Light Controlled Intersection Model Based on the Continuous Petri Net ⋆

Michal Kutil, Zdeněk Hanzálek

Centre for Applied Cybernetics, Department of Control Engineering,
Faculty of Electrical Engineering, Czech Technical University in Prague,
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
(e-mail: {kutilm,hanzalek}@fel.cvut.cz)

Abstract: We propose a light controlled intersection model based on a constant speed continuous Petri net. The Model is innovative, firstly, by the free space modeling together with the opposite direction of the vehicular flow, secondly by the continuous Petri net use only leading to a smaller state space. For this purpose, we show a new method for conflict resolution in a Petri net based on the maximal speed proportion. The model is compared with intersection model based on a classical discrete Petri net and it is evaluated in the simulation where real traffic data is used for the incoming flow.

Keywords: continuous Petri net, conflict resolution, traffic, intersection model, simulation

1. INTRODUCTION

The application of information technologies to the traffic system, that is Intelligent Transportation Systems (ITS), offers new and effective solutions to the congestion problem. These systems combine appropriate intersection control strategies, Papageorgiou et al. (2003), Kutil et al. (2006), Di Febbraro and Giglio (2006) and traffic flow models Castillo (1996), Masukura et al. (2007) to reduce congestion, vehicle delay time and fuel consumption. Traffic flow models consider traffic flow as a heterogeneous system which can be described from a macroscopic or microscopic point of view, Tolba et al. (2005). On one hand, the macroscopic point of view is described by global variables as the flow rate, the flow density and the flow average velocity, Gazis (2002). On the other hand, microscopic point of view focuses on the individual vehicles behavior on the road.

The traffic flow model based on a discrete Petri net (PN), Wang et al. (1993) is suitable for a microscopic description. On the other side, the model based on a continuous Petri net is suitable for macroscopic description, Tolba (2001). The combination of both concepts gives us model that consists of a continuous part in the street and a discrete part in the intersection. Hybrid Petri nets (HPN), David and Alla (2001), are used for describing these complex models, Di Febbraro et al. (2004). The main disadvantage of this concept based on HPN is a discrete part of the Petri net used for traffic flow, Tolba et al. (2003) or a discrete part of the PN used for intersection control, Di Febbraro et al. (2001), Júlvez and Boel (2005) which subsequently discretizes traffic flow. In these cases, the continuous flow is cut to the different flow levels by the intersection part, which is further processed by a continuous part of the HPN. Due to discrete part, processing such models is not efficient.

The aim of this paper is to develop a new light controlled intersection model based on the *Constant speed Continuous Petri Net* (CCPN), David and Alla (1998). This model divides the continuous traffic flow in consequence of the intersection structure and control without cutting the flow to different flow levels. Moreover, free space modelling together with the opposite direction of vehicular flow is considered. For this goal, we show a new method for conflict resolution in the CCPN based on maximal speed proportion. As a result, we show that the simulation based on our model is faster than based on the conventional approach.

This paper is organized as follows: Section 2 describes the Petri net background including the new conflict resolution, concept CCPN extensions and CCPN evolution based on Linear Programming. Section 3 shows the light intersection model by the CCPN. The performance evaluation of the continuous model on real data from the traffic in Prague is reported in Section 4. In that section, a discrete PN model is used for verification. The conclusion remarks and directions for future research are given in the last Section.

2. PETRI NET BACKGROUND

A constant speed continuous Petri net is defined as a sextuple $R = \langle \mathcal{P}, \mathcal{T}, V, Pre, Post, M_0 \rangle$, where the definition of $\mathcal{P}$, $\mathcal{T}$, Pre, $Post$ are similar to those of the discrete Petri nets, as well as ${}^\circ T$, T°, ${}^\circ P$, P° notation for predecessor and successor places and transitions, both described, for example, by David and Alla (2004). M_0 is the initial marking. $V : \mathcal{T} \to R^+ \cup \{\infty\}$ is a vector of maximal firing speeds. V_j denotes the *maximal speed* of transition T_j. Further more $v_j(t)$ denotes the *speed* of transition T_j at time t. The value of $v_j(t)$ is bounded by the interval

⋆ This work was supported by the Ministry of Education of the Czech Republic under Project 1M0567.

$\langle 0, V_j \rangle$. Transition T_j is *strongly enabled* at time t if all places P_i of ${}^\circ T_j$ are marked. Place P_i is *supplied* at time t if there is at least one transition T_j in ${}^\circ P_i$, which is enabled (strongly or weakly). Transition T_j is *weakly enabled* at time t if there is place $P_i \in {}^\circ T_j$, which is not marked and is supplied, and the remaining places of ${}^\circ T_j$ are either marked or supplied. For more information, see Hanzálek (2003). The following extensions of the CCPN will be used: Inhibitor arcs and the marking of continuous place can take the value 0+ in addition, both described by David and Alla (2004).

