REAL-TIME INFORMATION AND CONTROL SYSTEM FOR URBAN VEHICLE TRAFFIC

Tiberiu Leția Honoriu Vălean Adina Aştilean Mihai Hulea

Dept. of Automation, Technical University of Cluj-Napoca,
15 C.Daicoviciu St., Cluj-Napoca, Romania. e-mail:
{ Tiberiu.Letia, Honoriu.Valean, Adina.Astilean, Mihai.Hulea}@aut.utcluj.ro

Abstract: *The solutions of urban vehicle traffic problems imply the description of the system, the evaluation of its behaviour, the control algorithm design and implementation. For better results, the traffic control system must be enhanced with an advisor, able to communicate via mobile phones with the drivers, for providing best route information and acting as a primary traffic formatter. In the paper a real-time control and information system for urban traffic is presented. The control part uses a two levels structure connecting controllers and supervisors. The controllers implement extended state machine. The supervisors implement an algorithm that negotiates the granted rates of the car flows using their prices. The goal of the research is to obtain a control system that is able to adapt to the significant changing of the vehicle traffic characteristics, to increase the system throughput and to avoid or delay congestions. The information part uses different routing algorithms and provides best route information for the drivers.*

Keywords: *Discrete event dynamic systems, Distributed control systems, Traffic control, Routing algorithms, Multi-objective optimisation*

1. INTRODUCTION

Vehicle traffic control becomes important with the increase of the number of participants (i.e. cars). Vehicle traffic control is divided into urban traffic control and road traffic control [5], [11]. The difficulties in solving the problems of traffic routing and control are given by the continuous (and significant) variation of the characteristics and parameters of the traffic. These depend on the period of the day and on the days (working or holiday). So, it is difficult to find one model that describes completely the entire system during a long period of time. Instead of this, a set of models that depends on the time can be used.

The continuous modification of the characteristics and parameters of the traffic implies the change of the control algorithms with those that are most appropriate for the current situation.

The activities involved by this research include the design of the information system, the traffic control system and the traffic surveillance system for a medium size town. This implies:

the congestion prediction and avoidance, the control of the traffic lights, the best route advice, the dynamic route guidance, the estimation of the arrival times etc.

To get these aims, the following were achieved:

- the models of crossroads and of the traffic system [12];
- the adaptation of the models to continuous system changes [13];
- the evaluation of the performances of the control algorithms [7] to be able to choose the best algorithm
- the best control algorithms according to different situations.

Besides the control of the colour of the traffic lights that determines the system behaviour, route planners give detailed instructions on the sequence of roads to be followed in order to reach a particular destination. The information may be presented via mobile data services and may include:

- advisory related to the best route taking into account the minimum time, minimum distance or a combination criteria of these two;
- the expected arrival time at destination;
- warnings about current jamming etc.

For solving these problems, a hierarchical architecture is proposed. The R-T Information and Control System general structure [14], is presented in Fig. 1.

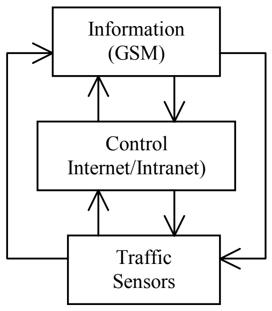


Fig. 1. The general structure of RTICS

2. THE R-T CONTROL SYSTEM

2.1 *State of the art*

The traffic signal control is often considered as being a part of the intelligent transportation system. The U.S. Department of Transportation considers that the traffic signal control systems control signal timing at individual signal controllers to coordinate the traffic flow. In the most advanced systems, traffic flow information is used as input data by algorithms in traffic control programs which automatically adjust signal timing plans in response to current traffic demand [26].

In [4] is proposed a solution for solving the problem of vehicle traffic control using timed Petri nets to model the urban areas and an optimiser module to improve the overall performance. Sensor signals about vehicle arrivals and departures plus intermediate information on the controlled urban area are considered as input events for an *observer* (implemented by the timed Petri net) subsystem. The optimiser uses the information furnished by observers to determine the references of the local controllers. The architectures of the vehicle traffic control are usually decomposed on two or more levels. The upper level is implied by the necessity to solve the conflicts between the controllers from the lower level. The hierarchical structure solves the coordination problem between neighbour intersections, but fails on the general purpose of the system. The urban vehicle traffic is a large-scale system. So, one of the most difficult problems to be solved at the upper level is the consistency of the global data [2] or the state obtained by supervisor. The controllers from the low level could implement different types of algorithms [13].

2.2 *The modelling of the urban traffic*

Many types of models of vehicle traffic have been constructed. Some of them concern with the one-dimensional vehicle flow on a road without intersections. Other models extend the previous on a two-dimensional space taking into account the effect of intersections, traffic lights, coupling the traffic in different directions, turning cars, etc. Some of the models predict the traffic jams [16] when a critical density is reached. Another use of the models is for the congestions prediction. Modelling and simulation play an important role in optimisation of traffic flow [10].

Other models of traffic systems are based on the assumption of stationarity. This assumption has been acceptable for many applications like city planning. It does not allow simulating certain characteristics of traffic systems such as heavily congested urban road networks. Phenomena such as the formation and dispersion of queues are relevant and cannot be reproduced by static models. Some research efforts in traffic modelling have been focused on the modelling of dynamic behaviour of the several components making up a traffic system. Micro-simulation models [22] have usually been used in research applications and less in planning applications due to the increase of computer power requirements.

In [10] are presented three types of mathematical models used in vehicle traffic simulation. They are based on: ordinary dynamic equations, fluid dynamic equations and equations of kinetic type.

The microscopic models or car models model the response of individual vehicles to their predecessor by ordinary differential equations based on Newton law. Macroscopic models are based on fluid dynamic equations.

