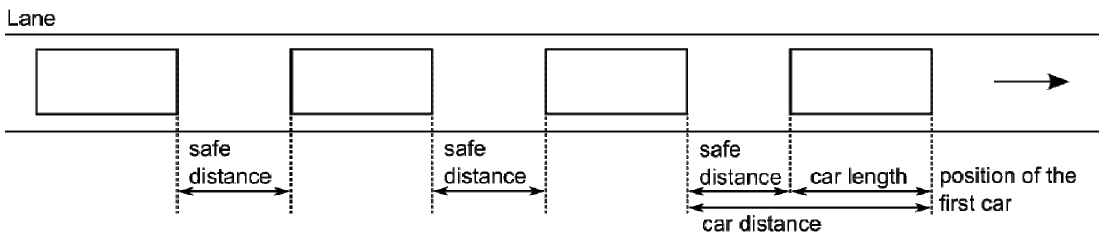

Fig. 1: Car following model

2.3 Cellular automaton

Cellular automatons (CA) become popular in seventies, with Conway's game of life. It is the best known 2D cellular automaton, designed to mimicking growth of cells. Cellular automatons are operating with state space created by cells. Automatons are working in discrete time steps. In each step, state of all cells is recalculated. State of cell is given only by state of cell, its neighbours in previous time step by applying rules. Automaton has no memory. Each cell can change its state according to set of rules Experiments with cellular automatons showed, that even 1D cellular automaton can produce useful results. Simple 1D cellular automaton is 1D array of cells, which can be in two states – "empty" and "occupied". For traffic simulation, cellular automaton marked as Rule 184 is important [WOLF]. This automaton represents simple model of traffic flow in one lane. "Vehicles", represented by occupied cells, can "move forward" only of there is free space in front of them (direction of "movement" is from left to right, or from lover indices of array to the higher ones). In other words, if there is free space in front of cell, it becomes empty in the next step and the cell in front becomes occupied. Rules for all combinations of cell and its neighbours are at Fig. 2.

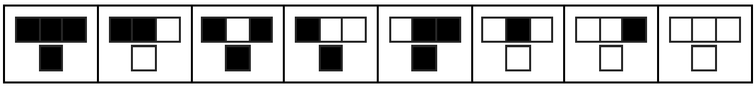

Fig. 2: Rule 184

Nagel and Shreckenberg in 1992 [NAGE] created one-dimensional cellular automaton, designed to model behaviour of vehicles on highway. As basis, they used cellular automaton originally designed to model dynamics of liquids. It has several advantages in comparison with car-following, especially more intuitive approach to space occupied by vehicles. Cars are not represented as points, but as occupied cells. Due to this, it is easy to prevent cars to passing through each other (which is allowed in some car-following models). This model is intensively studied and improved in last years, to achieve more convincing results.

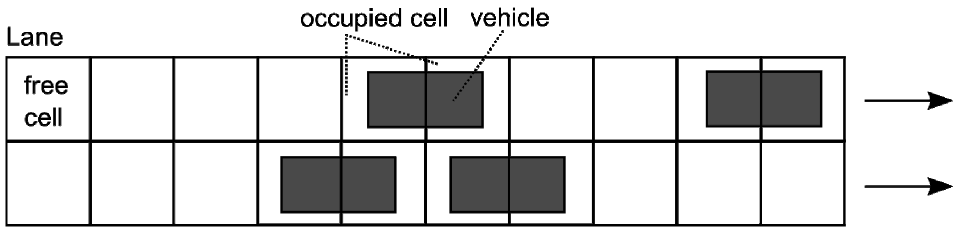

Fig. 3: Cellular automaton for traffic simulation

2.3.1 JUTS

Nagel-Schreckenberg model was also used as basis for JUTS (Java Urban Traffic Simulator, [HART]), simulation tool developed on our department. Originally, N-G model was designed to simulate traffic on highways, so several changes were performed. Cells in JUTS are smaller (2.5 m in JUTS instead of 7.5 m in N-G model), it is possible to use vehicles longer than one cell (Fig. 3) and it is possible to create static paths for vehicles. Simulations based on cellular automatons are usually representing vehicles as autonomous objects, capable of movement in space created by cells of CA. The CA is used only to determine which space in lane is occupied by vehicle. Further description of JUTS is in section 5.1.1.

2.4 Agent simulations

Increase of computing power allowed using more complex simulation systems. Several new simulators are using agents to model each vehicle in the simulation. Instead of having vehicles, whose behaviour is based on statistics data, each vehicle is connected with agent, acting usually as driver. It is often in agent simulations, that not only cars, but also traffic lights, roads and crossroads are modelled as software agents (simulation in [MIZU] is created in this way).

Typical example of this approach is [KOSU]. Each vehicle in the simulation is represented as an agent, with several attributes – position, speed, scope, destination and behaviour. It has set of rules, which allows searching for the optimal path to its destination. Vehicles are injected to simulation by random generators and they are to move towards their destination in each simulation step. Instead of describing probabilities of turning of vehicles on crossroads or preparing routes, they are recalculating their movement in each step. Due to this, vehicles are able to react to new obstacles in traffic network (such as accidents), however, there is question how far is their behaviour similar to human behaviour. Similar approach can be found in [MIZU], but there not only vehicles, but also roads and crossroads are created as independent agents.

Different example of using agent technology is in [BALM]. Each vehicle is connected with agent, which has 24-hour time plan, mimicking human behaviour (typically it contains activities as “go to work”, “go home”, “go with child to kindergarten” and so on). It can decide between walking, using public transport and using car, decision is based on utility function, working with price of travel, time spent on the way and coefficients obtained from statistical survey. It is also able to learn from past experience and from time to time trying new paths or type of transport. At the end, a set of travel plans is created, which may be used in simulation.

2.5 Calibration data

Quality of traffic simulation is heavily dependent on calibration. The oldest models were describing traffic as static and linear problem, which applicable for one road in short time period, but not for complex network in longer time span. It is necessary to correctly modify amount of vehicles in different parts of simulated network, as well as their behaviour. In simulators based on car-following model or cellular automaton, two main methods are used. One is based on prepared routes of vehicles and second is using generators of pseudorandom numbers. Both approaches require large amount of data, collected in real conditions.

Simulation based on prepared routes of vehicles is usually working with constant amount of vehicles in simulation. Each vehicle has its schedule, where all movement is described (used for example in [BALM]). The typical vehicle will be placed on some parking place in residential area. In the morning, it will try to reach some place in the centre of city or near factory, and afternoon it travels back (see on Fig. 4 – vehicles are moving only on marked paths). It leads to very convincing model of human behaviour, but collection of data is difficult. To achieve realistic results, it is necessary to observe huge amount of vehicles in a long time span. Another way to obtain vehicle routes is their computation. Instead of observing vehicles, macroscopic variables are observed (such as flow of vehicles or number of vehicles in the area). Using this data, routes of vehicles are calculated by path searching algorithm. It is possible then to use simulation for predicting of vehicles behaviour even for changes of traffic network.

Using of pseudorandom numbers is more typical for cellular automaton based simulations, but it is also used also in Aimsun, based on car-following. Instead of having prepared path, vehicles are travelling through simulation blindly. They have no target, and they use pseudorandom number generator to determine path at each crossroad. Instead of waiting on parking places, vehicles are injected to the simulated map at the edges (Fig. 5). This approach allows describing behaviour of vehicles by collecting statistics on roads and crossroads, instead of long-time vehicle observation.

Many crossroads has automatic systems to count number of passing vehicles, so data collecting is easier.

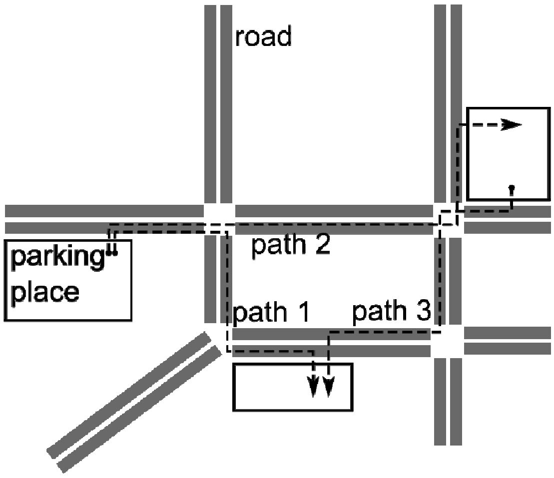

Fig. 5: Following paths

In [BRAC2] two important points of microsimulation models are discussed (amongst others). At first, safe distance maintained in simulators is compared to distances between vehicles in real world and it is show, that real drivers are not following advices from Highway Code and keep much smaller distance. The second important is result of different length of time steps. Using of too short time step in simulation leads to unnaturally fast reaction times of driver and also to too often changes of speed. Behaviour of real drivers, including violation of traffic rules, observed in real world, is discussed and described in terms of formal logic in [DONI].

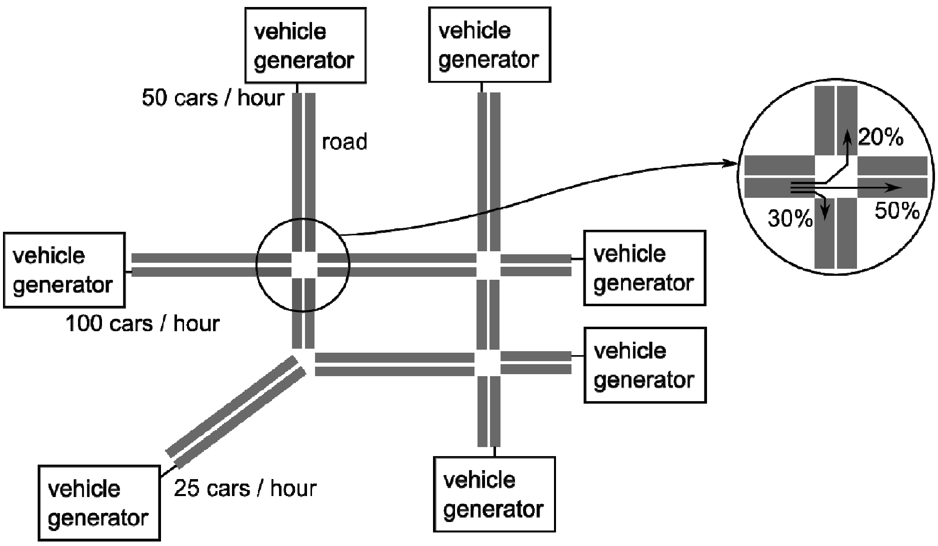

Fig. 4: Using of random number generators

3 Problem of traffic control

With increasing amount of road transportation, traffic control is becoming more important. Its importance is rising with using automatic systems. When each important crossroad was controlled by policeman, he was able to observe amount of vehicles in each lane and control them accordingly. It required experience, but it was possible to achieve good results. With the appearance of automatic traffic lights, the advantage of human decision was lost. First automatic traffic lights were based on simple static time plan, the same sequence was repeated all the time. It is obvious, that this cannot be very effective, when traffic flow is not constant. With development of cybernetics and informatics, more effective systems were designed, to improve results of traffic control.

3.1 Terminology

- Downstream – traffic lane behind crossroad. There may be several downstream lanes for one upstream lane (see on Fig. 6).
- Upstream – traffic lane in front of crossroad (see on Fig. 6).

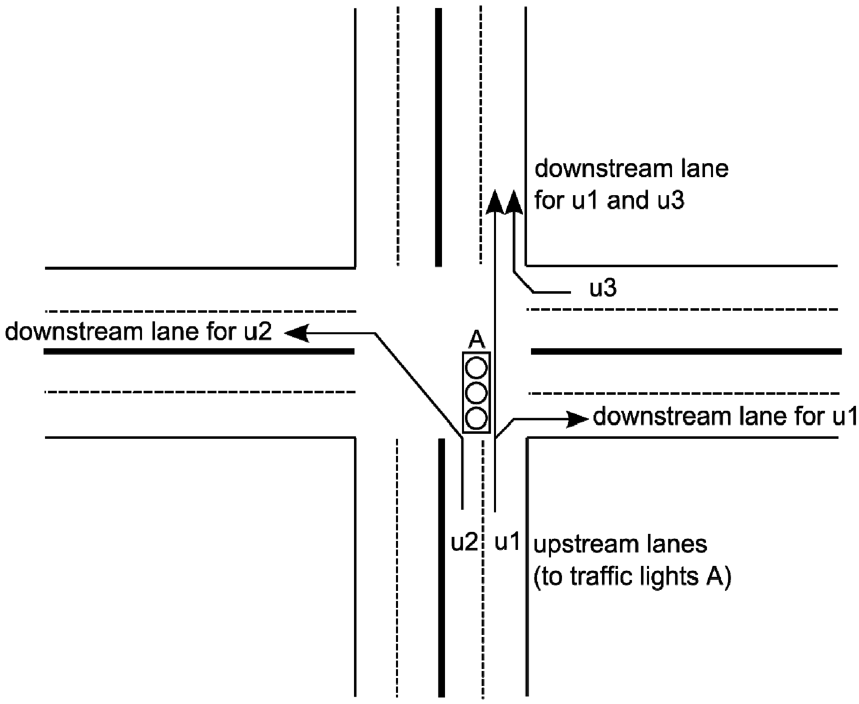

Fig. 6: Upstream and downstream

- Signal cycle – traffic lights are often operating in repeated cycles. In static traffic control, signal cycle is repeated over and over again (see on right side of Fig. 7).
- Signal group – group of traffic lights sharing the same time plan (for example traffic lights on roads 1 and 3 on Fig. 7 are in one signal group).
- Signal phase – part of signal cycle, when all signal groups are set to achieve safe situation in crossroad. In static control, signal phases have unchangeable order (time plan on Fig. 7 is using two phases and two switching processes between them).