2.1 Conflict Resolution

In a continuous Petri net, there is an actual conflict among s transitions in a set $\{T_{(1)}, T_{(2)}, \ldots, T_{(s)}\}$, if the speed of at least one of them has to be less than the maximal speed due to the speeds of the other transitions in this set.

When there is an actual conflict between two or more transitions, then the priority rule can be used. Another alternative is to use a conflict resolution method based on the *maximal speed proportion*. The maximal speed proportion means that the flow which runs through the place is divided into the subsequent transitions in proportion to their maximal speeds, if it is feasible in regard to other constraints. The method is based on a flow sharing rule, observing the *maximum firing speed* described by Hanzálek (2003) and can be used only in simple Petri nets. A simple Petri net is a net in which each transition can only be concerned with one conflict at most.

In order to illustrate the conflict resolution method we have shown an example in Fig. 1. There are three places P_1, P_2 and P_3 with zero marking. These places are supplied by the transitions T_1, T_2 and T_3 with a speed $v_1 = 35$, $v_2 = 40$ and $v_3 = 18$, respectively. Place P_2 is supplied with a smaller speed than the sum of the maximal speed of the transitions P_2° i.e. $v_2 < V_4 + V_5$. Therefore, there is an actual conflict between transitions T_4 and T_5.

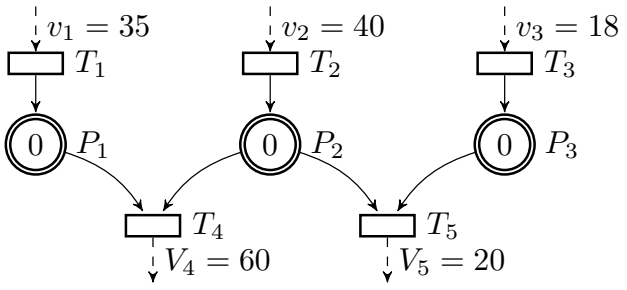
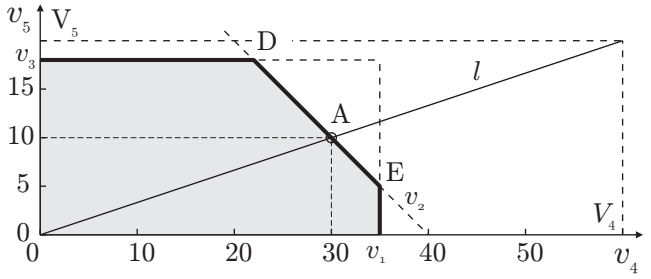
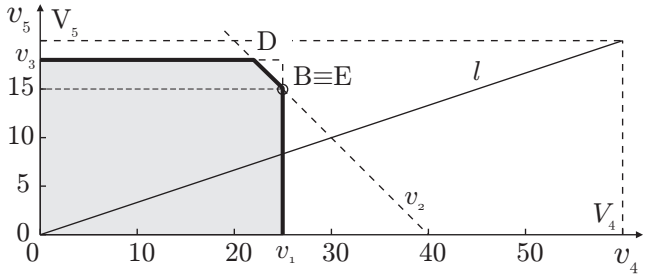


Fig. 1. Petri net example with actual conflict

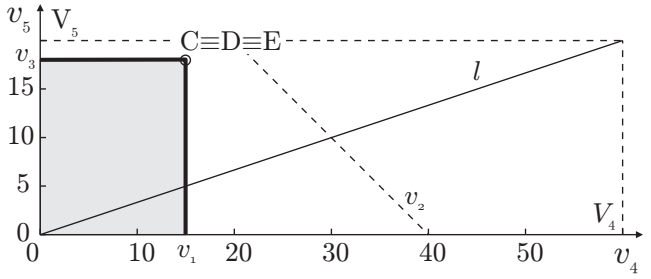
A geometrical interpretation of the conflict resolution is given in Fig. 2. The axes show the speeds of transitions T_4 and T_5. A gray polygon bounds the area where the possible speed v_4 and v_5 can be placed. This polygon is bounded by v_1, v_2 and v_3 from one side and by the axes from the other side. All combinations of the speeds v_4 and v_5 which are in proportion to the maximal speed V_4 and V_5 are located on the line l. Point A, in Fig. 2(a), is where v_4 and v_5 are at a maximum and are located on line l. The conflict resolution is given by this point A and the speed transitions for T_4 and T_5 are $v_4 = 30$ and $v_5 = 10$, respectively.