Kinetic models could be considered as being situated between the previous two [3]. They are based on Boltzman type kinetic equations. Compared to microscopic models, the computation time is strongly decreased. Such models treat the acceleration of the vehicles by means of heuristic relaxation term or by kinetic description. The macroscopic models use distribution function to describe the number of vehicles with certain location and speed at a time.

Cellular automata models [22], [16] for the traffic flow are derived in analogy to lattice gas automata in gas dynamics. This type of models simplifies the behaviour of the drivers extremely. The vehicles are described by discrete lattice points and discrete speed.

Tadaki et al. [21] model traffic flow using a coupled map. Each driver controls the acceleration of his car under the stimuli of the headway distance or relative velocity to the preceding car. In traditional car-following models, the delay in response to the traffic stimuli is introduced by hand. The delay of acceleration will be naturally introduced, if the motions of the cars are described by second order differential equations of the position of cars. Instead of using differential equations, a discrete-time model is used, so that this approach is highly suitable for computer simulations. For making the model applicable to several realistic traffic situations, open boundary systems and multi-lane roads are transformed to a coupled-map form by time discretisation.

Dia [6] presents a *neural network model* for freeway incident detection. In order to train the neural network to perform incident detection samples of input detector data (speed, flow and occupancy) and output states for both incident and incident-free conditions should be used. Therefore, the data required should at least have a description of the state of traffic along the freeway in addition to detector data comprising traffic flow measurements at regular time intervals for each detector station.

The *discrete event models* represent the queues of the cars waiting to cross the intersections, the traffic light events and the events produced by cars.

To solve the traffic control problem [20] propose the *hierarchical optimization* for linear systems with quadratic cost function. Such method solves the problem only when the system is in the normal parameters of the traffic and it ignores the variable times to collect the date (i.e. to build the state of the controlled system). More recent is the adaptive solution of the traffic control [19].

2.3 *Traffic models*

The street traffic models are based on the following elements: crossroads and streets. The core of these models is the crossroad.

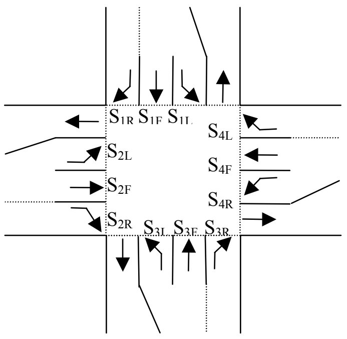


Fig. 2. The crossroad of two streets

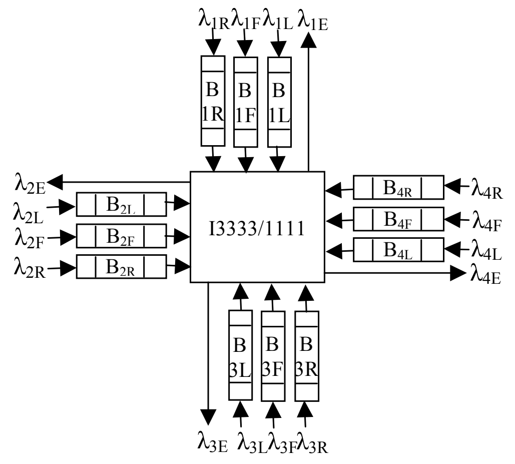


Fig. 3. Intersection 3333/1111 model

There are many types of crossroads .

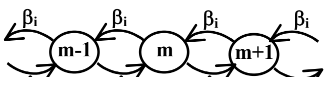


Figure 4. Markov chain

The corresponding relation is given by:

$$(\lambda_i + \beta_i) * P_i(m) = \beta_i * P_i(m+1) + \lambda_i * P_i(m-1) \quad (1)$$

where $P_i(m)$ expresses the probability that in the buffer are m cars

$$\lambda_i * P_i(0) = \beta_i P_i \quad (2)$$

For m=1, results

$$P_i(1) = \frac{\lambda_i}{\beta_i} * P_i(0) \quad (3)$$

Denoting $q_i = \frac{\lambda_i}{\beta_i}$, the general relation is

$$P_i(m+1) = (1+q_i) * P_i(m) - q_i * P_i(m-1) \quad (4)$$

From (4) the following relation is obtained

$$P_i(m+1) = q_i^m * P_i(0) \quad (5)$$

From (5), taking into account that

$$\sum_{m=0}^{\infty} P_i(m) = 1$$

results

$$P_i(0) * \sum_{m=0}^{\infty} q_i^m = 1$$

The condition to avoid the accumulation of the cars is q_i<1 or $\lambda_i < \beta_i$.

If $\lambda_i > \beta_i$ during a period the accumulating occurs that could lead to overflow.

From (5) (for q_i<1) the following relations can be obtained:

$$P_i(0) * \sum_{m=0}^{\infty} q_i^m = P_i(0) * \frac{1}{1-q_i}$$
$$P_i(0) * \frac{1}{1-q_i} = 1 \text{ result } P_i(0) = 1 - q_i \quad (6)$$

Such model can be used to determine the maximum green colour duration without overflowing the next crossroad capacities.

The traffic model includes intersections that have inputs and outputs denoted by the following rules:

- each character represents the number of buffers on the input of that direction;
- after the slash each number represents the number of the outputs of that direction.

For example, 3323/1123 means the following:
- there is a crossroad of 4 streets;
- the first and the second street have 3 input buffers and one output;
- the third has 2 input buffers and 2 outputs;
- the last has 3 input buffers and 3 outputs.

Using this notation the traffic system can be represented as in Fig. 5 making the links between intersections.