- Traffic control agent – software or hardware module used to control one crossroad.
- Traffic control system – system designed to control whole traffic network. In this work, we are dealing with systems based on using traffic control agents, but sensors in lanes, high-level managing agents or databases with description or historical data about traffic network are also part of traffic control system.

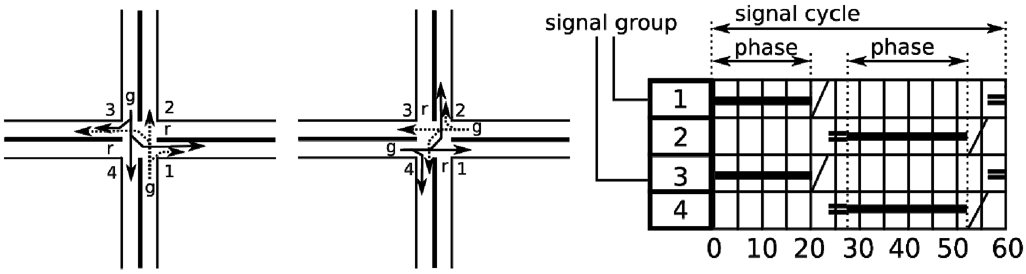

Fig. 7: Example of static time plan

3.2 Road traffic measuring

In simulation, it is easy to obtain any kind of data, required for the work of agent. But in real world, systems to measure traffic situation can be very expensive. It is easy to measure number of cars passing checkpoint, or to measure their average speed. It is more difficult to measure length of queue (which is necessary for agents in [HIRA], [LOPE] and [KOSU]) or to determine amount of free space in road, as required in [HIRA]. The most common methods are based on induction loops in the road, capable of detecting moving metal objects and IR gates able to detect passing vehicles. These methods can be used only to collect information about amount of passing vehicles. They may be connected with speed radar to gain also information about average speed of vehicles. Radar with IR gate can be used also to determine length of vehicles. Sensors based on recognition of acoustic patterns are also being used [GIBS]. More complicated systems may be based on image recognition software, with cameras placed on traffic lights on crossroads. They are able to measure length of queues and they also may be used to collect data about specific passing vehicles, by reading their plates. In the cities where electronic toll is used, data from toll gates can be also used to provide information to traffic control systems. Combination of toll gates and image recognition systems can provide vast amount of information about traffic, but such complex systems are criticized for reducing of privacy and possibility of abusing of private data.

The most important problem is detection of queues and measurement of length of queues. Induction loops or IR gates are able only to measure if vehicle is passing around. Because of this, three main positions for detectors are used ([KRAT], see Fig. 8). Detectors placed at the stop-line of crossroad are used to detect if vehicle is waiting at the traffic lights. It cannot provide any information about queue. Far detectors are installed 30 – 40 meters from the stop-lane are capable to detect, if the queue of vehicles is reaching them. Position of this detector is very important for the quality of traffic control. If it is too close to crossroad, queue is detected very often. If it is placed far from the crossroad, even long queues may stay undetected. On important roads, also strategic detectors may be used. These detectors are placed far from crossroad, so queues are usually not reaching them. They may provide information about average traffic speed and they are also useful to detect extreme traffic congestions. Information obtained from all three positions can be used to estimate the length of queue ([GART], [KRAT]). Image recognition systems, placed above crossroad, are able to recognize if one, two or more vehicles is waiting at the crossroad, but they cannot recognize more than four or five vehicles in the lane.

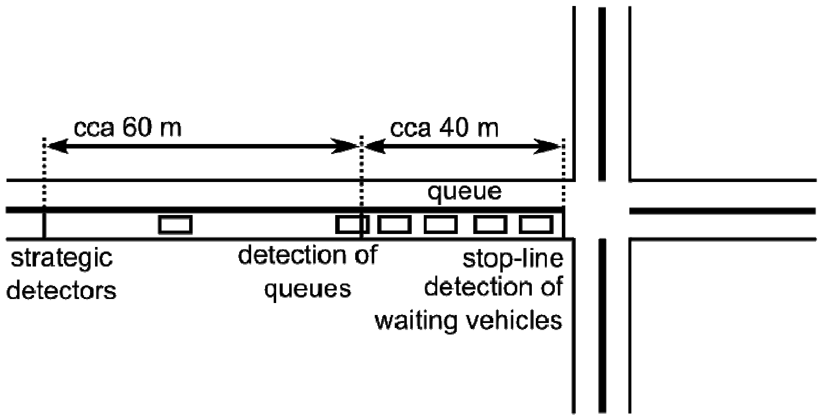

Fig. 8: Position of detectors

Another approach is using information obtained by observation of measuring vehicles. The measuring vehicle is sending information about its position, average speed and possibly other characteristic. From this, information about traffic character around measuring vehicle can be estimated (for example, if average speed is slow, we can assume that vehicle is moving trough congested area). To gain adequate information about traffic situation in the whole network, it is necessary to use enough of measuring vehicles. Penetration of 10% vehicles should be sufficient, according to [COME]. Increasing number of measuring vehicles will lead to improvement of traffic optimization. In last years, GPS navigations and wireless networks coverage in large cities are spreading rapidly. It is conceivable that even vehicles of common citizens will be equipped with some kind measuring devices. Also, most of drivers in western countries are using cellular phones. Because cell phone position can be tracked by using of current technology, it has been proposed to measure position of cellular phones and use it to reconstruct information about traffic situation. As mention previously, these technologies are controversial due to the loss of privacy. They may be used only if privacy issues will be solved satisfactorily. But even using only information obtained from vehicles of a public transportation system can be enough to optimize control of the main roads [COME].

In large cities, public traffic priority is also used. Traffic lights on crossroads on path of public transportation (especially of bus lines) are equipped with devices designed to detect approaching public transport vehicles. System is trying to optimize timing of traffic lights in order to allow non-stop ride (or at least as fast as possible) for public transport vehicles. Similar systems are used to prioritize emergency vehicles (such as fire engines or ambulances) [GROS]. Vehicles with priority are equipped by emitter, which identifies them to traffic control system. Emitters are using visible or infrared light flashes at a specified frequency. Receivers are mounted on traffic lights. When signal of prioritized vehicle is received, signal plan is changed to allow fast transit. The main disadvantage of this system is the risk of abusing signal system by unauthorized emitters of priority signal [PLUN].

3.3 Analytical approach

When static time plans were only option, several mathematical methods were proposed to calculate optimal times for green and red signals. Detail description of optimal traffic control on one isolated crossroad is in [GUBE]. On more than 300 pages is detail description how to create optimal static control of one crossroad, without any regard to surrounding crossroads. This can illustrate how difficult would be to create optimal control of the whole network (or to prove, that chosen control really is optimal, in terms of chosen criterion). Other problems raises from non-linear behaviour of traffic flow (as is shown in [SHAN]), which leads to even more complicated mathematics. Because of this, traffic control is handled more by heuristics method of artificial intelligence.

3.4 AI approach

Automatic counting of vehicles passing trough crossroad allowed adaptive control of traffic lights, based on artificial intelligence. Instead of trying to compute optimal cycle length, each cycle is modified according to current situation. Several techniques is used, such as neural networks [MING], expert systems, various kinds of logical reasoning ([HOEK]) or evolution algorithms ([HOAR]). These techniques are also used inside of traffic control agents.

3.4.1 Genetic algorithms

Static time plans can be created by genetic algorithms, instead of analytical calculations. Exact analysis in network with several crossroads is too difficult, so genetic algorithms may offer easy, but hardware demanding way to design time plans. It requires only ability to decide which set of time plans has better results. In [KWAS] genetic algorithms are used to calculate length of signal cycle, duration of phases and even their order. In one chromosome all time plans are described, they evolve together. Selection mechanism was choosing plans with minimal waiting time in queues and maximal fluency of vehicle flow. It is shown, that suitable time plan is obtained already in seventh generation, in network with 7 crossroads, with population 100 time plans and 1000 time steps for each simulation (so only $7 \cdot 10^5$ simulation steps was required).

3.4.2 Expert systems

For dynamic traffic control genetic algorithms are not convenient solution, due to their computational complexity. One of the ways is based on expert systems. These systems are based on of rules, where each rule is described as set of conditions and set of actions, which should be applied. Control program is trying to match conditions of rules to situation on crossroad and set lights accordingly. They may use fix set of rules, or update rules database with new one.

3.4.3 Agents

In traffic control, agents are used more as general abstraction. In the beginning of the next part we describe why traffic lights may be seen as agents. This work is dealing with control based on active traffic lights and passive vehicles (in terms of traffic control, vehicles are not active participating on control), but some agent approaches to traffic control utilizes also active vehicles. This approach is described in [DRES], where agents are placed also in cars. Car agents are informing central point about their position, speed and destination, and central point is sending back advices about path and trying to optimize crossroad control using reservation protocol. Because central has information about paths and speed, it can predict amount of cars on crossroads and prepare time plans accordingly [BALA].

3.5 Real-world traffic control systems

Methods of traffic control are one of the most important parts of intelligent transportation systems (ITS). It is large group of computer systems, designed to optimize all kinds of traffic. They are used by transport companies to plan routs for cargo, by municipal authorities to improve traffic conditions in cities or to optimize mass transportation or by administrators of highways and freeways to collect toll. Surveillance systems designed for observation of traffic may be also used by police to enforce law. Some kind of ITS is now deployed in most major cities around the world ([MONA]). Most of them have developed from systems for traffic surveillance.

Usually, they consists from module for data collection (depending on purpose of ITS, it can be detectors around roads, toll gates or tracking devices in vehicles), infrastructure for traffic control (such as traffic lights, variable signs or communication with drivers) and decision support system for the operator. So far, important decisions are taken by human operators, computers are used only to provide information necessary to make decision. But some subsystems, such as traffic lights control are usually fully automated and operator’s decision is required only in critical situations ([MONA]).

4 Agent approach to traffic control

According to definition in [RUSS] (see in the chapter 1), dynamic traffic lights can be naturally modelled as agent. If they are able in any way react to real state of transportation, they has to have some way to observe their environment (from the simplest possibilities such as button for pedestrian or detection of vehicle waiting just in front of the traffic light to complicated systems based on image recognition, queue detection and information shared amongst several crossroads) and naturally they are able to interact with environment (although only indirectly, they can only switch on or of signalling lamps). To all this, traffic lights are distributed and often able to control one crossroad independently on the other, they can be seen as autonomous. And because traffic flow can be taken as a form of information, all types of traffic lights have ability to react on their neighbours (of course, some systems contains also common ways of communication, based on computer networks). And all systems controlling traffic network have goal, they are trying to improve traffic situation by various criteria. Thus, traffic control also fulfils requirements to be seen as intelligent agent, according to [RUSS].

First dynamic control systems were using vehicle actuated signal control (VASC). Agent based on VASC would be only reactive, able to react on immediate situation. With development of computers and decrease of their prices, more sophisticated control methods may be used. Using of methods of artificial intelligence allows creating goal-driven behaviour, when crossroad control system is trying to achieve long-term goals. Other methods allows learning, negotiating between systems on different crossroads. Agents exploiting these methods fulfil conditions required from learning and smart agents.

4.1 Agent structure

Abilities of agents and tasks they will be able to solve are highly dependent on structure of agent, but some parts are common for all types of agents. Agent has to be able to observe surrounding environment, thus it has to have module able to obtain data from real hardware sensor. It also has to interact with its environment trough actuators. And inside agent has to be program able to select action for actuators according to state of sensors (general scheme is on Fig. 9). In [RUSS] five general agent structures are described, named according to ability of their inner structure:

- Simple reflex agent: It chooses action according to current state of sensors.
- Model-based reflex agent: It has internal model of the environment updated according to current state of sensors (so it remembers past of its environment or things that cannot be now seen on sensors). Action is chosen according to internal model.
- Model-based, goal-based agent: It has internal model as previous type and also set of goals it wants to achieve. This agent is able to predict impact of its actions (using internal model) and chooses action that leads to achievement of its goal.
- Model-based, utility-based agent: This agent is similar to previous, but instead of goal, it has utility function. Again, using internal model, it is trying to predict

impact of its action and chooses action that leads to maximization of utility function.

- Learning agent: It is able to obtain feedback on its performance (it can review impact of its actions or obtain critique from some other system). This feedback is used to improve selection of actions. Agent is also using problem generator, which allows it to explore - to try new, suboptimal actions, which may lead to new, better solution.