(a) Proportional resolution



(b) Resolution close to proportion



(c) Resolution close to proportion without the transition T_2 effect

Fig. 2. Geometrical interpretation of the conflict resolution

Let us consider the modification of speed v_1 from $v_1 = 35$ to $v_1 = 25$ (see Fig. 2(b)). This situation corresponds to point B where v_4 and v_5 are at a maximum and are located closest to line l. The conflict resolution is given by this point and the speed of the transitions T_4 and T_5 is $v_4 = 25$ and $v_5 = 15$, respectively. Note that the speed is not in proportion to the maximal speed V_4 and V_5, but it is the maximum firing speed combination.

Let us assume v_1 to be equal to 15 (see Fig. 2(c)). This situation corresponds to point C where v_4 and v_5 are at a maximum and are located closest to line l. The conflict resolution is given by this point and the speed of the transitions T_4 and T_5 is $v_4 = 15$ and $v_5 = 18$, respectively. Note that there is no actual conflict in this case.

We can generalize that the conflict resolution corresponds to the point located on the line segment DE closest to line l. In the special case when point D is equal to E i.e. $|DE| = 0$ the transition speed is placed to this point, see Fig. 2(c) for point C.

This geometrical interpretaion of speed computing can be formulated as follow: consider place P_i and the set of

Algorithm 1. Transition speed computing

while $v_i^s(t) > 0$ and $\mathcal{C}_i$ is not empty **do**
  $\mathcal{C}_i' = \mathcal{C}_i$
  for all j such that $j \in \mathcal{C}_i'$ **do**
    $v_j'(t) = \min\left(v_j^{max}(t) - v_j(t), v_i^s(t)\frac{V_j}{\sum_{k\in\mathcal{C}_i'} V_k}\right)$
    $v_j(t) = v_j(t) + v_j'(t)$
    if $v_j(t) = v_j^{max}(t)$ **then**
      $\mathcal{C}_i = \mathcal{C}_i \setminus j$
    end if
  end for
  $v_i^s(t) = v_i^s(t) - \sum_{j\in\mathcal{C}_i'} v_j'(t)$
end while

transitions P_i° having a conflict. The set of transitions P_i° indexes is denoted as $\mathcal{C}_i$. Variable $v_i^s(t)$ denotes the sum of the speeds supplying the place P_i at time t. For each transition T_j where $j \in \mathcal{C}_i$ initialize $v_j^{max}(t)$ as a minimum from the maximal speed V_j and separately sum the speed of transitions which supplied the place from ${}^\circ T_j$ with zero marking. Set the transition speed $v_j(t) = 0$ and modify it by the iterative Algorithm 1.

2.2 Linear Programming for Conflict Resolution

This subsection shows how linear programming (LP) can be used for conflict resolution computation. At first, we will use the example in Fig. 1 to show how a conflict resolution can be solved by linear programming at time t. In this example, we are looking for the speed of T_4 and T_5 with the maximal speed proportion condition. The LP problem using the variables defined above is:

$$\max\left(v_4(t) + v_5(t) - \epsilon\left|v_5(t) - v_4(t)\frac{V_5}{V_4}\right|\right), \tag{1}$$

subject to:

$$\begin{aligned} v_4(t) &\geq 0,\\ v_5(t) &\geq 0,\\ v_4(t) &\leq V_4,\\ v_5(t) &\leq V_5,\\ v_4(t) &\leq v_1(t),\\ v_5(t) &\leq v_3(t),\\ v_4(t) + v_5(t) &\leq v_2(t), \end{aligned} \tag{2}$$

where ϵ is a small constant for which $0 < \epsilon < \min(1, \frac{V_4}{V_5})$ holds. The fraction $\frac{V_5}{V_4}$ is the slope of line l in Fig. 2.

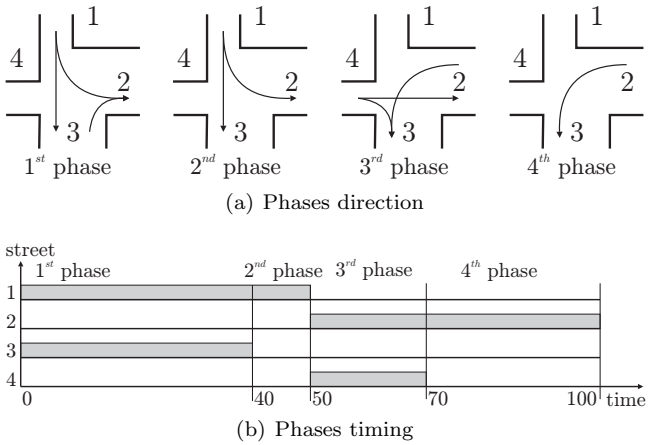
The objective function (1) can be rewritten with the use of a new variable. This variable z_{45} replaces the absolute value, which includes the v_4 and v_5 variables as follows:

$$\max\left(v_4(t) + v_5(t) - \epsilon z_{45}\right), \tag{3}$$

on the assumption that we add two new constraints:

$$\begin{aligned} v_5(t) - v_4(t)\frac{V_5}{V_4} &\leq z_{45},\\ v_5(t) - v_4(t)\frac{V_5}{V_4} &\geq -z_{45}. \end{aligned} \tag{4}$$