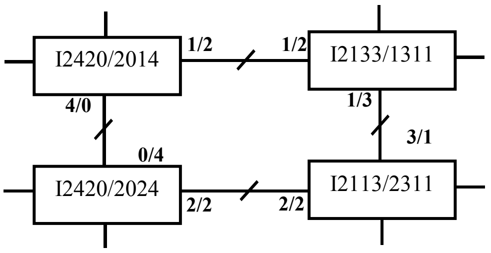


Fig. 5. Connections between intersections

Fig. 2 represents a crossroad of two streets each of them having four lanes. The cars can be driven on the three directions: left, right or forward. The traffic is controlled by traffic lights (S_{iL}, S_{iF}, S_{iR} for left, forward and right, where i =1..4). The duration of the green colour of a traffic light on a direction D_i for a specified lane is denoted by β_i. That expresses the maximum number of cars from the lane that crosses the intersection during a cycle. The sum of the green colour durations of all traffic lights of a crossroad can be fixed or variable.

Fig. 3 represents the the buffers of the crossroad. The symbols λ_{iL}, λ_{iF}, λ_{iR} denote the input rates (numbers of cars entering on a buffer during a cycle). For a buffer, Markov chain can be constructed as that represented in Fig. 4.

Singh and Titli [20] proposes for normal flow rates the model of an intersection as depicted in (7):

The variables as the following meanings:

$$x(k+1) = A \cdot x(k) + B_0 \cdot u(k) + \\ + B_1 \cdot u(k-1) + \cdots + B_\delta \cdot u(k-\delta) \\ y(k+1) = C \cdot x(k) \tag{7}$$

where: u is the input vector representing the input rates; x is the state vector representing the waiting queues; y is the output vector representing the output rates of different lanes; δ is the delay introduced by the distance between intersections.

The matrices A, B and C have the appropriate dimensions. This model works only around a steady point. The control concerns the reducing (eliminating) of the deviation of the state vector (i.e. the waiting queues).

Such linear model has a disadvantage on solving the problem only for normal traffic.

This model is extended to all the lanes of the intersections; otherwise the sequential usage of this model could lead to an unacceptable deviation from the real situation. Because that is not a realistic constraint, another more reasonable one is proposed. All the intersections have the period of the cycle a multiple of a fixed duration τ called *sub-cycle*. The green colour of each lane is given (planned) related to this duration. In this case, it is accepted that a lane is open during a cycle in more than one sub-cycle. The system is modelled with the time step given by τ time units. For the usage of the model an object model of the traffic control system is developed. The previous model is enhanced with a simple controller that applies the specified opening duration.

When the distance between intersections leads to significant delays of the car arrivals to the next intersection, this can be modelled by introducing fictive intersections (without crossroad) that are not open during a sub-cycle (of duration τ) but having the maximum transfer capacity on the next sub-cycle. The model precision of temporal approximation of the traffic system is given by τ.

2.4 *The controllers*

For the control, a hierarchical two level architecture is used. Two types of control structures are proposed: a centralised structure and a distributed structure. Depending on the

A model taking into account the non-linearity of the traffic around an intersection is proposed. A waiting queue having $x_i(k)$ cars and the β_i the transfer capacity (number of cars) for the current cycle is considered. At the end of the cycle, the waiting queue contains $x_i(k+1)$ cars and $y_i(k)$ cars went out.

The model that is given by the following relations:

$$x_i(k+1) = \begin{cases} 0 \text{ if } x_i(k) + u_i(k) \le \beta_i(k) \\ x_i(k) + u_i(k) - \beta_i(k) \text{ otherwise} \end{cases} \tag{8}$$

$$y_i(k) = \begin{cases} x_i(k) + u_i(k) \text{ if } x_i(k) + u_i(k) \le \beta_i \\ \beta_i(k) \quad \text{otherwise} \end{cases} \tag{9}$$

β_i represents the transfer capacity of the lane for the specified duration (i.e. cycle) that is given to controller by the upper level.

This model is extended to all the lanes of the intersection. The duration (periods) of the colour cycles should be the same for all the

structure type, different control algorithms are implemented.

2.4.1 The centralized control structure

A centralized two level control architecture is presented in Fig. 6.

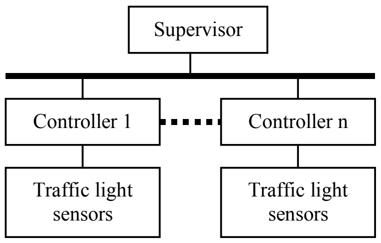


Fig. 6. The centralized architecture

The supervisor is a client (program) that asks the distributed controllers (considered as server programs) to perform some services as:

- send information about the intersection state and throughput
- apply a specified control algorithm with the given parameters

The supervisor has to solve a problem of resource allocation in a large scale system, [8].

The supervisor performs the following:

- adapts the system model to the current situation taking into account the information continuously received from the controllers
- using the models and an optimisation algorithm (in our case we propose a hybrid

genetic algorithm) it tries to get the best suited control algorithm for each intersection (the algorithm parameters are determined too) - this activity is performed off line

- evaluates the intersections capacity utilisation and the system throughput
- simulates the system behaviour using other algorithms with the aim to change the current algorithms to improve the traffic system behaviour
- predicts and determines the traffic congestion analysing the violation of constraints.
- solves the congestion using (artificial intelligence) rules without taking into account the system performances
- stores information about traffic with the aims for further evaluation and analysis.

To determine the control algorithm parameters was used a genetic algorithm. Each gene of the chromosome codes the duration of the green light (i.e. the transfer capacity of lanes) during a sub-cycle. The fitness function appreciates the transfer capacities of the intersections, but penalises the buffer overflow and the maximum delay to cross an intersection.

The Petri net model of the supervisor is presented in Fig. 7. The significance of transitions is: t_0 – the supervisor starts the activities; t_1 – sends messages to the controllers; t_2 – it receives the message from controllers or the waiting time has expired; t_3 – it reads the messages and updates the traffic model; t_4 – it starts the GA to find better control algorithms; t_5 – it builds a chromosome and evaluates the solution; t_6 – updating the population; t_7 – because the time expires or it gets a desired solution it sends to controllers the new control algorithms and their parameters; t_8 – it receives the acknowledgment from the controllers, or the waiting time has expired; t_9 – sending the updating model to the UDAS.