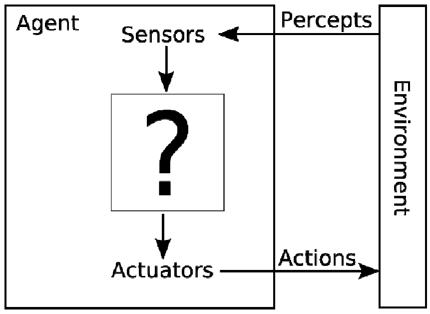

Fig. 10: General agent scheme

Another common model is called Belief-Desire-Intention (BDI). In this model, agent is composed from three main parts. Model of surrounding world (possibly including knowledge about other agents) and rules allowing predictions about changes of world after applying some actions are called belief. This word also shows, that agent's belief don't have to be true. Agent's goals, utility function or other mechanism used to choose the best action are called desire. And actions already chosen to be performed are called intentions (see scheme on Fig. 10). BDI model is wide spread and there is lot of tools allowing implementation of own, BDI-based agents (such as Jadex - [JADE]).

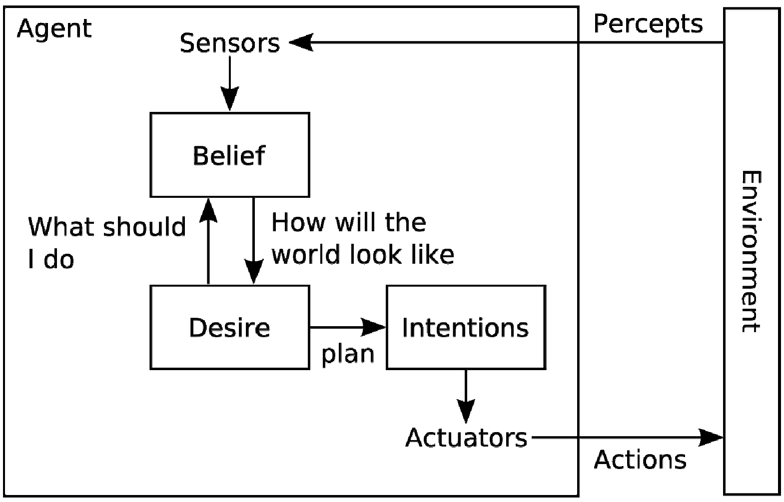

Fig. 9: Schema of BDI agent

4.2 Reactive vs. Deliberative agents

Development of software agents leads to creating more and more complex agent systems. Agents are becoming “smarter”, with sophisticated inner models of outer environment and complex planning systems. This type of agents is called “deliberative” agents, as they are capable of planning. But another approach exists, an approach based on mimicking real world organisms – reactive agents ([BROO]). Reactive agents are not using any kind of representation of outer world. [BROO] claims, that “world is the best representation of itself”. Reactive agents are choosing their activity only according to information from their sensors; they are not capable to remember past states of the environment.

Reactive agents are simple, in comparison with deliberative agents, but in some conditions, they are supposed to work equally or even better. They are easier to implementation and maintenance. Despite their simple nature, it may be difficult to design them. Usually, they are expected to work in cooperating groups, communicating directly or indirectly. If they are designed efficient, they may show more intelligent behaviour, than it can be predicted from their structures – this is called emergent behaviour.

The term reactive agent was originally developed for robots designed to avoid collisions in real environment [BROO]. Such robot is dealing only with obstacles and in one moment, it can observe state of the environment. Traffic control agents are in different position; they are supposed to optimize flow of vehicles in the traffic network. Values such as current average speed, traffic density or flow cannot be easily observed directly, they are computed from historical data. We have decided to omit this memory and we still consider and agent as reactive, if it is storing only short-term data about traffic (for example characteristic of last signal phase).

4.3 Agent deployment

There are several different positions, where traffic control agents can be placed. The most typical position is on crossroad. Each crossroad is controlled by one agent, who has authority to set all of its traffic lights. Other agents may be placed on important roads, to observe traffic situation and inform adjacent crossroads. Road agents may be also to work with variable traffic signs (suggested speed, information about forthcoming congestion, suggested detour). Using of variable message signs is described in [KATW].

Agents may be also arranged in hierarchical, usually tree-like, structure. Agents on crossroads are subordinated to managing nodes. This hierarchy is usually used to improve traffic control according to global criteria. Agents on crossroads are able to perform local optimization, so superior nodes may manage their collaboration – they possess more information about whole network. Examples of such structure are in [RONG] or [FRAN]. Problematic of roles and responsibilities on different levels of hierarchy is discussed in [KATW], also with suggestions for suitable position for human operator. Hierarchical structure can also be used, when each lane is controlled by one agent, and superior agent is controlling the whole crossroad ([SANC]).

4.4 Overview of exiting agents

A lot of different agents was proposed to optimize traffic control. In this section, there is a brief description of most typical agents and their way of work. We choose these agents, because they are representative examples of different approaches to agent-based traffic control. Because they are described in high detail, it is easy to use the articles they are described in as a basis for their implementation. Because usually agents do not have names, we use shortcuts from references to distinguish them.

4.4.1 ROOZ

Danko A. Roozemond in [ROOZ] proposes traffic control system based on interaction of three types of agents. Each crossroad is controlled by Intelligent Traffic Signalling Agent (ITSA). ITSA are grouped and each group is control by Authority Agent (AA). The last type is Road Segment Agent (RSA). ITSA is able to observe traffic situation on its crossroad. These observations are stored and shared with neighbour ITSA. Information about vehicle flow, together with chosen control strategies, road condition, information about weather and other correction factors are used to predict traffic situation in near future (prediction model is described in [ROOZ] in high detail).

To choose control strategy (which is in fact time plan for next cycle) ITSA is using information about current traffic, prediction and historical data. Control strategies have to be prepared by operator, but ITSA is able to learn, under which conditions is strategy appropriate. Each strategy is described as set of rules for lights and set of conditions when strategy is appropriate. AA is used to help ITSA to search global optimum, instead only optimize traffic flow on one crossroad. RSA are optional, if it is useful, they may provide additional information to ITSA. Typically, they are used to measure length of queue, because this information cannot be obtained by measurements in crossroad.

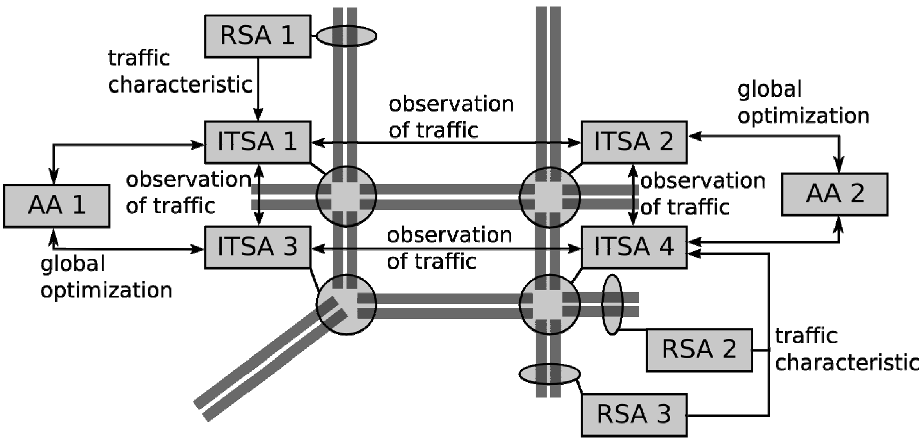

Fig. 11 : Deployment of ROOZ agent

ROOZ represents the most complex type of agent. It is capable of learning; historical data are stored and used to optimize traffic control. Agents on crossroads are capable to communicate with each other, in order to share information about traffic condition. And superior agents are used to centralized optimization of traffic control.

In contrast to FRAN (see below), ITSAs are not dependent on AA agents. They are able to communicate with each other and optimize traffic control.

4.4.2 FERR

In [FERR], 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 neighbour agents and together with opinion of neighbour 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.

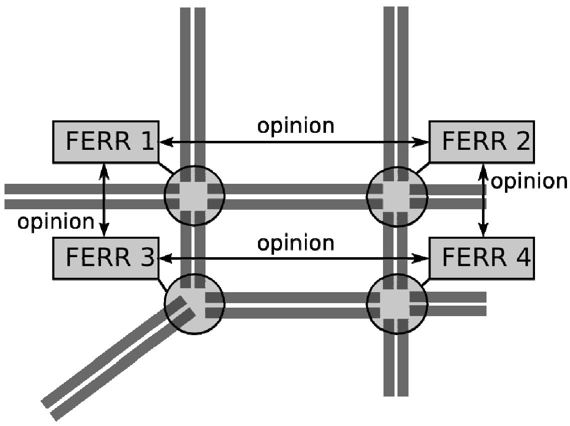

Fig. 12 : Deployment of FERR agent

4.4.3 LOPE

Agent described in [LOPE] is using fuzzy logic to modify time plan, according to vehicle flow. Each agent controls one crossroad and there is no communication among agents. Three different implementations are described. The first one is based on vehicle actuated signals system, thus each control phase has set minimal and maximal duration. Phase is kept for minimal duration and then traffic flow is observed. If vehicles are still passing crossroad, phase is maintained, till its maximal duration is reached. Then the lights are switched to next phase. Order of signal phases is static, agent cannot change it. [LOPE] adds several rules to this system, to take in account behaviour of real drivers (such as accelerating when amber signal is on lights, to pass crossroad before red signal will came).

Second proposal in [LOPE] is based on longer observations, instead simple detection of approaching vehicles it is using traffic density and average speed of vehicles. Traffic density and speed of vehicles is compared to stored “normal” values, based on long-term observation. If traffic density in one of lines is greater than normal, agent can increase duration of its green phase. If the speed is rising, agent is raising duration of amber during switching phases, to prevent accidents (more cars will try to ride trough crossroads on amber signal, because they cannot safely break). If the average speed overstep given maximum, green time may be shortened, to slow traffic in the lane (and similarly, if average speed is too low, agent may increase duration of

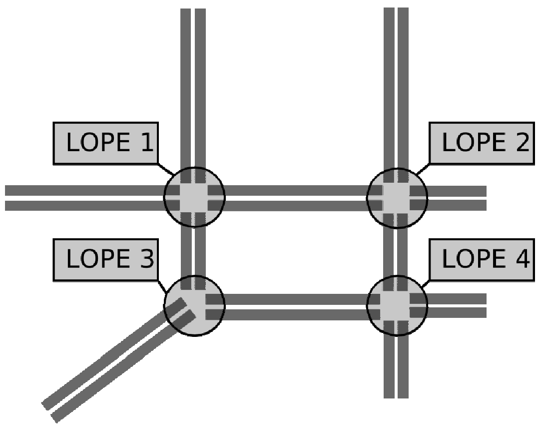

Fig. 13 : Deployment of LOPE agent

green signal).The third proposal is using queue lengths and average waiting times at the crossroad. Duration of green and red signal is derived from these values, in order to minimize waiting times.

First proposal of LOPE agent can be classified as reactive agent. It is capable of storing limited amount of information about past traffic situation, but only to compute current average speed or traffic density. These values describes current state of traffic situation, but cannot be easily measured directly. Second and third variety is using very simple model of environment, where lane is represented by average speed and traffic density.

4.4.4 HIRA

In [HIRA], agent is observing not only vehicles approaching to crossroad, but also free space behind crossroad, in downstream. To control traffic lights in one lane, agent need information about length of queue in the lane, availability of space in downstream, number of cars incoming to the queue within one second and number of cars leaving queue in one second (to determine, if during green signal, queue is prolonging itself). Agent has prepared cycle of all possible signal phases and its default duration. It can change duration and order of phases, according to actual traffic situation. To choose control strategy, it uses set of 13 rules. Moreover, agent is capable to change activation of green signal during one signal phase by using a mechanism of the friendly directions. Working of this mechanism may be seen at the Fig. 12. To the direction A, a set of friendly directions is found. These directions can use green signal simultaneously with direction A. When vehicles cannot move during active phase in one of its directions (because there are no vehicles in upstream or because there is congestion in downstream), one of friendly directions to remaining active

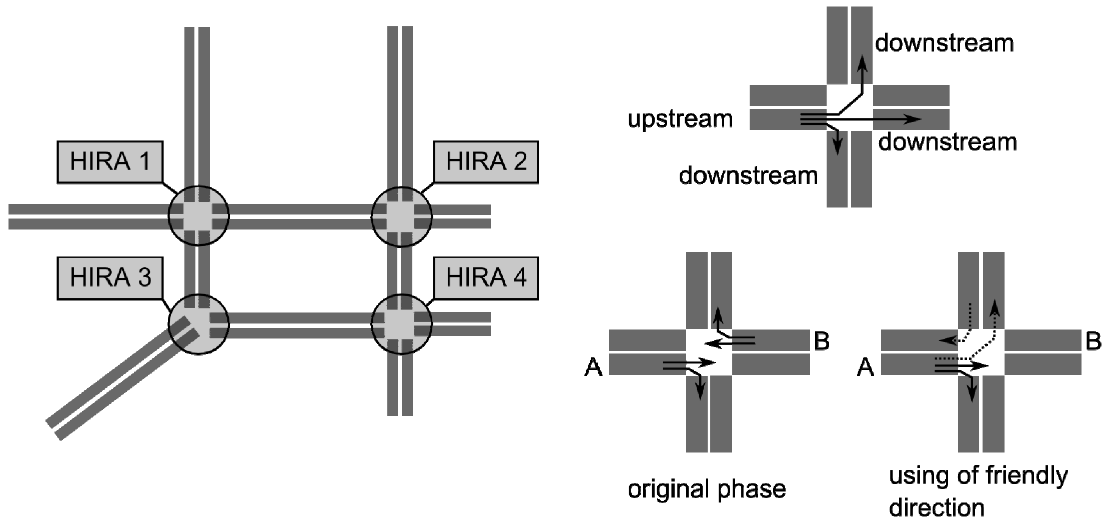

Fig. 14 : Deployment of HIRA agent

direction is chosen, for the rest of the phase. See the example at Fig. 14 – during one phase, directions A and B are set to green. If there are no vehicles in B or vehicles in B cannot continue because of congestion in downstream, signal on B is set on red. Consequently, one of friendly directions (marked by dotted line) is chosen for the rest of the phase and its signal is set to green.