This LP problem gives us a solution in conformity of Fig. 2.



(a) Phases direction

(b) Phases timing

Fig. 3. Intersection description

Moreover, if there is a conflict among more transitions T_j where $j \in \mathcal{C}_i$ then it can be formulated as an LP problem as follows:

$$\max\left(\sum_{j\in\mathcal{C}_i} v_j(t) - \epsilon \sum_{\substack{k,l\in\mathcal{C}_i\\ k<l}} z_{kl}\right), \tag{5}$$

subject to:

$$\begin{aligned} v_j(t) &\geq 0, && \forall j \in \mathcal{C}_i\\ v_j(t) &\leq v_j^{max}(t), && \forall j \in \mathcal{C}_i\\ \sum_{j\in\mathcal{C}_i} v_j(t) &\leq v_i^s(t), &&\\ v_l(t) - v_k(t)\frac{V_l}{V_k} &\leq z_{kl}, && \forall k,l \in \mathcal{C}_i, k < l\\ v_l(t) - v_k(t)\frac{V_l}{V_k} &\geq -z_{kl}, && \forall k,l \in \mathcal{C}_i, k < l. \end{aligned} \tag{6}$$

where ϵ is a small constant number which $0 < \epsilon < \min_{\substack{k,l\in\mathcal{C}_i\\ k<l}} (1, \frac{V_k}{V_l})$ holds.

The solution of the LP problem mentioned above gives us the same results as Algorithm 1. Nevertheless, it is not intended for all Petri nets generally. It can be used only for a PN where the number of subsequent transitions which are under consideration for the objective function (5) for each P_i° are the same, which is our case.

3. LIGHT CONTROLLED INTERSECTION MODEL

In this section we will study the light controlled intersection. The intersection under consideration (see Fig. 3(a)) includes four input and two output flows.

The light controlled intersection is characterized by several parameters: the number of phases, duration of the phases and a list of streets from which the vehicles flow. Our model considers a constant sum of the phase time intervals, i.e. a constant *control period* T in seconds. The next parameter is the *distribution rate*. The distribution rate $\alpha_{s\to d}$ indicates the proportion of the vehicles which are crossing the intersection from the source street s to the destination street d, where the variables s and d denote the indexes of the street numbers. The sum of the distribution rates for each input street must be equal to one. For each distribution rate, we consider the *real*

Table 1. Intersection parameters

s	d	$\alpha_{s\to d}$	$w_{s\to d}$ [km/h]	$t_{s\to d}$ [s]	$d_{s\to d}$ [s/uv]	$V_{s\to d}$ [uv/s]	$V_{s\to}$ [uv/s]
1	2	0.2	30	50	0.60	0.83	1.23
1	3	0.8	50	50	0.36	1.39	
1	2	0.2	30	10	0.60	0.17	0.25
1	3	0.8	50	10	0.36	0.28	
2	3	1	30	50	0.60	0.83	0.83
2	3	1	30	30	0.60	0.50	0.50
3	2	1	20	40	0.90	0.44	0.44
4	2	0.4	50	20	0.36	0.56	0.29
4	3	0.6	20	20	0.90	0.22	

vehicle speed $w_{s\to d}$ in kilometers per hour. Further more, $t_{s\to d}$ denotes the *duration* how long the vehicles flow with the considered distribution rate.

The intersection under consideration includes four control phases. The source and destination streets are marked in Fig. 3(a). The duration of the vehicle flow through the intersection from a street can be found in Fig. 3(b). For this intersection, the control period T is assumed to be 100 seconds. The summary of all the parameters under consideration is presented in the left part of Table 1.

We can see that some of the source-destination pairs are there twice differing in duration. This difference is given by a give way rule in addition to the light control. For example, see Fig 3(a) where, during the 3$^{\text{rd}}$ phase the flow from street 2 gives way to the flow from the street 4. If there is a flow from street 4 then the duration is equal to 30s, otherwise it is equal to 50s.