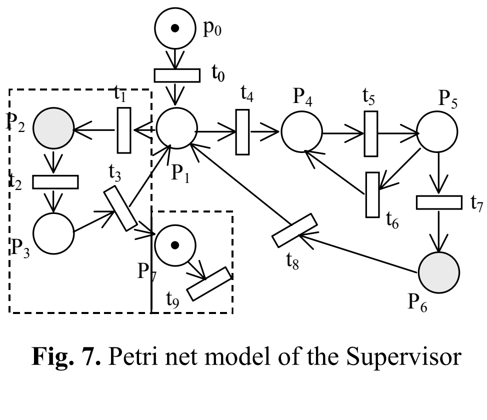


Fig. 7. Petri net model of the Supervisor

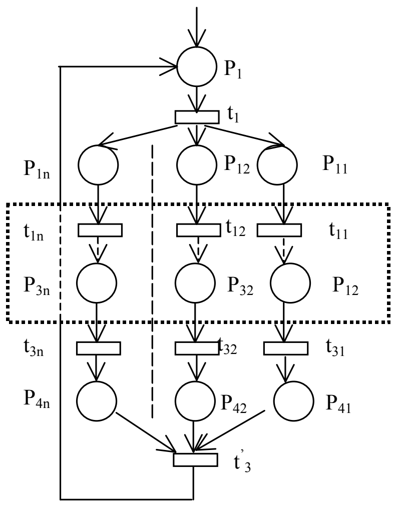


Fig. 8. The Petri Net of the subnet 1

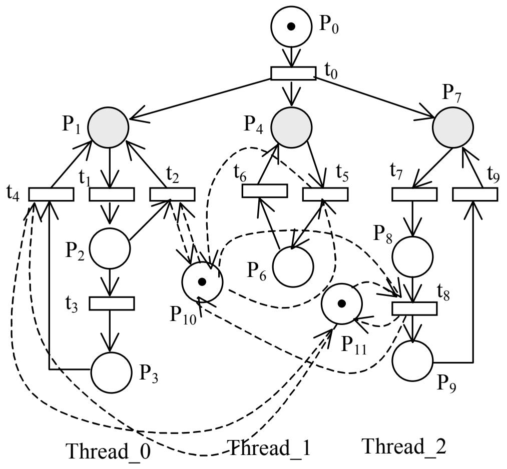


Fig. 9. The controllers Petri net model

In Fig. 8 the subnet 1 structure is in detail presented. The significance of the transitions is: t'_1 – building different threads for each controller; t_{1i} – the supervisor sends the corresponding message to the controller i (i=1,..,n).

The controller Petri net model is presented in Fig. 9. Three threads of executions are constructed for different sets of activities implementation.

The significance of places and transitions is: p_0 – initial state; t_0 – building the threads for controller; p_1 – thread_0 waits to receive a message from supervisor; t_1 – thread_0 reads the message; t_2 – if the supervisor requests state information, the controller sends it; t_3 – if the supervisor requests to change the controller algorithm the controller performs it; t_4 – the controller updates the parameters of the controller algorithm; p_4 – the thread_1 waits for a car to enter or to exit; t_5 – cars enter or exit from crossroad; t_6 – the thread_1 updates the state information of the controller; p_7 – the thread_2 waits (for time event) to execute the control algorithm; t_7 – the chosen control algorithm is started; t_8 – it calculates and applies the commands to traffic lights; t_9 – the current control algorithm is finished.

2.4.2 The distributed control structure

Many cars enter or exit the traffic inside the towns because they were or should be parked. In this case, the usage of the global states of the system is difficult since the current number of cars couldn't be known with a sufficient approximation. For this reason, the information given by the state is extended with the car flows to increase the precision of the proposed control method. When the input flows of a crossroads are larger than the output flows, an accumulation of cars waiting to cross it appears. If this exceeds the capacities of the lanes, the neighbour crossroads are blocked. The phenomenon extends quickly to the other crossroads, so the entire vehicle traffic system is congested.

The main idea of our research is to avoid the congestions by imposing flow rates through crossroads that don't allow the increasing of the length of the queue over the lanes capacities.

To control the traffic in a city neighbourhood, it is decomposed in a number of areas [15]. Each crossroad is controlled by a controller (possibly implemented on a Programmable Logic Controller (PLC)) and each area is supervised by a supervisor (usually implemented on a Personal Computer (PC)). So, the control system is composed by distributed supervisors that coordinate controllers (Fig. 10).

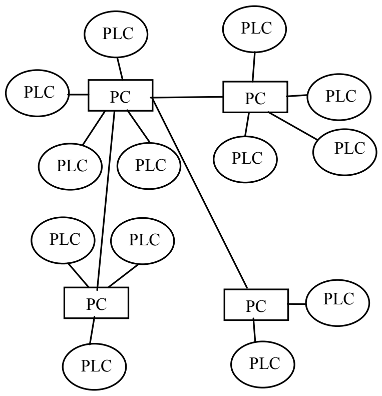


Fig. 10. The distributed control system

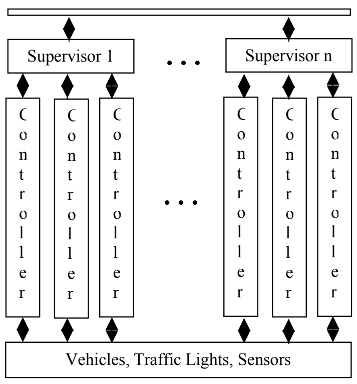


Fig. 11. The control system architecture

Fig. 11 represents the control system architecture. A controller is connected to the sensors (detectors) of the crossroad (that it controls) and can send control signals to the corresponding traffic lights. The controller architecture is drawn in Fig. 12.