At the end of [HIRA] paper, agent is compared to vehicle actuated signal control and proves itself better, in terms of average delayed time of each car. This agent has an interesting ability – due to observation of downstream lane free space, it can block traffic in lanes sending vehicles to crowded lane, thus preventing obstruction in the crossroad. Vehicles are accumulating in the upstream lanes and cannot enter crossroad, if they cannot leave it. HIRA is the best example of reactive, non interacting type of traffic control agent.

4.4.5 MIZU

[MIZU] contains description of collaborating agents on crossroads. The main goal of agents is to prevent traffic congestions, or to solve congestion of it occurs. To work properly, agents has to know distance to adjacent crossroads, to determine offset of green signal and allow vehicles to pass several crossroads continuously. Each agent is observing average speed and amount of vehicles in upstream lanes, to determine if congestion occurs. Using this information, agent is creating time plans for all its signal groups, trying to satisfy three main constraints.

The first constrain is assuring, that all control groups in one crossroad will use the same length of cycle, all of them will have some time for green signal and they will not create situation leading to possibility of collision. This constraint has to be fulfilled. Second constraint is based on negotiation with agents on adjacent crossroads. Agents on adjacent crossroads are trying to use the same length of cycle and set start periods for green signals in the way, in which vehicles can pass without stop. This influences mainly order of signal phases. The third constrain is using information about vehicles in upstream lanes. A road measurement agent is used, to provide information about each lane. Control agent is trying to let pass as many vehicles as possible and not to leave unnecessary green signal in lanes, where are no vehicles. This constrain has the biggest influence to the length of signal phase. Second and third constrain are used only for optimization, they may not be used if it is not possible due to first constrain.

MIZU is good example of reactive, social agent. It is not capable of storing any information about traffic (except of values required to determine actual average speed and traffic density). Adjacent agents are capable of collaboration without using central point, which makes the whole system more robust, than ROOZ or FRAN is.

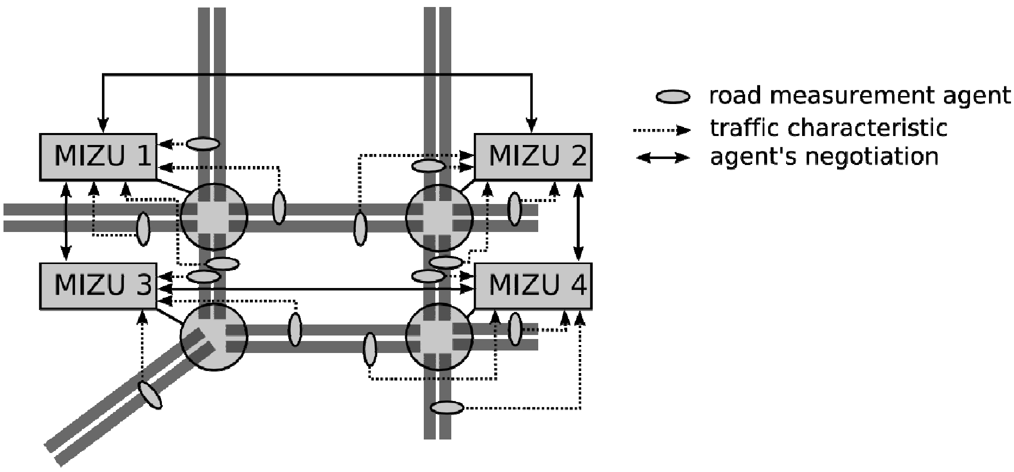

Fig. 15 : Deployment of MIZU agent

4.4.6 KOSU

[KOSU] shows three simple control agents. All types are observing only one crossroad and there is no communication amongst them. They are observing 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 another example of very simple, non interactive reactive agent.

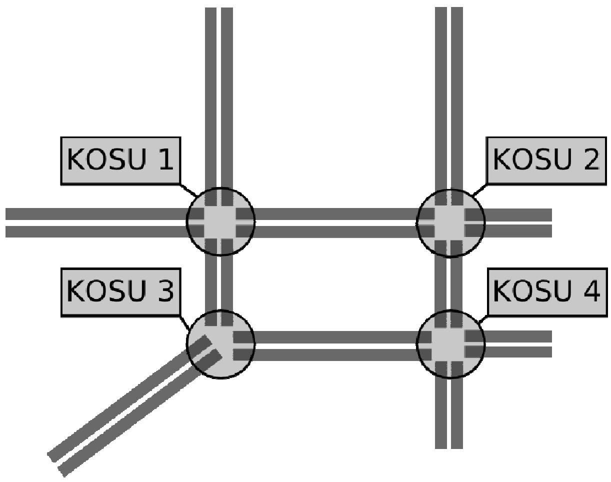

Fig. 16 : Deployment of KOSU agent

4.4.7 FRAN

[FRAN] shows hierarchical structure created from two types of agent. At each controlled crossroad, Local Traffic Agent (LTA) is placed. Its goal is to optimize control of its crossroad, disregarding global optimum. According to traffic density, LTAs can change duration of signal phases. At the end of cycle, LTA computes new duration of phases and sends this to its superior Coordinator Traffic Agent (CTA). Then CTA retrieves information about traffic densities in all neighbour crossroads to served LTA and using CTA's own expert systems decides how global optimal control should look like. If there is huge difference between control strategy created by LTA and CTA, CTA creates new time plan, to reduce difference and sends it back to LTA. Changes in time plan in CTA are created only with regard at traffic densities around neighbour crossroad, not to time plans of agents on neighbour crossroads – CTA is not capable of creating green waves.

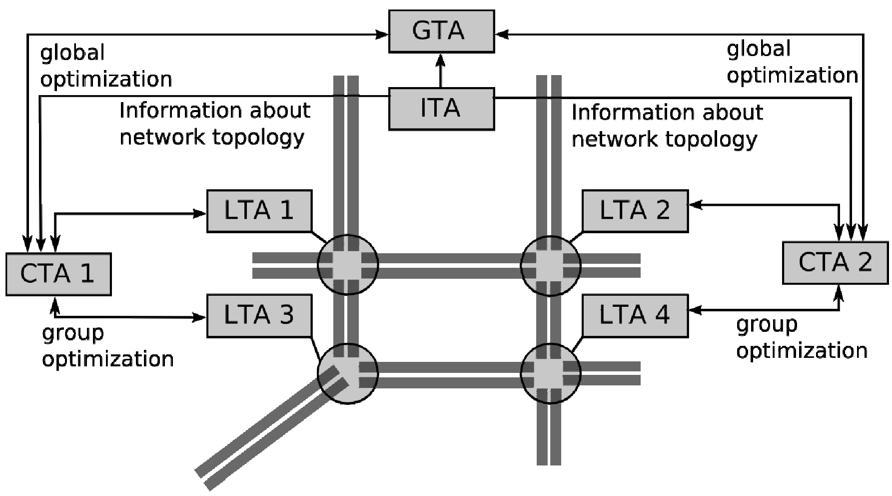

Fig. 17 : Deployment of FRAN agent

If controlled network is too large, Global Traffic Agent (GTA) may be used. This agent is connecting CTA and helps them to find better control. It works in similar way as CTA for LTAs. An Information Traffic Agent (ITA) is providing information about network topology, adjacency of LTAs and CTAs about groups of LTAs managed by one CTA.

FRAN system is using combination of the reactive LTAs and the learning CTA. There is no communication amongst LTAs, so without help of CTAs they are not able to optimize traffic between adjacent crossroads.

4.4.8 VASC

Vehicle actuated signal control ([TAAL]) is not usually denoted as agent, but we need describe it, because it was used as basis for reactive agents mentioned above (HIRA and LOPE). 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 (see Fig. 18). Lights are consequently switched to next phase.

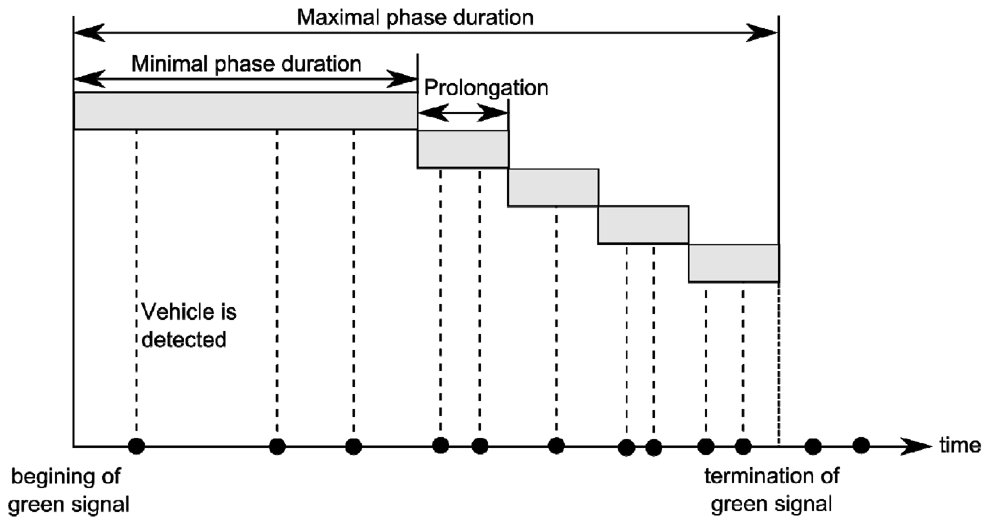

Fig. 18 : Example of timing in VASC

In following table, most important properties of described agents are shown.

Agent	Used values	Communication	Strategy selection
ROOZ	number of passing cars	between neighbours + superior agent	expert system
FERR	number of cars in upstream lanes	between neighbours	Optimization function
LOPE 1	number of passing cars	none	fuzzy logic
LOPE 2	traffic density, average speed	none	fuzzy logic
LOPE 3	queue lengths, average speed	none	fuzzy logic
HIRA	queue length, number of incoming and departing cars, downstream space	none	expert system
MIZU	average speed, number of cars in upstream lanes	between neighbours	distributed constrain satisfaction
KOSU Even	number of passing cars, queue lengths	none	expert system
KOSU 2	number of passing cars, queue lengths	none	expert system
KOSU 3	number of passing cars, queue lengths	none	expert system
FRAN	traffic density	superior agent	expert system

Table 1: Overview of traffic control agents

4.5 Agent groups

According to the used ways of communication between agents, agent-based traffic control systems may be divided into three main groups. We have decided to use communication ways as criterion, because realization of communication between agents is one of the most difficult problems. If crossroad is controlled by computer system, we suppose it is not so important if expert system, fuzzy logic or other methods of artificial intelligence are used. The same hardware can be used as platform to implement any of these methods. But if system requires connection between agents on adjacent crossroads, it is necessary to build and maintain infrastructure for communication. The groups are:

- Centralized: Agents deployed on crossroads are responsible for local control. To achieve better global results, a central managing agent exists. Local agents are consulting their plans with central agent and central agent is suggesting (or directing) changes. Central agent usually has full knowledge of topology of controlled network and it has to be able to gain information about traffic situation in the network. Systems FRAN and ROOZ from our list are belonging to this category.
- Social: In this system, agents are only managing their crossroads, there is no central point. But agents are still able to negotiate with each other. Usually, communication links exists only among adjacent crossroads. Global optimization is achieved by propagation of information among agents. MIZU and FERR systems from our list are belonging to this group.

- Non-interacting: Agents in this group are not able to communicate directly; they are managing traffic only according to their local information. They may use measurements of road traffic situation as a form of "communication channel", since traffic is affected by agents at adjacent crossroads. Typically, they posses information about priority of roads and turning directions or about average traffic characteristics in each lane. These information help them to optimize traffic without any knowledge about other agents. They can be set as fixed values, or calculated by some kind of learning process. Systems LOPE, HIRA and KOSU are belonging to this group.

5 Comparison of agents

Comparing different agents is a difficult problem. Usually, each agent is published with some tests of its performance. Due to this, there is no simple way to compare two agents from different articles. To make convincing comparison, three mains conditions need to be fulfilled.