From these intersection parameters, we can compute the parameters necessary for the continuous Petri net. The first one is the delay $d_{s\to d}$, which denotes the time for which one vehicle flows throw the intersection in seconds per unit vehicle:

$$d_{s\to d} = \frac{3.6 l_{uv}}{w_{s\to d}}. \tag{7}$$

The second parameter is the maximum average speed $V_{s\to d}$ over the period T in unit vehicles per second:

$$V_{s\to d} = \frac{w_{s\to d} t_{s\to d}}{3.6 l_{uv} T}. \tag{8}$$

Where the constant l_{uv} is the length of the unit vehicle including the distance between the vehicles in meters. In this paper, we will consider l_{uv} to be equal to 5m. The speeds with a common source can be combined with respect to the distribution rate to one common speed as follows:

$$V_{s\to} = \frac{\sum_d \alpha_{s\to d}}{\sum_d \frac{\alpha_{s\to d}}{V_{s\to d}}}. \tag{9}$$

The summary of all these parameters is presented in the right part of Table 1.

3.1 The continuous Petri net intersection model

The Petri net intersection model described in Fig. 3 is shown in Fig. 4. The net is divided to six parts. First, part A includes eight places, which are used as an interface for the inputs into the intersection from the streets. For each street, there are two places, the first one for vehicles waiting in the queue before the intersection (P_{101}, P_{102}, P_{103}, P_{104}), the second one for the free space in the street (P_{151}, P_{152}, P_{153}, P_{154}). The *free space* denotes the tokens which represent the available space for the vehicles, which can flow into the street. The free space modelling together with the opposite direction of vehicular flow is an innovative approach to this model. The distribution rate through the intersection is shown in parts B, C, D and E, each one for one input street. These parts of the network are very similar, thus only part C will be described. The transition T_{40} consumes vehicles from street 4 and the free space is returned back to the place P_{154}. Vehicles are divided up into distribution rate acceptance, to the places P_{40} and P_{43}. The weights of the arcs are equal to the distribution rate $\alpha_{4\to 2}$ and $\alpha_{4\to 3}$. From places P_{40} and P_{43}, the vehicles flow through the transitions T_{42} and T_{44} to the output streets represented by places P_{202} and P_{203}, respectively. The free space from the output streets close the loop l_1 (l_2) and flow to the place P_{154} through transition T_{40}. The speed of all the transitions in the loop is the same. It is bounded by maximal speed which is taken from Table 1, i.e. for example, the maximal speed V_{42} and V_{43} are equal to $V_{4\to 2}$.

Transitions T_{12}, T_{17}, T_{32}, T_{42} and T_{14}, T_{19}, T_{20}, T_{21}, T_{44}, respectively are in conflict (places P_{252} and P_{253}). This conflict is solved by the above described (see subsection 2.1) conflict resolution method based on maximal speed proportion. This method guarantees the same free space distribution as in a real traffic intersection.

Parts D and E include two parallel similar sub networks. Input transitions to those sub networks T_{10}, T_{15} and T_{20}, T_{21}, respectively, are in conflict (places P_{101} and P_{102}). This conflict is structural but not effective, because only one of them may be fired at the same moment. It is given by the arcs and the inhibitor arcs between places P_{103} and P_{104} respectively and the mentioned transitions.

See Fig. 3(a) there is no flow into street 2 during the 4^{th} phase. During this time the free space is accumulated in place P_{252}. Thirty percent (duration of the 4^{th} phase in the control period T) of the free space is saved to the temporary place P_{352}, by the arc from T_4 to P_{352} with weight equal to 0.3. Transition T_4 is a part of the street model. This free space must be utilized during the next traffic phase. The order of the phases is given (see Fig. 3(a)) and together with the give way rule it designates the priority for transitions T_{11}, T_{16}, T_{31}, T_{41}. Those transitions consume marking from places P_{352} and P_{252} and the conflict is solved by the priority.

4. PERFORMANCE EVALUATION

In this section the performance evaluation of the previously described intersection model simulation is shown. Prague intersection of "V Botanice" and "Zborovská" streets (GPS location: 50°4′31.484″N, 14°24′27.36″E) is used as a real system for the simulation. The street "V Botanice" includes three traffic lanes and the street "Zborovská" includes two traffic lanes. The intersection scheme and control phases are shown in Fig. 5.

The real data for the performance evaluation of the simulation is measured by the inductive loop detectors.