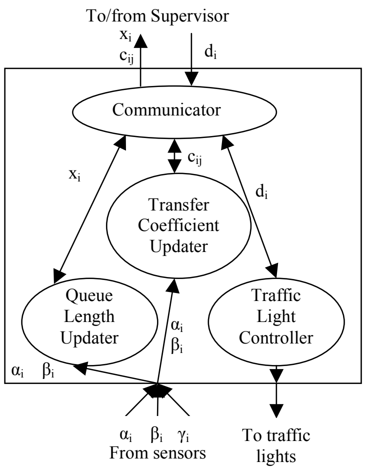


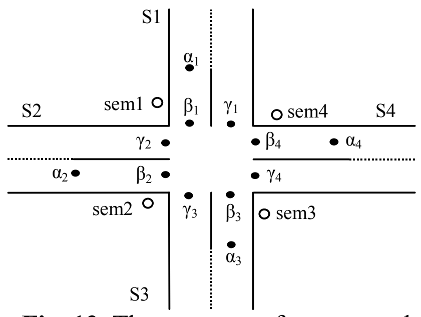
Fig. 12. The controller architecture

A controller is composed by four components (threads): Queue Length Updater (QLU), Transfer Coefficient Updater (TCU), Traffic Light Controller and Communicator (TLC).

For specification of the controller the Extended State Machines (ESM) [18] were used. This is a modelling tool useful for the computer, software and control engineers that provides facilities for describing communications, parallelism and real-time.

The *Queue Length Updater* has the task to count the number of cars on each lane that solicit to cross the intersection. This component receives the events (α_i, β_i) from the detectors and calculates the queue lengths increasing, respectively decreasing the (integer) values x_i representing the number of the cars.

Fig. 13 represents the structure of a simple crossroad linking the streets S1, S2, S3 and S4.


S1
S2
S3
S4
sem1
sem2
sem3
sem4
α_1 β_1 γ_1
α_2 β_2 γ_2
α_3 β_3 γ_3
α_4 β_4 γ_4
Fig. 13. The structure of a crossroad

The symbols denote:

- α_i (i=1,2, …) the input event produced by a car entering the input buffer on a lane
- β_i (i=1,2, …) the output event produced by a car exiting the input buffer on a lane and starting to cross the intersection
- γ_i (i=1,2, …) the input event produced by a car entering the output buffer on a lane that could be also an input buffer for the next crossroad
- sem1, sem2, … represent the traffic lights

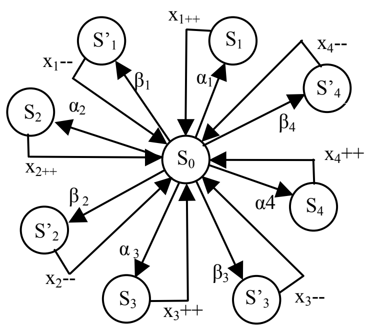
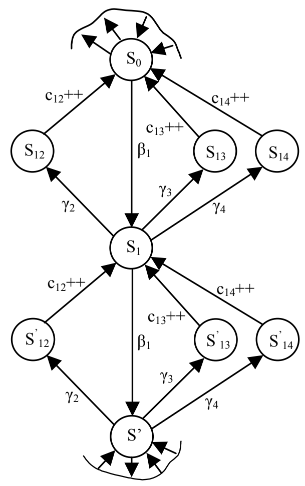


Fig. 14. The ESM of the QLU

The ESM of the *Queue Length Updater* implemented for the crossroad from Fig. 13 is represented in Fig. 14. The symbols x++ and x-- denote the increment respectively the decrement of the value x.

The *Transfer Coefficient Updater* receives the events (β_i, γ_i) from the detectors and calculates for each cycle the transfer coefficients c_{ij} from each input lane to each output lane of the crossroad. Its ESM is partially represented in Fig. 15.

Queue Length Updater and Transfer Coefficient Updater play the role of observers. The guard expression of the ESM could be used to make hem more robust at driver's incorrect behaviour.


S_0
S_{12} S_{13} S_{14}
S_1
S'_{12} S'_{13} S'_{14}
S'
c_{12}++ c_{13}++ c_{14}++
β_1 γ_2 γ_3 γ_4
Fig. 15. An ESM part (S1) of the TCU

The *Communicator* sends to the Supervisor the values of the queue lengths and the transfer coefficients. It receives from Supervisor the durations (d_1, d_2,…) of the green colour and sends them to Traffic light controller.

The *Traffic Light Controller* gets the durations (d_1, d_2, …) of the green colour and applies them to the traffic lights.

As in [9], a hierarchical solution was chosen. The Supervisor architecture is given in Fig. 16. The Local Supervisor has the task to coordinate the subordinated controllers activities and to obtain their traffic light durations. The sum of the input flow rates of a crossroad must not exceed the maximum crossroad's transfer capacity. The supervisors use the flow prices as information about how important it is for the current flows to use the requested resources. The supervisors distribute the times of using the resources taking into account their capacities and the criticality of the requests.

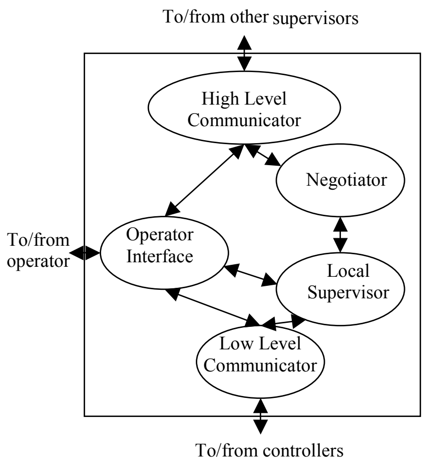


Fig. 16. The supervisor architecture

The criticality of each resource is distributed further into the entire net through prices.