- Compared agents have to operate in the same traffic network. Some agents are tested only in very simple conditions, such as two or more crossroads on one road. In real world we may need to deploy them in more complex environment.
- Traffic situation used for comparison has to be the same for both agents. It is also necessary to compare both agents in wide spectrum of different traffic conditions, for example agent in [HIRA] is effective in non-crowded road, but in crowded road, it effectiveness is decreased. Traffic condition should change several times during test, to test agent's ability to adapt.
- The same simulation engine should be used, to prevent ambiguity from possible mistakes caused by simulator. For example in some types of car-following simulators car can freely pass through vehicles in transverse lanes in crossroad, while in simulations based on cellular automatons this is not possible.

We speak about comparison of traffic control agents, but the same system may be used to compare any kind of traffic control, which is able to use our interface. It is not important, if traffic control algorithms are fulfilling any of conditions given in part 1.

5.1 Experimental software

5.1.1 Simulator

We use Java Urban Traffic Simulator (JUTS) as simulation engine for comparison ([HART]). JUTS is traffic simulator, allowing to model traffic situation at the level of individual vehicles. It is basically a cellular automaton. Traffic network is composed from cells, which can be free or occupied, depending on presence of vehicle. Each cell represents 2.5 m of road. From the top view, simulation is composed from crossroads, connected by roads, divided to lanes. Lanes and crossroads are arrays of cells. Connections between roads and crossroads are provided by access places. Vehicles are able to move trough this network in steps. Simulation is discrete; each step is equivalent of one second in real world.

Vehicles in JUTS are not equipped with the ability to find their path through traffic network. Instead of that, both systems described in part 4.5 may be used. Method based on pseudorandom numbers generators is used by most of vehicles, but if it is necessary, vehicle may also contain exact path through traffic network. This is used mainly for simulation of public transport systems. Because discrete nature of simulation, even speed of vehicles is discretized – they may move from 1 to 6 cells in one simulation step, which corresponds to speeds from 9 km/h to 54 km/h. We are not using larger values of speed, because JUTS is developed as simulator of urban traffic. If necessary, this value can be easily increased.

Lanes are represented as arrays of cells, which can be occupied or empty. Because vehicles can occupy more cells, each lane also contains list of vehicles present in lane, with information which cells are occupied by each vehicle. Road is created as set of lanes in one direction. Typical city street has to be represented by two roads with opposite directions. The lanes in one road don't have to have same length, but they all have to end at the same position. This is used to represent branching of roads in front of the crossroad (see Fig. 19).

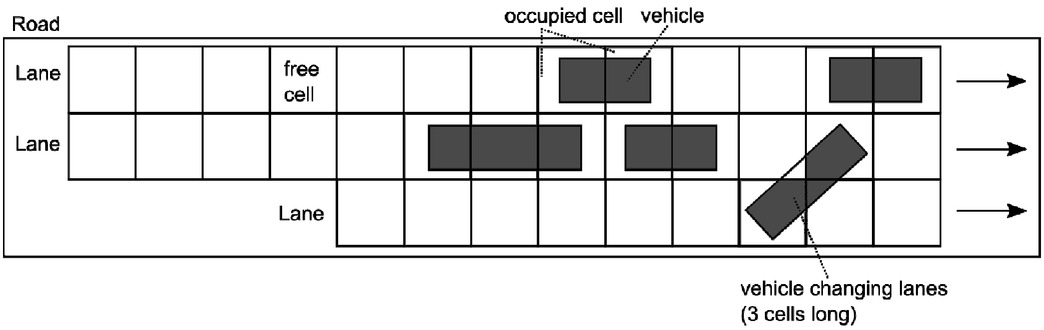

Fig. 19: Road and lanes in JUTS

Crossroad in JUTS is a map segment, created from several cells - crossroad places. Vehicles are moving through crossroad in similar way like through the lane. Vehicles do not have any knowledge of the map; they only know the access place they are heading to. Vehicles are not able to find their way through crossroad, so the ways has to be prepared in map. Way is vector of crossroad cells and it provides to vehicle an information about route to the access point. Each crossroad place can contain list of successors – other crossroad places or access places at the edge of crossroad. Crossroads and traffic lines are connected by access places.

Traffic lights in JUTS are represented by three levels structure. Semaphores (control of the whole crossroad) are at the highest level. One semaphore is at each controlled crossroad. It provides loading and storing of traffic lights, synchronization, and allows access to all traffic lights in crossroad at once. At this level, very simple mechanism of switching time plans is implemented. Because semaphore has no knowledge about used time plan, it cannot work with switching points. All traffic lights are set to "stop" during switch, so vehicles have enough time to leave crossroad. This mechanism can be suppressed, if some different switching mechanism will be implemented in time plan object.

Each semaphore is composed from multiple traffic lights. Traffic light is connected with one access place, which is controlled by it. Access place is placed at the end of one line, so traffic light is able to control one lane. It is possible, that lane is controlled by multiple lights, and each of them is controlling different direction. In that case, multiple traffic lights are associated with appropriate access place. Traffic lights also have connection to lane and crossroad they are controlling, to allow them getting statistical information, if it is required.

Time plan is at the lowest level. Each traffic light has to be associated with some time plan. If multiple traffic lights are in one signal group, they can share one time plan. Time plans are now static, represented by simple finite state automaton. Traffic control agents are used instead of time plans. Agent container (see below) is connected with traffic light and acts as regular time plan.

5.1.2 Implementation of agents

So far, we have implemented only simple reactive agents HIRA, LOPE and KOSU. Our implementation is created only according to information obtained from corresponding articles with the description of agents. We were trying implementation as simple as possible, in order gain all functionality described in article.

HIRA is based on set of 13 rules, which are not changing in time. This agent has no ability of learning, so its implementation was the easiest one. It was created as modification of vehicles actuated signal control, which is basic form of dynamic traffic lights control. The only problem was with measuring of queues. HIRA is using length of queues in upstream lanes as basic criterion for choosing rule which will be used to optimize traffic control. Because no method of queue length estimation is specified in [HIRA], we have decided to use real information about length of queue from simulator (methods of data collecting is described in 5.1.3.2). In real world, this kind of information is not available, length can be only estimated.

First variation of KOSU is using only switching of static time plans, according to traffic situation during previous signal cycle. No method for preparation of signal plans is given in [KOSU], so we are using two types of plans – with even distribution of green signal, and with prioritization of one direction. Usually, we prepare one variety of prioritization for each road in crossroad. KOSU was created in similar way. It also can be seen as modification of vehicle actuated signal control. Second and third varieties of KOSU are modification of VASC, as is described in section 4.4.6.

LOPE is using rule-based system similar to HIRA, with different set of rules and with different observed traffic characteristics. It was implemented as modification of HIRA.

5.1.2.1 Verification and validation of agents

To verify our implementation of agents, we are using tests from corresponding articles. We are trying to run the same test in our simulator and compare the results. Unfortunately, articles gives us only very limited set of data, so the most important method of verification is only review of code of our implementation and its comparison with description in the article. As a part of future work, we want to contact authors of compared agents and ask, if they will be willing to share more detailed information about their agents. The biggest problem of our tests is initial settings of agents. Most of them contain parameters, which has to be set by operator before agent is used. But in articles, there is no clear way how to find out values of these parameters. Now, we are running each agent with several settings of parameters and trying to determine which setting will give the best result. But we cannot tell, if we have really find out optimal values, or values used by authors in original articles. We are now considering possibility of creating of automated system to determine those parameters, as is mentioned in future work.

5.1.3 Comparison system

Comparison system gives JUTS the ability to collaborate with different types of agents, implemented in Java. It allows agents to control traffic lights and to get necessary information about traffic conditions in simulation. At the end of simulation, it is also able to evaluate agent performance according to selected criteria.

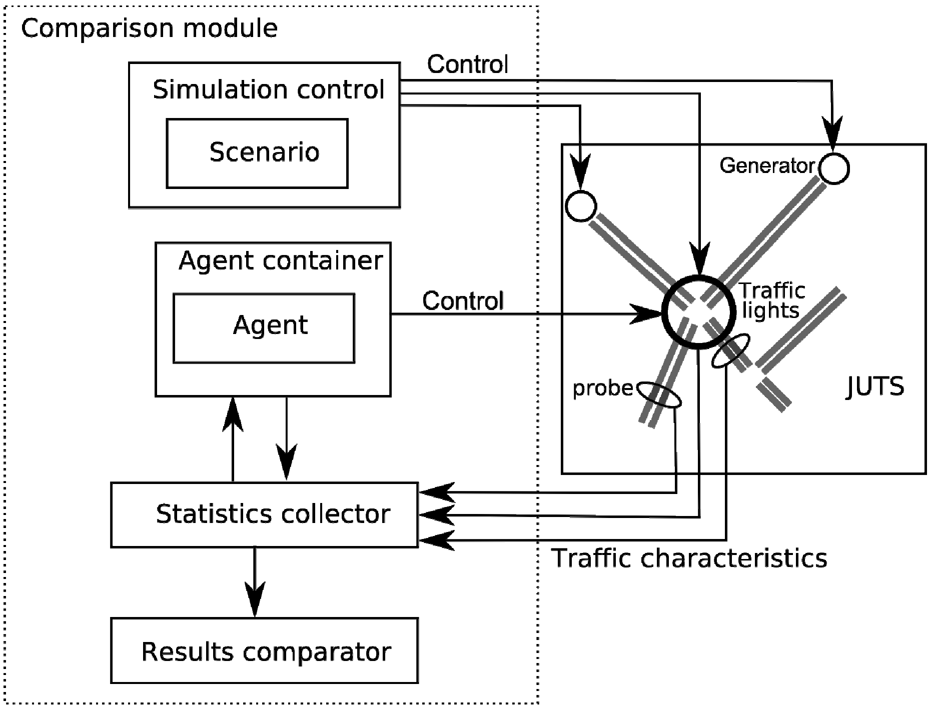

Fig. 20: Scheme of system for agent comparison

5.1.3.1 Agent support

To connect agents with simulation, two types of agent container are implemented in JUTS. First and more important is container for crossroad control agent. It provides access to all traffic lights in crossroad and to information about lanes leading from and to crossroad. Agent may use this container to obtain information about traffic situation in upstream lane. Because some agents (as [HIRA]) needs also information about downstream, container is able to explore possible paths in crossroad and find all downstream lanes belonging to one upstream lane/traffic line. Second type of container is design only to control one traffic lane; the whole crossroad would be then controlled by several collaborating agents. Lane container allows only access to traffic lights of appropriate lane ant to information about upstream and all downstream lanes. In current implementation, we allow using only one type of container – agents may control whole crossroads or lanes, but combination is not possible.

Both containers provide also support for communication among agents. Crossroad container is able to send message to any specific crossroad, to multicast message to all neighbour crossroads or to broadcast message to all crossroads in simulation. Lane container can send message to another specific lane container, multicast to all lane containers in one crossroad or broadcast to all lane containers in simulation. Because distributed version of JUTS is developed, it is important to communicate only trough prepared methods, otherwise agent will not be able work correctly in distributed simulation. Messages are marked by its senders and they may have assigned priority. They are stored in priority queues in the container until they are read by agent (they cannot be lost or overwrite by newer message).

Containers also provide methods to search other agents in the simulation. In JUTS all objects are marked by ID numbers. Because agents are controlling specific objects, they are using their ID's. Thus, agent may be found by ID of controlled crossroad or lane. If this ID is not known, container is able to find ID's of neighbour agents, if they exists.

Implementation of two more containers is now being prepared. It will be road container and group container. In some multiagent control systems, road agents are proposed to observe situation on road and inform crossroad controlling agents about it (this is used in [KWAT] or [MIZU]). Originally, we intended to emulate this only by measurements in appropriate lanes, carried out from crossroads. But for two reasons, we decided to use agent container instead. At first, road agents may be also designed to work with variable speed limit signs, according to their own plans or to request from crossroad agent. And second reason is problem with distribution. Using of agent container will allows us to use existing methods for communication, if measured road and crossroad will be distributed on different nodes.

Group container will be used for managing agents, if they are necessary. In some multiagent traffic control systems (for example in [ROOZ] or [FRAN]), hierarchy of managing agents is used. Container will allow creating and managing group of crossroad agents or other managing agents, if more complex structure is used. It will provide methods to send messages to subordinate agents or to broadcast to whole group. This broadcasting has to be invoked by agent in the group container – if one of subordinate agents needs to broadcast to the whole group, it is allowed only to send message to managing agent and it can decide if this message will be broadcasted or not.

5.1.3.2 Measurements

All measurements are provided by statistics collector. Collector has full access to objects in simulation map and it is able to obtain all kinds of information available in simulation. Lanes and crossroads in JUTS are creating their own statistics, such as long-term average vehicle speed or flow and they may be made accessible to agent containers trough statistics collector. Statistic collector is also able to detect queues and measure its lengths or collect statistics as actual average speed of traffic density. Samples may be taken regularly, according to prepared plan – this is useful to evaluate agents' performance at the end of simulation. Collector is also able to take sample on request from agent container.