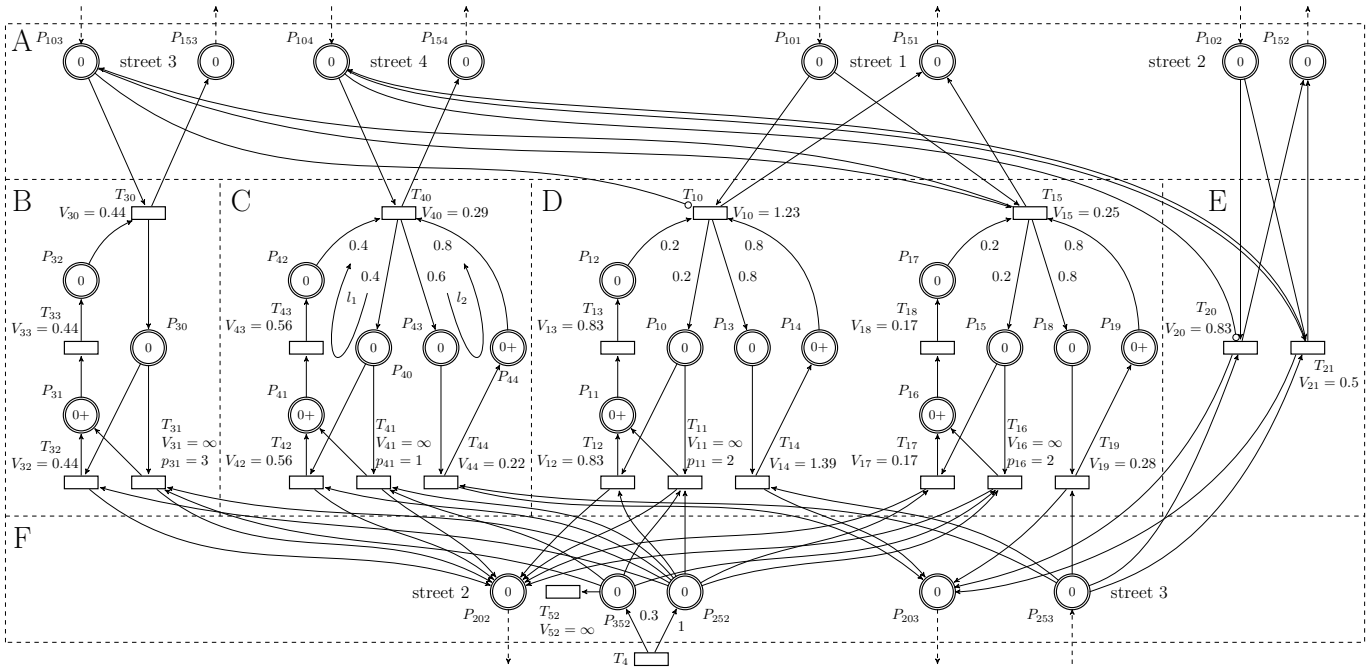


Fig. 4. Light controled intersection Petri net model

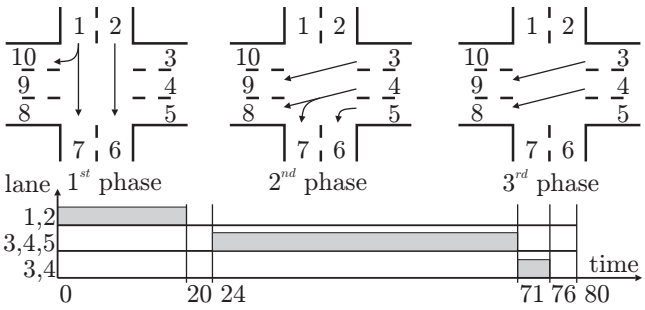


Fig. 5. Prague intersection diagram

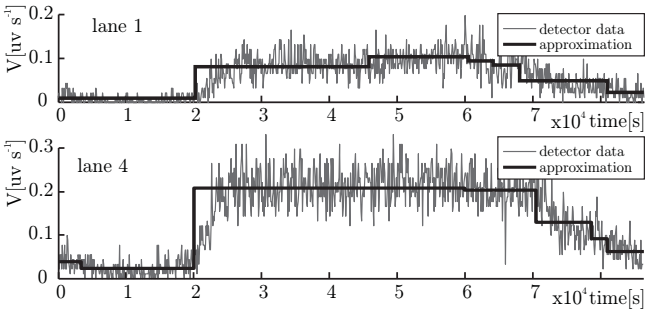


Fig. 6. Real data output from the inductive loop detectors

Inductive loop detectors monitor the traffic conditions on the road for each lane separately. [1] Data from the detectors is produced as time-averaged over a 90s long period. See Fig. 6 for an example of the data from the detectors in unit vehicles per second during a one day period for lane numbers 1 and 4 (other lane detectors produce a similar output). This data from the detectors was subsequently processed by a step-wise filter. The step-wise filter is based on smoothing and produces a constant level of the output data during the variable time interval (see the "approximation" line in Fig. 6). Data from this filter is used as the input for the simulation during one day.