The Negotiator cooperates with the neighbour supervisors to obtain the car rates crossing the border between controlled areas.

The approach of the current method of the traffic control system is based on *the principles of the free market*. The transfer prices of cars through intersections should control the traffic. The reason for using this is given by the difficulty to collect all the information about the traffic parameters in time, to process it and to react (to send the control signal to the distributed system) fulfilling the real-time constraints.

Each input car flow in the system has a price assigned by the traffic operator. The crossroad's Supervisor assigns further prices to its output flows taking into account its criteria. Each supervisor asks its neighbour supervisors to accept flows at calculated rates and prices. Each supervisor grants totally or partially the requested rates taking into account some rules and maintains at least a minimum (specified safely) rate for each flow.

The supervisor acts using the principles of the market. It allocates for all crossroad input lanes a minimum specified duration. The rest of the transfer capacity of the intersection is negotiated with its neighbours. The supervisor's algorithm has three phases:

Initially, the requested rates and the offered prices for the input lanes in the systems are given by the operator. The granted rates are set such that each input lane of a crossroad gets the same duration; this means the same granted output rate from the lane. The matrix of the transfer coefficient (rates) of each crossroad is calculated continuously from events signalled by detectors .

1. The bid phase. Each crossroad has a local negotiator (a representative of intersection) that reads the rates requested at its input lanes and their prices. It calculates the rates and the prices of its output lanes and stores them as request rates and offered prices. The negotiator sums its requested input rates to calculate the crossroad load. The prices are calculated using the prices offered for the input lanes and the load of the crossroad. This phase continues until no negotiator changes its requested rates and the offered prices.

2. The grant phase. Each negotiator uses the granted rates of its output lanes and calculates the rates which it grants for its input lanes. For this aim it considers that the rates of its output lanes cannot be exceeded. Also the sum of rates which it grants for its input lanes cannot exceed its maximum throughput (capacity). The negotiator grants the input rates giving priority for the lanes with higher prices. The phase is finished when no negotiator changes the rates (or the prices) that it grants for its input lanes. This represents a solution to the problem.

3. The evaluation phase. The supervisor calculates the performance of the solution summing up for each crossroad the differences between the crossroad capacity and its current throughput.

The results of this algorithm are the number of cars crossing the intersection on each lane (or approximately equivalent, the durations of the green light of the traffic lights).

The distributed supervisors perform nearly the same algorithm as the local supervisors. They have some gates on the border between the controlled areas (Fig. 9) where the flows pass from an area to another. The negotiators from different areas determine the flow rates crossing their borders. For this aim they use:

- the numbers of cars soliciting to cross the intersection on each lane per minute;
- the transfer coefficients corresponding to each lane;
- the prices of the flows; the load factors of the input buffers (i.e. how much of the input lanes is occupied by cars)

The two limits for choosing the period to grant (i.e. the time horizon of allocation) the resources are given by the dynamic of changing the parameters of the traffic system and the duration necessary to collect the new transfer coefficients.

3. THE UDAS

The routing algorithm computation is embedded in an Urban Driving Advisory System [1].

In addition to the detailed control of the traffic, the UDAS can improve the overall traffic quality and the decrease of traffic flow. For providing a good routing algorithm, the UDAS receives real-time information about the current travel time from the Real-Time Control System, which uses sensors placed at specified intervals along the route. Since traffic condition can fluctuate over shorter or longer time periods, the real-time information on current traffic conditions represents only a part of the information necessary to find a convenient route. Therefore, current traffic information is not sufficient for route guidance and it is important that a designed system to provide guidance to drivers based on the predicted travel times in the network.

The main goal of the routing strategy is to ensure an optimum utilization of the capacities offered by the existing main streets, to equalize the flow on these streets during the periods when the traffic flow increases, and combined with the control algorithm, to maintain traffic fluidity. The problem of drivers advisory falls into the problem of shortest path computation on a specified graph, under some special circumstances. The computation of shortest path over a network has been the target of many research efforts. These efforts have resulted in a number of different algorithms and a considerable amount of empirical findings with respect to performance [25]. To solve the shortest path problem, there are a lot of requirements that should be taken into account for choosing the better algorithm:

- the optimisation may be performed with respect to various metrics, usually distance, travel time, fuel consumption, etc. ;
- traditional algorithms may impose an unacceptable long computational time when applied to realistic road networks;
- additional constraints may be imposed;

The shortest path algorithm is usually a multi-objective minimization problem. Instead of finding the best solution considering as metric the distance or the travel time, it is necessary to find the best solution on distance and time, taking into account the actual context in the street network [23].

3.1 *The UDAS model*

The tasks performed by UDAS are complex

- receives information about the actual flows on the streets from RTCS;
- updates the streets model (streets graph) in accordance with the recently received information;
- computes the shortest path;
- sends advice to customers;

The time Petri net model of the UDAS is presented in Fig. 17. The significance of the

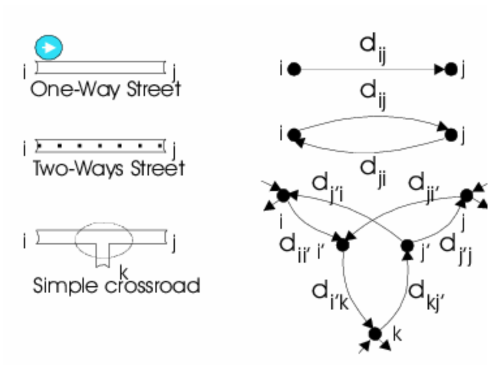


Fig. 18. Different graphs related to streets

transitions are: t_1 - connecting to the network; t_2 - acknowledgment from network; t_3 - connecting to the Data Base Server; t_4 - accessing the Data Base; t_5 – computing the shortest path ; t_6 - send the message; t_7 - receive the message; t_8 - receive data from RTCS; t_9 – apply penalty function and compute the new link weights on the graph; t_{10} – update the model.