Statistics collector may also access to information from vehicles, not only from traffic network. It can measure average time they spent in simulation, by waiting in queues or how long they are waiting in the last queue. If necessary, every vehicle (or only selected vehicles) may be used as measuring vehicles (see section 3.2). These data cannot be easily obtained in real world (at least without using of special hardware in vehicles), but they are used in some proposed control agents ([BALA], [HOAR]).

Collector can provide two kinds of characteristics: it can measure values in lanes (or roads) and provides impartial information about traffic in the lane. This is used mainly for comparison of control systems, because this type of characteristic is not available in real world. Or it can provides a “probe”, which simulates detector placed near (or in) traffic lane. Position of probe can be defined by index of measured cell because lanes are implemented as arrays of cells (see 5.1.1). Probes are used mainly by control systems, because they are providing similar information as detectors in real world. During implementation of agents, we are trying to use characteristics from probes, if possible. The only exception is length of queues. If agent is using length of queues to optimize its work, and there is not specified method of estimation of queue lengths, we are using queue lengths measured in directly in simulation.

So far, statistic collector can provide following characteristic:

- Actual average speed in the lane: speed is calculated as average from actual speeds of all vehicles in selected lane. This value cannot be obtained from sensors in real world, but it is useful for visualization.
- Average speed in lane: Average speed in given number of steps (JUTS is working in discrete time steps, one step matches to one second). Actual average speed is calculated in every step and average is calculated. Values of actual speed are stored, maximal number of stored values corresponds to number of steps in which average speed is computed.
- Average speed in vicinity of probe: Probe is storing speeds of passing vehicles for given number of steps. Average speed is calculated from stored values.
- Queue length in lane: We consider all stopped vehicles (vehicles with zero speed), staying in the lane without any free space amongst them. System is searching for queue from the end of the lane and only the length of first found queue is measured.
- Actual occupancy: True or false value, registered by probe. Probe is registering true value, if cell with probe is occupied.
- Average occupancy: Value calculated by probe from stored values of actual occupancy. As in case of average speed, number of steps is given, for which occupancy should be collected.
- Lane occupancy: Percentage of occupied cells in lane
- Free space: Number of free cells at the beginning of the lane. This value is used to determine, how many vehicles may enter the lane.
- Intensity of traffic in lane: Number of vehicles passing lane in unit of time (the unit of time can be given as number of simulation steps)

- Density of traffic in lane: Density is calculated as average of occupancies in given unit of time. If density in the lane is measured, occupancy is sampled in each step and stored for the given number of time steps.
- Average speed of selected vehicle: This allows agents to work with measuring vehicles. There are no parking places in the simulation; vehicles are spending all their time in simulation by moving. Average speed is computed for the whole time vehicle spent in simulation.
- Average time spent by vehicle in simulation: This value is closely related with average speed of vehicle. It can be used to estimate how long time vehicles spent in queues.

5.1.3.3 Simulation control

Simulation control is used to automatically run prepared scenarios with different agents. Because amenities of dynamic traffic control are apparent mainly when traffic flow is not static, it is necessary to change settings of vehicle generators or probability of switching on crossroads. These changes are described in scenarios. Each scenario contain its length (in time steps, which corresponds to seconds), changes of pseudorandom number generators (which are used in vehicle generators and at crossroads) settings and time when change should apply. Settings of crossroad switching probabilities may be used even to simulate accident, to test if agent is able to deal with sudden vast changes. Vehicles in JUTS are not able to search way in traffic network, so operator has to prepare the whole accident scenario, including paths used by vehicles to avoid accident. Simulation control ensures that changes will be applied after given amount of time. It is also possible to set seed of pseudorandom generators, thus achieve identical scenario.

This module can be configured to execute several runs with different scenarios and agents and to summarize results of each run to xml files.

5.1.3.4 Comparison of agents

To compare agents' performance, results comparator is using data obtained from statistics collector. Compared values and places of their measurement have to be set before the simulation is started. There are three types of criteria used to evaluate traffic control of crossroad in [GUBE] – criteria based on capacity, criteria based on queues and environmental criteria. Because our simulator is not designed to observe fuel consumption or CO, CO_2 and other pollutants dispersion, we focus only on first two. Capacity criteria are observing amount of vehicles passing trough crossroad. There is detail description in [GUBE] how to compute maximal theoretical capacity of crossroad, which can be compared to achieved amount of passing vehicles. Queuing criteria are observing delay of vehicles in crossroad, caused by red signal or by insufficient capacity of crossroad.

Criteria for global evaluation may be based on average values obtained from each crossroad, but it is possible to use other information. The aggregate waiting time of vehicle on all crossroads can be measured. If vehicles are using static paths, it is possible to embrace average time spent by travelling, if random numbers generators are used, the same role is fulfilled by average time spent by vehicle in simulation. It is also possible to focus on situation on roads, not only crossroads. Average speed or traffic density may be used.

It is important to know, that not only selection of criterion, but also selection of places where it will be measured is important. For example on junction of two main roads, agent may try to treat them both as equal or to prefer one of them – which of course leads to deterioration of measured criteria in the other road.

5.2 Simulation experiment

In this part we show method of experiment used to compare performance of two traffic control systems. Each simulation experiment consists of four main parts – map of tested network, scenario, set of agents and their settings and chosen criteria used to compare their performance. Selection of criteria is very important, because there is no general way how to tell if traffic control is optimal or not. For example agents (or traffic control systems) minimizing length of queues may not be optimal if average speed is observed. It is possible to observe more than one criterion, but then operator has to decide which of them is more important.

Traffic network contains full description of roads and crossroads. It cannot be changed during experiment. Even accidents and blocking of traffic lanes has to be done in scenarios. This is because vehicles in JUTS are not able to find their own way trough simulation map, so if the map is changed, prepared paths or turning probabilities has also to be changed. Because in our simulation only vehicles are active parts, it may be complex, number of roads and crossroads has only little influence on speed of computation.

Scenario contains description of vehicles in the simulation. Two most important parts are description of vehicle generators at the edge of simulated network and random numbers generators on crossroads, responsible for turning of vehicles. In contrast to traffic network, these values are changed during experiment, in order to determine ability of agent to adapt to new conditions. Data for scenario can be based on values measured in real world, or they may be prepared by operator. In this case, it is important to try scenario before experiment is performed, to check if it is suitable and it represents required situation (for example, wrong setting of vehicle generators may lead to rapid congestion of traffic network, wrong setting of crossroads may caused accumulation of vehicles in some parts of network). To test all abilities of control agent, several experiments has to be performed, with different scenarios. Agent capable for example of optimal control of uncrowded road may have problems with solving heavy congestions. Because vehicles are active part of the simulation in JUTS, scenario has the biggest influence on the speed of computation. With higher number of vehicles, simulation is getting slower.

Description of agents contains not only algorithm of each agent, but also their position and settings. Even if only one type of agent is used, agents from different crossroads must not be confused. Each agent has its own unique settings, which has to be prepared for crossroad controlled by it. Some types of agents are able to change these settings due to their learning mechanism, but in many cases at least some of setting is static and cannot be changed by agent itself. It may be only set of parameters, as preference of one direction or minimal and maximal duration of green signals, but some agents also requires preparation of signal phases or initial estimates of traffic density in all controlled lanes. If agents are able to learn, they should be run in typical scenario for the area, in order to give them initial state. Unfortunately, it is often, that method how to set similar properties is not described in articles with descriptions of agents. Even very efficient agent may give poor performance, if these settings are incorrect. Settings also cannot be part of scenario, because agents are supposed to act autonomously, without outer control. Another important thing is hierarchy of agents. If hierarchical structure is used (for example ROOZ or FRAN), it also has to be part of description of the experiment. Even with the same setting of each crossroad-controlling agent, results may be different if they are grouped differently. It is caused by managing agents. They are helping to find global optimum within the group, so in different groups agents will be working on search of global optimum for different parts of the network.

The last part of description of experiment is selection of criterions and position of their measurement, if necessary. It is possible to observe and compare any amount of places in the network, as well as global statistics. Selected criteria have no effected on the work of agents, they are used only for their comparison.

Evaluation of experiment is basically evaluation of hypothesis that according to chosen criterion in given network and scenario, optimal control is achieved by one of used traffic control agents. Because this hypothesis is valid only for the whole experiment, it is important to perform several experiments to determine quality of control of one agent.

5.2.1 Typical scenarios

In this part, several basic scenarios and the way how to create them is given. These scenarios may be used to compare performance of all described traffic agents.

The most basic scenario is steady traffic flow with static density. It shows ability of agent to ensure traffic fluency or priority of important roads. No changes are done during experiment, but several experiments with different densities should be performed to see capacity limits of agents and traffic network. In this case, even with very simple agents, traffic congestions are caused more by maximal capacity of the network, than by insufficient ability of agent to control network.

Similar experiment is dealing with fluctuating traffic flow. During experiment, density of traffic flow raises and drops several times. Typical example from real life is control of roads leading from residential areas to the city and back – during morning, there is a lot of traffic to the city and only few cars are going in opposite direction. During evening, situation is inverted; heavy traffic is in the direction from the city. This scenario shows ability of agent to adapt to different situations and to control traffic for the whole day. The period with unchanged traffic flow should be long enough in order to simulate real situation.

Ability of agent to solve congestions may be tested by heavy traffic flow, causing congestion at the beginning of the experiment. It may be created only on main road or on crossing of two main roads and propagates itself to the whole network. Some agents, as LOGI are created specifically to solve congestions.

The last important scenario is reaction of agent to accident or any other type of roadblock. In this scenario, vehicles generators are working in the same manner as in normal situation, but turning probabilities or paths has to be changed. During preparation, it is necessary to decide where detour will be and change turning probabilities in all affected crossroads, not only on the one with accident. This shows ability of agent to react on rapid change of situation in controlled crossroad.

5.3 Examples of experiments

The experiments described below were performed above all to show, that our system is working and is capable to provide required functionality. Much bigger amount of experiments will be necessary to make confidential comparison of agents mentioned above. These experiments were performed mainly to check, if the comparison system is working. They should not be used to judge the abilities of used traffic agents.

5.3.1 Comparison of VASC and HIRA

This is very simple example of comparison of two traffic control agents, in small map. Map of used traffic network is on Fig. 17, left panel shows screenshot from the simulator and right shows logical structure of the map. This experiment is example of experiment for verification of agent – it is recreating experiment from [HIRA].

Agent HIRA and vehicle actuated signal control (VASC) were tested in simple map, shown at Fig. 17. Traffic lights were placed at each crossroad; results were taken only from highlighted crossroad at bottom right. Directions, where queues were measured are denoted by letters *a*…*f*.

Two scenarios were created. In both, the densest traffic was simulated at the main road at right side. In the first scenario, three traffic peaks were created, in the second one, six peaks were simulated. Duration of each peak and gap between them was set to 10 minutes (600 simulation steps). During each peak, flow of vehicles was two times higher than in normal traffic.

In initial setting of VASC the main road was prioritized by getting longer duration of green phases. Minimal time duration for directions *a, b, d* and *e* was set to 40 seconds, maximal to 60 seconds. In directions *c* and *f* minimal duration was 20 seconds and maximal 40 seconds. Duration of green signal was increased in 2 seconds steps. A three phase control is used (it is not possible to go to the left from directions *c* and *f*).

In HIRA similar prioritization was made. All patterns containing directions *a, b, d* and *e* has minimal green duration set to 40 seconds and maximal to 60 seconds. Other patterns use 20 seconds as minimal duration and 40 seconds as maximal. Increasing step is again set to 2 seconds. Patterns 6, 8 and 11 are not used, because it is not possible to turn to left from *c* and *f* direction.

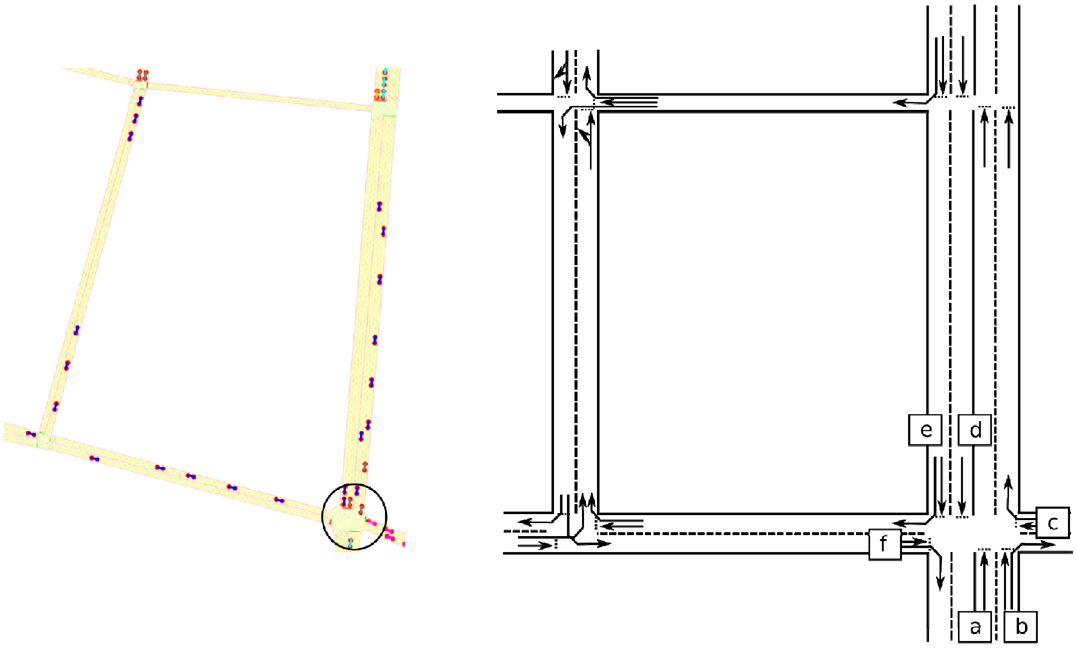

Fig. 21: Screenshot from simulator and scheme of tested traffic network

Average length of queues in controlled crossroad was chosen as criterion for comparison. It was sampled at the end of red signal in each lane. When scenario is completed, average length of queue is calculated for each lane in the observed crossroad.