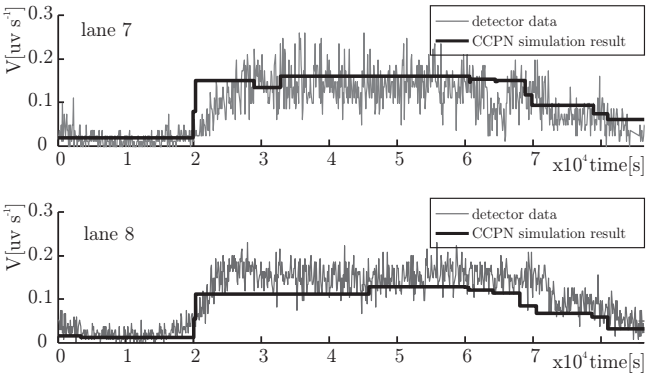
The intersection model uses the same principle described in Section 3. The parameters are summarized in Table 2. The places which are intend for *vehicles waiting in the queue before the intersection* (see to part A in Fig. 6) are connected by the arc from the new *source transitions*. Each source transition is connected to just one place. The maximal speed of the source transition is set to the value given by the approximation for the correspondent lane. Thus, the intersection places are supplied by the speed which is equivalent to the real data measured in the streets. Note that the data from the step-wise filter is not constant during the whole simulation time. From this point of view, the simulation time must be split into several shorter intervals during which the data is constant. These intervals are simulated separately and the simulation result of the first interval is used as an initial condition for the second interval and so on for the whole simulation time.

Table 2. Intersection parameters

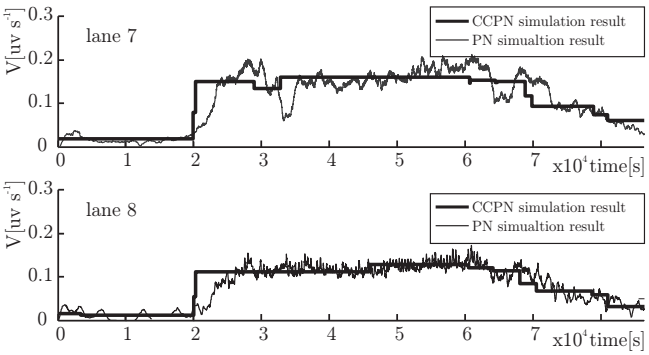
s	d	$\alpha_{s\to d}$	$w_{s\to d}$ [km/h]	$t_{s\to d}$ [s]	$d_{s\to d}$ [s/uv]	$V_{s\to d}$ [uv/s]	$V_{s\to}$ [uv/s]
1	10	0.21	26	20	0.69	0.36	0.58
1	7	0.79	50	20	0.36	0.69	
2	6	1.0	50	20	0.36	0.69	0.69
3	9	1.0	50	52	0.36	1.81	1.81
4	8	0.77	50	52	0.36	1.81	1.81
4	7	0.23	28	47	0.64	0.91	
5	6	1	28	47	0.64	0.91	0.91

The result of the simulation is shown in Fig. 7. There are two representative outputs from lanes 7 and 8 in the

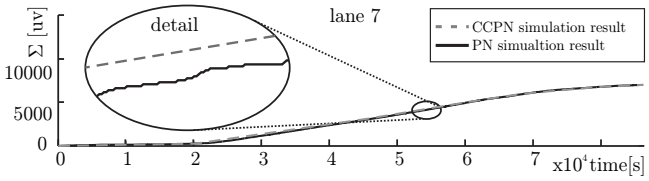
[1] The detectors distance from the intersection is 35m for the input lanes and 50m for the output lanes. This is a sufficient distance to minimize the effect of the intersection control to continuity flow.



(a) Real data from detector and the CCPN simulation



(b) Data from the CCPN simulation and the discrete PN simulation



(c) Sum of all the vehicles during the simulation time

Fig. 7. Prague intersection simulation results

unit vehicles per second. The simulation results for the other lanes are similar, hence they are not shown. Fig. 7(a) shows the real data from the detector (thin line) versus the result of the CCPN model simulation (thick line). Fig. 7(b) shows the result of the CCPN model simulation versus the result of the discrete PN model (thin line). The simulation based on the discrete PN uses the same real data which is used for preprocessing by the step-wise filter described above. The parameters of both Petri net models are the same. The delay time for the discrete PN model is taken from the column $d_{s\to d}$ of Table 2. The figures show that the simulation model corresponds to the real system.