3.2 *The street model*

The first step is to represent the street network as a graph, and store the graph on a database. Each vertex on the graph is oriented, denoted by (i,j), labelled with a label d_{ij}, representing the cost of the link. A two-ways street is represented by two links, (i,j) and (j,i), where only in particular cases $d_{ij}=d_{ji}$. A simple crossroad is represented by a graph consisting in 5 nodes and 6 links, and so on (Fig. 18).

The algorithm consists in the following steps:

- build the associated graph from the geographic information stored in the database;
- label the links with the costs $d_{ij}=\delta_{ij}$, where δ_{ij} is the distance (in km.) between two nodes;
- query the database and extract the actual traffic densities on the streets and compute the average speeds υ_{ij} for the arks related to the streets;
- for each link (j,i), compute a penalty function p_{ij} given by [23]:

$$p_{ij} = \max(d_{ij}) - \frac{\max(d_{ij})}{1 + e^{-\gamma(\upsilon_{ij} - \frac{V_{ij\max}}{2})}} \qquad (10)$$

where

- $\max(d_{ij})$ is the length of the longest street;
- υ_{ij} is the actual average speed reported by RTCS for the correspondent street, at the moment;
- γ is a weighting coefficient and
- $V_{ij\,\max}$ is the maximum speed allowed on the specified street;

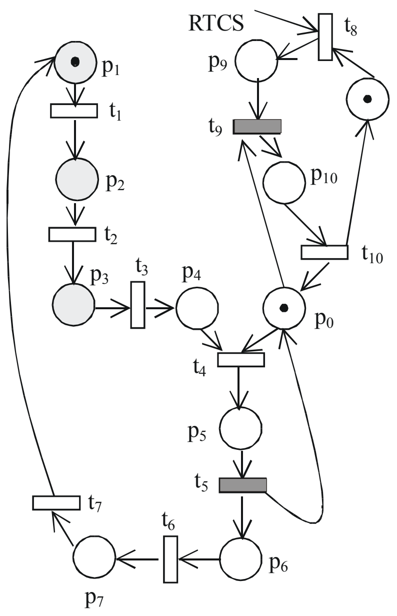


Fig. 17. The Petri Net model of the UDAS

- compute the actual label of the links as:

$$D_{ij} = d_{ij} + p_{ij} \tag{11}$$

if on a street the average speed is very low (i.e. a congestion is imminent), the power of the link for the correspondent link will be significantly increased;

In Fig. 19 is depicted the Petri net for the model building.

The significances of the transitions are: t_{91} – read from database street lengths and information for computing average speeds; t_{92} – store data and compute the average speeds; t_{93} – compute and apply the penalty functions; t_{94} – wait until buffers are empty, for a new data read; The p_{94} place is initially marked, because new data can be read before updating the new model, if a valid model is already available.

3.3 *The shortest path algorithm*

Two routing algorithms were tested. The first is the well known Dijsktra algorithm. It finds the shortest path in a simple way and is easy to implement, but it leads to large computation time which increases exponentially with the number of nodes in the graph. An alternative is the modified Dijkstra algorithm, which reduces the number of operations.

Better results were obtained by using a genetic algorithm for computing the shortest path [24].

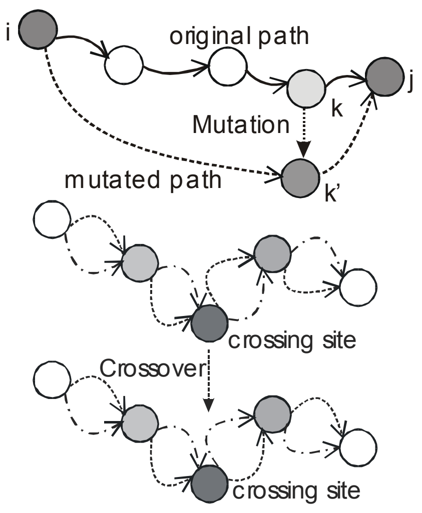


Fig. 20. Genetic operators

$$f_i = \frac{\frac{1}{\eta_i}}{\sum \frac{1}{\eta_i}} \quad \text{where} \quad \eta_i = \frac{\frac{1}{D_i}}{\sum \frac{1}{D_i}} \tag{12}$$

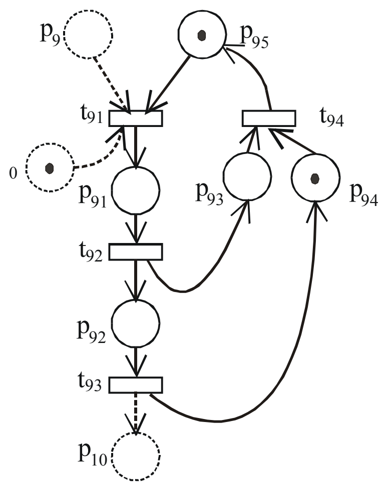


Fig. 19. The model building

The objective of the genetic algorithm is to find the shortest path from source to destination, taking into account the new metric on the graph, using a chromosome inspired from Ford-Fulkerson algorithm. The fitness function for chromosome i is depicted in equation 12 [17]

The chromosome encodes the possible ways. Only two genetic operators were used, the *mutation* which changes the way, by selecting an alternative intermediary node and the *crossover*, which exchange from an intermediary node two alternative routes.