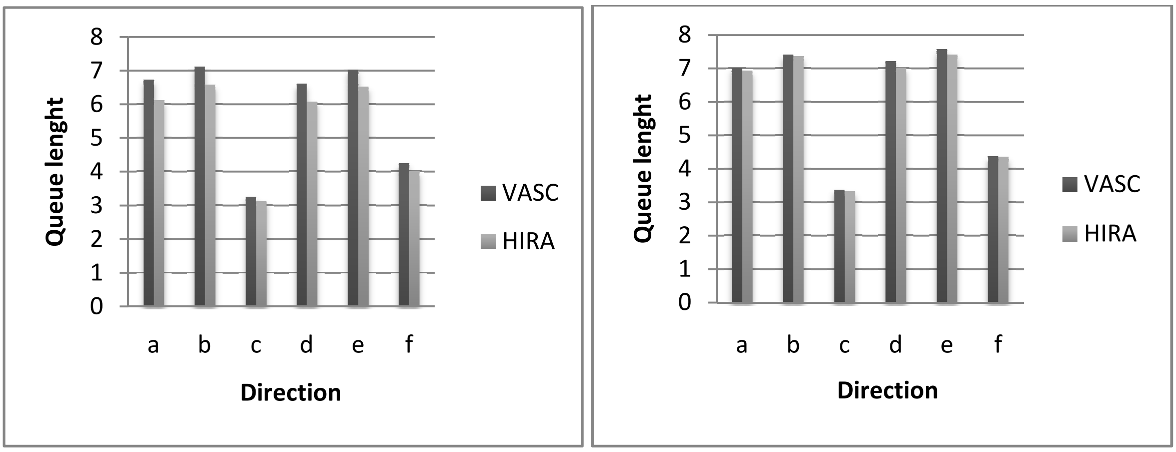

Fig. 22: Comparison of VASC and HIRA for first (left) and second (right) scenario

Results are shown at Fig. 22. Left bar chart shows results from the first scenario, right bar chart stands for second scenario. We may see that HIRA agent control leads to better results, especially in the first scenario. Results of both agents are becoming more similar with higher traffic density, which may be caused by approaching to maximum road capacity. Better result of HIRA in first scenario is caused by better utilization of time. HIRA is capable to use "friendly directions" (see section 4.4.4), so if it is possible, more vehicles can pass crossroad during one signal phase. These results are in accordance with observations in [HIRA]. More importantly, it was shown that comparison module is able to run tests with different agents and generate demanded output.

5.3.2 Comparison of queues length for HIRA, KOSU 1 and LOPE 1

This experiment is similar to the first one. It shows how test of performance of different traffic control agents may look like. Length of queues is used again as a criterion.

We have created larger map, where all crossroads were controlled by agents. It can be seen at Fig. 23. There are two main roads, crossed in the middle. Main roads have two lanes in each direction, and they are prioritized by settings of all agents. Other roads are having only one lane in each direction. Vehicles are injected to all lanes at the edge of map, Gen 1 – 4 are highlighted only because they are used other experiment (5.3.3).

Intensity of traffic on horizontal main roads is three times higher, than on other roads, intensity on vertical main road is 1.5 times higher. We have used only one scenario, with three traffic peaks in vertical lane. Each peak lasts for 5 minutes (300 simulation steps), 5 minutes periods of calmer traffic are amongst them. During peak, intensity traffic in vertical lane is the same as in horizontal lane.

Turning probabilities are set in the way, that around 60 percent of vehicles is continuing straight, 30 percent is turning to right and 10 percent is turning to left on

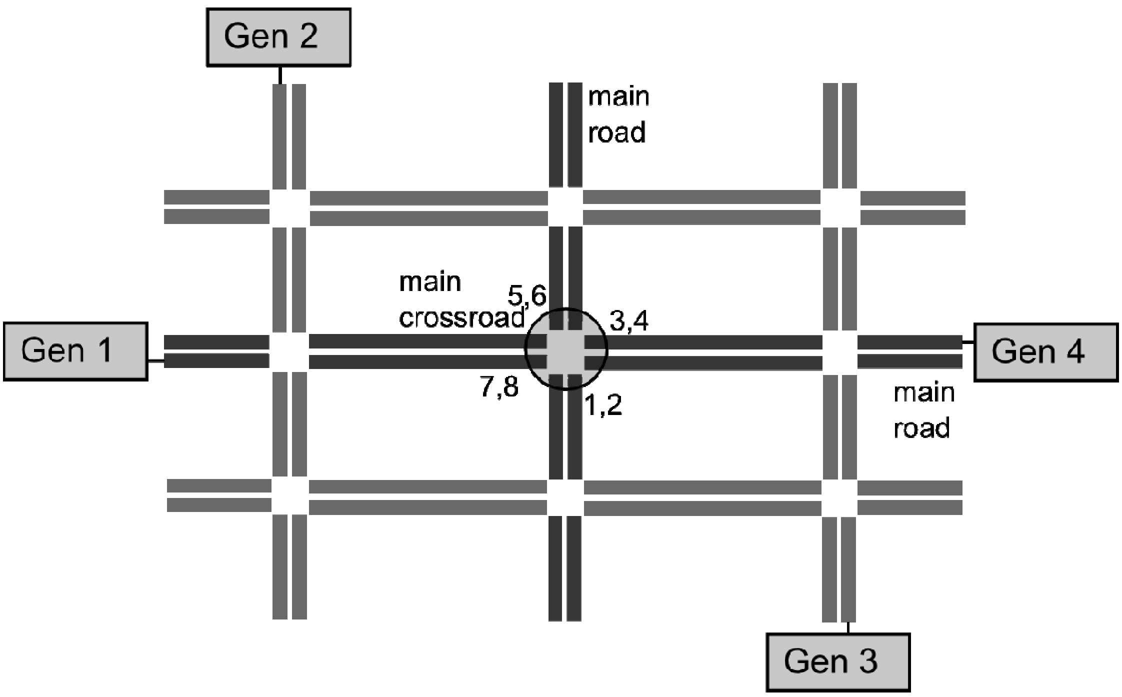

Fig. 23: Experimental map

each crossroad. Random number generators used for injecting vehicles to the simulation and to determine turning of vehicles on crossroads were set at the initial values for all agents, to ensure the same conditions.

HIRA agent on main crossroad is set to provide from 40 to 60 seconds of green signal for lanes of the vertical road and from 30 to 50 seconds of green signal for lanes of the vertical road. On crossings of the main and side roads, from 40 to 60 seconds of green light is provided for the main road and from 20 to 40 for side roads. On crossing of side roads, from 30 to 50 seconds of green light is provided for both roads symmetrically. LOPE 1 is sharing this setting (it is using similar system of prepared signal phases, but different set of rules for their selection and it measures different values).

KOSU 1 is choosing from the prepared time plans. On the main crossroad, it has three time plans – one with 50 seconds of green light for the vertical road and 40 second for the horizontal road, one with 45 seconds of green light for both roads and one with 60 seconds of green for the horizontal and 30 for the vertical lane. On crossing of the main and side road, three plans are also prepared – one with 40 seconds of green for the main road and 40 for the side road, one with 60 seconds of green for the main and 20 seconds of green for the side road and one with 40 seconds for both roads. Plans for crossing of two side roads are also three, one with 30 seconds of green for both roads, one with 20 seconds of green for the horizontal and 30 seconds for the vertical road and one inverse, with 30 seconds for the horizontal and 20 seconds for the vertical road.

Results of experiment can be seen at Fig. 24. HIRA agent is producing shortest queues, so it proves it is the most effective according to this criterion. Results achieved by KOSU are very similar (average difference between HIRA and KOSU is 0.16), results of LOPE are worst (average difference between HIRA and LOPE is 0.35, between KOSU and LOPE 0.21).

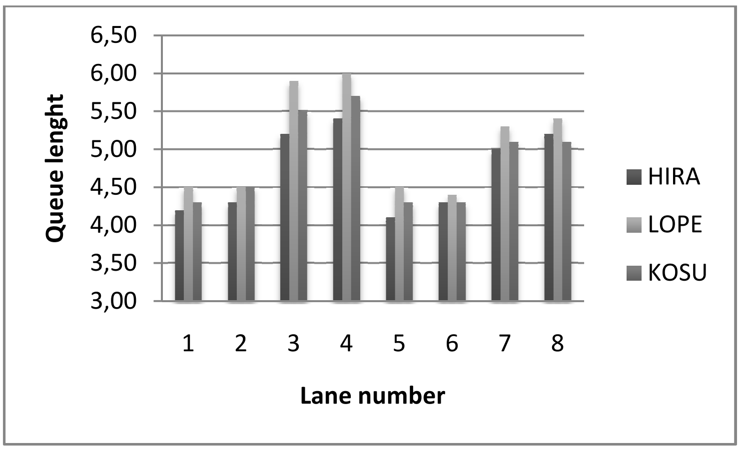

Fig. 24: Queue lenghts

5.3.3 Comparison of average time spent in simulation for HIRA, KOSU 1 and LOPE 1

In this experiment, we have used the same map and agent settings as in previous one. Instead of measuring length of queues, average time spent by vehicles in simulation is measured. This experiment shows comparison of agents' performance from vehicles' point of view, not from global values. Vehicles are marked by number of generator they came from (see Fig. 23), and divided into four categories according to their place of origin. We used the same scenario as in the previous experiment, with three periods of calm traffic and three traffic peaks. After the last peak, simulation was terminated. Vehicles, which have not been terminated at that moment weren't used in statistics.

A result of this experiment shows again (Fig. 25) that the most effective agent is HIRA – average time spent in the simulation is the lowest, so vehicles passed through crossroads with shortest delays in queues. KOSO is again second, average difference between HIRA and KOSU is 10 steps (or seconds). Average difference between HIRA and LOPE is even 20 steps, which means that vehicles in the map had to spend another 20 seconds by waiting in queues.

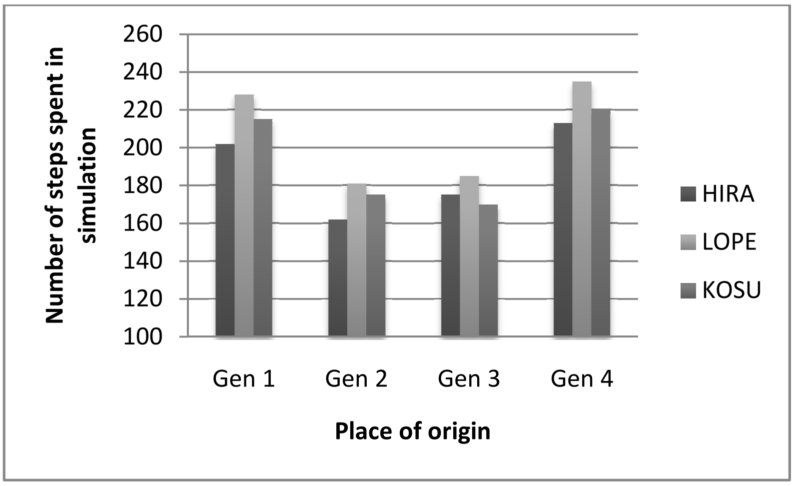

Fig. 25: Time in simulation

6 Future work

So far, we have implemented comparison system and three simple traffic control agents. The next step will be to implement more agents, at least one representation of each group described in section 4.5. Then, we want to perform a number of experiments, to determine abilities and weak points if implemented agents. In next work, we want to focus on two main fields. The first one is a multi-objective comparison and optimization of traffic control. The second one is using of proposed system to evolution optimization of parameters of traffic control agents.

6.1 Multi-objective comparison

One of problems in traffic control is decision what should be optimized. There is lot of criteria, and no clear way how to tell which is more important than the others. Designers of traffic networks and control systems may try to minimize length of queues or amount of time spend in congestions, or to maximize amount of vehicles passing through network during one hour, fluency of car movement or average speed of vehicle (other used criteria may be seen [GUBE]). We want to create automated system, which will be capable to choose the best fitting traffic control agent to optimization criterion selected by operator. Tests will be performed for all available agents. We are now considering creation of artificial criterion, composed of multiple measured values, in order to make comparison simpler.