The quality of the simulation can be analyzed from Fig. 7(c), where the sum of all the vehicles in lane 7 during the simulation time is shown. We can see that the number of vehicles which flow through the intersection is similar for both models. The model based on the CCPN computes its state continuously in contrast to the discrete PN model where each vehicle in the intersection initiates the state update, implies the computation performance. The total real simulation time for the model based on the discrete PN is 4927s and the model uses 149523 states from the state space. On the other side, the total real simulation time for the model based on the CCPN is 2s and model uses 34 states. We used standard PC (Intel Core2 CPU T7200 @2.00 GHz, 4GB RAM) for the simulations.

5. CONCLUSION AND FUTURE WORK

In the paper, a new method for conflict resolution in CCPN has been presented. The conflict resolution method is based on the maximal speed proportion. Next, the iterative algorithm for this problem and its solution by LP was shown. LP gives us an effective method to solve complex Petri net models.

Further more, the described method for conflict resolution was used in a light controlled intersection model based on CCPN. The model describes the traffic flow from the macroscopic point of view. This model is innovative, firstly, by the free space modeling together with the opposite direction of the vehicular flow. Secondly, by the constant speed continuous Petri net use only, i.e. without a discrete part for intersection control. The performance evaluation shows that the real time of the simulation is much better than for an equivalent model based on the discrete PN. The accuracy of the simulation results was successfuly compared with real data from traffic in Prague.

The current work is aimed at incorporating several intersections into a traffic region model. As future work, we would like to include additional practical constraints to the problem (e.g., supervisory systems performing high-level optimization on the model).

REFERENCES

Castillo, D.J.M. (1996). A car-following model based on the Lighthill-Whitham theory. In J. Lesort (ed.), *Proceedings of the 13th International Symposium of Transportation and Traffic Theory*, 517–538.

David, R. and Alla, H. (1998). A modeling and analysis tool for discrete events systems: continuous petri net. *Performance Evaluation*, 33, 175–199.

David, R. and Alla, H. (2001). On hybrid petri nets. *Discrete Event Dynamic Systems*, 11(1-2), 9–40.

David, R. and Alla, H. (2004). *Discrete, Continuous, and Hybrid Petri Nets*. Springer.

Di Febbraro, A. and Giglio, D. (2006). Urban traffic control in modular/switching deterministic-timed petri nets. In *11th IFAC Symposium on Control in Transportation Systems*.

Di Febbraro, A., Giglio, D., and Sacco, N. (2001). Modular representation of urban traffic systems based on hybridpetri nets. In *Intelligent Transportation Systems*, 866–871. Oakland, CA, USA.

Di Febbraro, A., Giglio, D., and Sacco, N. (2004). Urban traffic control structure based on hybrid petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 5, 224–237.

Gazis, D.C. (2002). *Traffic theory*. Kluwer Academic Publishers.

Hanzálek, Z. (2003). Continuous petri nets and polytopes. In *IEEE International Conference on Systems Man & Cybernetics, Washington, D.C.*

Júlvez, J. and Boel, R. (2005). Modelling and controlling traffic behaviour with continuous petri nets. In *16th IFAC World Congress 2005*. Elsevier.

Kutil, M., Hanzálek, Z., and Cervin, A. (2006). Balancing the waiting times in a simple traffic intersection model. In *11th IFAC Symposium on Control in Transportation Systems*.

Masukura, S., Nagatani, T., Tanaka, K., and Hanaura, H. (2007). Theory and simulation for jamming transitions induced by a slow vehicle in traffic flow. *Physica A: Statistical Mechanics and its Applications*, 379, 263–273.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., and Wang, Y. (2003). Review of Road Traffic Control Strategies. *Proceedings of the IEEE*, 91(12), 2043–2067.

Tolba, C., Lefebvre, D., Thomas, P., and Moudni, E.A. (2005). Continuous and timed petri nets for the macroscopic and microscopic traffic flow modelling. *Simulation Modelling Practice and Theory*, 13(5), 407–436.

Tolba, C., Thomas, P., ElMoudni, A., and Lefebvre, D. (2003). Performances evaluation of the traffic control in a single crossroad by petri nets. In *IEEE Emerging Technologies and Factory Automation*.

Tolba, C. (2001). Continuous petri nets models for the analysis of traffic urban networks. In *Proceedings of IEEE Systems, Man, and Cybernetics Conference, Arizona, USA*, 1323–1328.

Wang, H., List, G.F., and Dicesare, F. (1993). Modeling and Evaluation of Traffic Signal Control Using Timed Petri Nets. In *IEEE International Conference on Systems, Man and Cybernetic*.