4. CONCLUSIONS

In the paper, the problem of urban traffic control was presented. Two types of control structures were proposed: a centralised structure and a distributed structure. The centralised structure solves the problem around a state, which often has to be changed during a day. They have the advantage of a global overview of the problem. Unfortunately, this leads to large computation requirements for large scale systems. The distributed structure transforms the events produced by the cars passing over detectors into information used by the upper levels; adapts itself to the changing of the characteristics of the traffic through transfer coefficients; finds a solution for distributed allocation of the resources taking into account the importance of the requests; works even if a part of the system is blocked.

The control system is enhanced with an advisory system, which sends advice concerning the best route to the drivers and also acts as a primary traffic formatter.

REFERENCES

[1] Aştilean, A., Avram, C., Leţia, T., Vălean, H., Hulea, M., "Routing Algorithms As A Part Of Control Startegy In An Urban Driving Advisory System", The 4th Int. Conference on Electromechanical and Power Systems, pp. 155-158, 2003.

[2] Babaoglu, O., Marzullo, K. "Consistent Global State of Distributed Systems: Fundamental Concepts and Mechanisms", Distributed System. Ed. Sape Mullender, ACM Press New York, 1993.

[3] Ben-Naim E., Krapivsky P. "Kinetic Theory of Traffic Flows", Ed. Springer, Berlin, 1996.

[4] Carbone, C.,Boel, R. "A Timed Discrete Event Model for Urban Traffic Control", Proc. of the Benelux Meeting on System and Control, 2004.

[5] Cremer, M., Zhang X.. "System Architectures for Complex Road Traffic Information and Control Systems", Proc. of IFAC 12th World Congress, pp. 229-232, 1993.

[6] Dia, H. "Neural Network Models for Freeway Incident Detection", Proc. of the 16th Conference of Australian Institute of Transport Research, 1994.

[7] Gerken, J. "A Practical Approach to Managing Intersection Traffic Data for Large Scale Studies." Mid-Continent Transportation Symposium, 2000.

[8] Gessing, R., Duda, Z., "Two-level Optimal Resource Allocation in a Large Scale System", Preprints of the IFAC World Congress Automatic Control, Vol. 10, p.74-78, Talin, Estonia, 1990.

[9] Godbole, D.N., Lygeros, J., "Hierarchical Hybrid Control: a Case Study", Hybrid Systems Institute of Transportation Studies, University of California, Berkley, p.166-190, 1994.

[10]Klar, A., Kuhne, R.D., Wegener, R. "Mathematical Models for Vehicle Traffic", Surveys on Mathematics for Industry, 6, p. 215-239, 1996.

[11]Leţia, T., Aştilean A., Hulea M., Avram C., Vălean H. "Urban traffic control system." 2002 4th International Workshop on Computer and Information Technology, pp. 165-171, 2002.

[12]Leţia, T., Aştilean A., Hulea M., Avram C., Vălean H. "Object street traffic model." Proc. IEEE-TTTC International Conference AQ-TR 2004, pp. 170-175, 2002.

[13]Letia T., Astilean A., Hulea M, „Algorithms for urban vehicle traffic control", Proc. of The 14th International Conference on Control Systems and Computer Science, Ed. Politehnica Press, Vol. 1, Bucharest, Romania, p. 390-393, 2003.

[14]Leţia, T., Hulea, M., Avram, C., Vălean, H., "Real-Time Traffic Information and Control System", 27th IFAC/IFIP/IEEE Worksop on R-T Programming WRTP'03, Poland, pp. 205-210, May 14/17, 2003;

[15]Leţia, T.,Aştilean, A., Hulea, M., Vălean, H., "Flow Price Based Vehicle Traffic Distributed Control", Proc of DESDes'04, Poland, pp. 91-96, ISBN 83-89712-16-4 , 2004;

[16]Militzer, B. "Traffic Jams.", http://militzer. gl.ciw.edu /traffic_jams/index.html, 1998.

[17]Munetomo M., Takai Y., Sato Y. "An Intelligent Network Routing Algorithm by a Genetic Algorithm", Proceedings of the Fourth International Conference on Neural Information Processing, pp.547-550 , 1997.

[18]Ostroff, S.J."Temporal Logic for Real-Time Systems", Research Studies Press Ltd., Tauton, Somerset, England, 1989.

[19]Porche, I.,Lafortune, S., "Adaptive Look-ahead Optimization of Traffic Signals", Proceedings of the 1997 IEEE Conference on Intelligent Transportation Systems, Boston, MA., 1998.

[20]Singh M., Titli A., “Systems: Decomposition, Optimization and Control”, Pergamon Press, Oxford, 1978.

[21]Tadaki, S., Kikuchi, M., Sugiyama, Y. Yukawa, S. “Coupled Map Traffic Flow Simulator Based on Optimal Velocity Functions”, http://hirose.ai.is.saga-u.ac.jp, 1997

[22]Van Aerde, M., Hellinga, B., Baker, M. Rakha, H., “Integration: An Overview Traffic Simulation Features” Transportation Research Board, Annual Meeting. Queen’s University, Kingston, Canada, 1996.

[23]Vălean, H., Leția, T., Aştilean, A., “Urban Routing Algorithm For A Driving Advisory System”, Annals of the University of Craiova, no. 27, Vol. I, pp. 204-210, ISSN 1223-530X, 2003.

[24]Vălean, H., Leția, T., Aştilean, A., “Urban Driving Advisory System Based on Genetic Algorithms”, Proc of DESDes’04, pp. 117-122, ISBN 83-89712-16-4 , Poland, 2004.

[25]Zhan, B., Noon, C., “Shortest path algorithms: An evaluation using real road networks”, Transportation Science, 1998.

[26]*** “Developing Traffic Signal Control Systems Using the National Architecture. U.S.” Depnt. of Transportation. Intelligent Transportation Systems Joint Program Office, Report FHW-JPO-98-026, 1998.