The final system we want to create will consist from database of traffic control agents, maps and prepared sets of scenarios for each map. When map and scenario will be selected, system will run a set of experiments with available agents. At the end, comparison of agents will be shown, according to selected criterion or criteria. This can be used as decision supporting system for operator responsible of settings of real world traffic lights.

During comparison, we want to find answer for two main questions: how big difference is between results of reactive and deliberative agents and how important is direct collaboration of agents. The test will be focused on comparison of typical representatives of groups described in section 4.5. We consider the question of communication amongst agent or using central point to optimize time plans of several crossroads to be very important. Using of this technology on real crossroads would require installation of additional hardware to provide communication amongst crossroad controllers. It is important to determine, if such expenses will bring significant improvement of traffic control performance. This question can be answered by simulation.

6.2 Evolution optimization

With ability to choose better traffic control, we hope to design system for evolution optimization of traffic control agents. Agents are able to change their inner state according to traffic situation, but often they contents also set of constant parameters, which has to be set by operator in advance. It may be for example length of used time cycle, minimal and maximal duration of green signal, thresholds in fuzzy logic or numeric parameters in internally used equations. Unfortunately, the method how to set these parameters is often not given. With evolution approach, it may be possible to determine value of these parameters automatically, without need of experienced operator.

Evolution optimization of control of complex traffic network will be very hardware-demanding (complexity of task is exponential, considering number of controlled crossroads). Because of that, we will probably need to create simpler simulator (possibly even a macrosimulator) to choose the most promising agent generations and test only them on microsimulator. We are also considering using of distributed variety of JUTS, which is now being developed.

7 References

[BALA] BALAN, Gabriel; LUKE, Sean. History-based Traffic Control. Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS'06). 2006, p. 616-621. ISBN: 1-59593-303-4

[BALM] BALMER, Michael; CETIN, Nurhan; NAGEL, Kai; RANEY, Bryan. Towards truly agent-based traffic and mobility simulations. *Proceedings to 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'2004)*. 2004, p. 64-67. ISBN: 1-58113-864-4

[BRAC] BRACKSTONE, Mark; McDONALD Mike. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*. 1999, vol. 2, no. 4, p. 181-196. ISSN 1369-8478

[BRAC2] BRACKSTONE, Mark; McDONALD Mike. Driver Behaviour and Traffic Modelling. Are we Looking at the Right Issues? *Proceedings of the IEEE Intelligent Vehicles Symposium*. 2003, p. 517-521. ISBN: 0-7803-7848-2/03

[BROO] BROOKS, Rodney A. Intelligence without representation. *Artificial Intelligence*. 1991, vol. 47, no. 1-3, p. 139 – 159. ISSN: 0004-3702

[CHAN] CHANDLER, Robert E.; HERMAN, Robert; MONTROLL, Elliot W. Traffic dynamics: Studies in car following. *Operational Research*. 1958, vol. 6, no. 2, p. 165-184. ISSN 0030-364X

[COME] COMERT, Gurcan; CETIN, Mecit. Queue length estimation from probe vehicle location and the impact of sample size. *European Journal of Operational Research*. 2009, vol. 197, issue 1, p. 196-202. ISSN: 0377-2217

[DONI] DONIEC, Arnaud; ESPIÉ, Stéphane; MANDIAU, René; PIECHOWIAK, Sylvain. Non-normative behaviour in multi-agent systems: some experiments in traffic simulation. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06)*. 2006, p. 30-36. ISBN: 0-7695-2748-5/06

[DRES] DRESNER, Kurt; STONE, Peter. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. *Proceedings to 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'2004)*. 2004, p. 530-537. ISBN: 1-58113-864-4

[FERR] FERREIRA, Enrique D.; SUBRAHMANIAN, Eswaran; MANSTETTEN, Dietrich. Intelligent agents in decentralized traffic control. *2001 IEEE Intelligent Transportation Systems Conference Proceedings*. 2001, p. 705-709. ISBN: 0-7803-7194-1/01

[FRAN] FRANCE, John; GHORBANI, Ali A. A Multiagent System for Optimization Urban Traffic. *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03).* 2003, p. 411-414. ISBN 0-7695-1931-8

[GART] GARTNER, Nathan H.; POORAN, Farhad J.; ANDREWS, Christina M. Optimized Policies for Adaptive Control Strategy in Real-Time Traffic Adaptive Control Systems:

Implementation and Field Testing. *Transportation Research Record: Journal of the Transportation Research Board.* 2002, vol. 1811, p. 148-156. ISSN: 0361-1981

[GIBS] GIBSON, David; MILTON, Mils; KLEIN, Lawrence. A new look at sensors: FHWA's updated Traffic Detector Handbook describes in-roadway and over-roadway technologies for vehicle detection as key to ITS deployment. Public Roads. 2007, no. 11-12. ISSN: 0033-3735. URL: http://findarticles.com/p/articles/mi_m3724/is_3_71/ai_n25127927 (4.4.2009)

[GROS] GROSS, Dave; GROSS, Jon. Emergency vehicle traffic signal pre-emption and collision avoidance system. 2001. United States patent 6,326,903.

[GUBE] GUBERINIĆ, Slobodan; ŠENBORN, Gordana; LAZIĆ Bratislav. Optimal Traffic Control. *New York: CRC Press*, 2008. ISBN: 1-4200-6268-9

[HART] HARTMAN, David. Modelování dopravní situace systémy hromadné obsluhy (Traffic simulation Using Bulk Systems). *University of West Bohemia, diploma thesis*. 2003

[HIRA] HIRANKITTI, Visit; KROHKAEW, Jaturapith. An Agent Approach for Intelligent Traffic-Light Control. *Proceedings of the First Asia International Conference on Modelling and Simulation* (AMS'07). 2007, p. 496-501. ISBN: 0-7695-2845-7

[HOAR] HOAR, Ricardo; PENNER, Joanne; JACOB Christian. Evolutionary Swarm Traffic: If Ant Roads Had Traffic Lights. *Proceedings of the Evolutionary Computation on 2002 (CEC '02)*. 2002, p. 1910-1915. ISBN: 0-7803-7285-4

[HOEK] HOEK, Wiebe van der; JAMROGA, Wojciech; WOOLDRIDGE, Michael. A Logic for Strategic Reasoning. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS'05)*. 2005, p. 157-164. ISBN: 1-59593-093-0

[JADE] Jadex – BDI Agent System. University of Hambug, Distributed Systems and Information Systems Group. URL: http://vsis-www.informatik.uni-hamburg.de/projects/jadex/ (5.1.2009)

[JAMI] JAMISON, S; McCARTNEY, M. Discrete vs. continuous time implementation of a ring car following model in which overtaking is allowed. Nonlinear analysis: real world applications. 2009, vol. 10, no. 1, p. 437-448. ISSN: 1468-1218

[KATW] KATWIJK, Ronald van; KONINGSBRUGGEN, Paul van. Coordination of traffic management instruments using agent technology. *Transportation Research Part C: Emerging Technologies*. 2002, vol. 10, no. 5-6, p. 455-471. ISSN: 0968-090X

[KAVI] KAVIČKA, Antonín; KLIMO, Valenta; ADAMKO, Norbert. Agent oriented simulation of traffic nodes (Agentovo orientovaná simulácia dopravných uzlov). Žilina: EDIS – vydavateľstvo Žilinské university, 2005. ISBN: 80-8070-477-5

[KOSU] KOSUKE, Ono; HATA, Takashi; MAETANI, Toyfumi; HARAO, Masateru; HIRATA, Kouichi. Development of a Multi-Agent BASED Generic Traffic Simulator. *Lecture notes in computer science 4012*. 2006, p 249-260. ISSN: 0302-9743

[KRAT] KRATOCHVÍLOVÁ, Jitka, NAGY, Ivan. Model dopravní mikrooblasti (Model of traffic microarea). Automatizace. 2004, vol. 47, no. 12, p. 752 – 767. ISSN: 0005-125X

[KWAS] KWASNICKA, Halina; STANEK, Michal. Genetic Approach to Optimize Traffic Flow by Timing Plan Manipulation. *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*. 2006, p. 1171-1176. ISBN: 0-7695-2528

[LAWA] LAW, Averill M; KELTON, David W. Simulation Modeling and Analysis. New York: McGraw-Hill, 1991. ISBN: 0070366985

[LIGH] LIGHTHILL, M. J.; WHITHAM, G.B.; On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*. 1955, vol. 229, no. 1178, p. 317-345. ISSN 1364-5021

[LOGI] LOGI, Filippo; RITCHIE, Stephen G. A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transportation Research Part C: Emerging technologies*, vol. 10, no. 5. 2002, p. 5007-527. ISSN: 0968-090X

[LOPE] LÓPEZ, José Alejandro; GARCÍA, Rafael; BLANCO, Alejandro García; ZÚÑIGA, Ismael A. Félix. Traffic Lights Fuzzy Control Proposals to Improve Vehicular Flow. *Proceedings of Fourth Congress of Electronics, Robotics and Automotive Mechanics*. 2007, p. 406-411. ISBN: 978-0-7695-2974-5

[MING] MINGZHE, Liu; RUILI, Wang; JIANSHENG, Wu; KEMP Ray. A genetic-algorithm based neural network approach for short-term traffic flow forecasting. *Lecture Notes in Computer Science 3498*. 2005. ISSN: 0302-9743

[MIZU] MIZUNO, Kazunori; FUKUI, Yukio; NISHIHARA, Seiichi. Urban Traffic Signal Control Based on Distributed Constraint Satisfaction. *Proceedings of the 41st Hawaii International Conference on Systems Sciences*. 2008, p. 65. ISBN ~ ISSN: 1530-1605, 0-7695-3075-8

[MONA] MONAHAN, Torin. "War rooms" of the street: Surveillance practices in transportation control centers. *The Communication Review*. 2007, vol. 10, no. 4, p. 367 – 389. ISSN: 1547-7487. URL: torinmonahan.com/papers/war_rooms.pdf (5. 4. 2009)

[NAGE] NAGEL, Kai; SCHRECKENBERG, Michael. A cellular automaton model for freeway traffic. *Journal de Physique I*. 1992, vol. 2, no. 12, p. 2221-2229. ISSN: 1155-4304

[NAGI] NAGI, Khaled. Overview of Agent Technology. *Lecture Notes in Computer Science 2249*. 2001, p. 21-39. ISSN: 0302-9743

[NWAN] NWANA, Hyacinth S. Software Agents: An Overview. *Knowledge Engineering Review*. 1996, vol. 11, no. 3, p. 1-40. URL: http://agents.umbc.edu/introduction/ao/ (15. 12. 2008). ISSN: 0269-8889

[OSSO] OSSOWSKI, Sascha; Hernández, Josefa Z.; BELMONTE, María-Victoria; FERNÁNDEZ, Alberto; SERRANO, Ana García; PÉREZ-DE-LA-CRUZ, José-Luís; SERRANO, Juan-Manuel; TRIGUERO, Francisco. Decision support for traffic management based on organisational and communicative multiagent abstraction. *Transportation research. Part C, Emerging technologies*. 2005, vol. 13, no. 4, p. 272-298. ISSN: 0968-090X

[PLUN] PLUNKETT. Traffic Lights. *Phrack*. 2002, issue 60. ISSN 1068-1035 URL: http://www.phrack.org/issues.html?issue=60&id=14&mode=txt (5.4. 2009)

[RONG] RONGHUA, Du. Urban Traffic Coordination Control System Based on Multi-Agent-Game. *Proceedings of International Conference on Intelligent Computation Technology and Automation (ICICTA'08)*. 2008, p. 817-821. ISBN: 978-988-98671-0-2

[ROOZ] ROOZEMOND, Danko A. Using intelligent agents for pro-active, real-time urban intersection control. *European Journal of Operational Research 131*. 2001, p.293-301. ISSN: 0377-2217

[RUSS] RUSSEL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach (2nd Edition). *New Jersey: Prentice Hall*, 2002. ISBN: 0137903952

[SANC] SÁNCHEZ, Jesús Héctor Domínguez; AGUIRRE, José Luis. Traffic light control through agent-based coordination. *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*. 2007, p. 145-150. ISBN: 978-0-88986-629-4

[SHAN] SHANG, Pengjian; LI, Xuewei; KAMAE, Santi. Chaotic analysis of traffic time series. *Chaos, Solitons and Fractals*. 2005, vol. 25, no. 1, p. 121-128. ISSN: 0960-0779

[TAAL] Taale, H. Comparing methods to Optimise Vehicle Actuated Signal Control, Proceedings of Eleventh International Conference on Road Transport Information and Control. 2002, p. 114-119. ISSN: 0537-9989, ISBN: 0-85296-746-2

[WOLF] WOLFRAM, Stephen. A New Kind of Science. *Wolfram Media Inc*, 2002. ISBN: 1-57955-008-8

[WOOL] WOOLDRIDGE, M.; Jennings, N. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*. 1995, vol. 2, no. 10, p, 115-152. ISSN: 0269-8889