



HAL
open science

P Systems for Traffic Flow Simulation

Jiří Dvorský, Zbyněk Janoška, Lukáš Vojáček

► **To cite this version:**

Jiří Dvorský, Zbyněk Janoška, Lukáš Vojáček. P Systems for Traffic Flow Simulation. 11th International Conference on Computer Information Systems and Industrial Management (CISIM), Sep 2012, Venice, Italy. pp.405-415, 10.1007/978-3-642-33260-9_35 . hal-01551745

HAL Id: hal-01551745

<https://inria.hal.science/hal-01551745>

Submitted on 30 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

P Systems for Traffic Flow Simulation

Jiří Dvorský²¹, Zbyněk Janoška¹, and Lukáš Vojáček²

¹ Department of Geoinformatics, Palacký University,
Třída Svobody 26, 771 46, Olomouc, Czech Republic
`jiri.dvorsky@upol.cz, zbynek.janoska@cdv.cz`

² Department of Computer Science,
VŠB – Technical University of Ostrava, 17. listopadu 15,
708 33 Ostrava, Czech Republic
`lukas.vojacek@vsb.cz`

Abstract. Membrane computing is an emergent branch of natural computing, taking inspiration from the structure and functioning of a living cell. P systems, computing devices of this paradigm, are parallel, distributed and non-deterministic computing models which aim to capture processes taking place in a living cell and represent them as a computation. In last decade, a great variety of extensions of model, introduced by Paun in 1998, were presented. In this paper, we focus on modelling the traffic flow by the means of P systems. P systems enable mezosopic representation of traffic flow with individual modelling of each cars behaviour. Theoretical model is presented together with an XML scheme to store the output of the model.

Keywords: membrane computing, P systems, traffic flow, XML

1 Introduction

Membrane computing represents new and rapidly growing branch of natural computing, which starts from observation that the processes taking place in a living cell can be understood as a computation. Membrane computing and its computational device – *P system* – were introduced by Păun [9] and gained a lot of interest in last decade. P systems start from observation, that membrane plays a fundamental role in the functioning of a living cell. Membranes act as three-dimensional compartments which delimit various regions of a living cell. They are essentially involved in a number of reactions taking place inside cell and moreover act as selective channels of communication between different compartments of a cell [3].

P systems take inspiration from cell on two levels – the structure and the functioning. Structure of cell is represented by its membranes and functioning is governed by biochemical reactions. Every P system therefore has three main elements: a *membrane structure*, where *object* evolve according to given *evolution rules* [11]. Some authors add fourth basic element of membrane systems – *communication* [3, 10]. Communication is always encoded in rules (they are called

communication rules instead of *evolution rules*) and will be dealt with later in the text. From the point of view of transportation modelling, communication (e.g. topology) is essential feature.

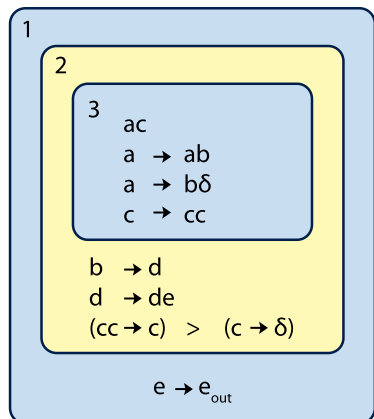


Fig. 1. Graphical representation of P system, [17]

Simple example of P system is depicted in Fig. 1. Membrane structure is hierarchically arranged set of membranes, contained in a distinguished outer membrane, called *skin* membrane. System is surrounded by the *environment*, which may collect objects leaving the system, or in some variants of P systems, the environments can actively support system with objects [2, 4]. Membranes delimit *regions*, with which they are in one-to-one relation. Therefore the terms *membrane* and *region* are mostly interchangeable. Each membrane is identified by its *label*, which can be with membranes in one-to-many relation. The position of inner membranes does not matter; we assume, that in membrane *there is no ordering, everything is close to everything else* [11].

Second basic element of P systems are *objects*. By objects in biological sense are meant chemicals, ions, molecules etc. Those substances are present in a cell in enormous amounts, but the ordering again does not matter. What matters is the concentration, the population, the number of copies of each molecule [11]. Abstracting from biological reality, we represent each substance by a symbol from given alphabet and since the multiplicity matters, instead of objects we use *multisets* of objects. Common notation of multisets in P systems is following: if, for example, objects a, b, c are present in 7, 2 and 5 copies, they will be represented by multiset $a^7b^2c^5$.

In basic variant of P systems, multisets of objects are considered to be floating in inner regions of membrane systems. They evolve by the means of *evolution rules*, which are localised with the regions of the membrane structure. There are three main types of rules [11]: (1) multiset-rewriting rules, (2) communication rules and (3) rules for handling membranes. In this section only first type of rules will be described.

Multiset-rewriting rules take form $u \rightarrow v$, where u and v are multisets of objects. For example, rule $ab \rightarrow cd^2$ says, that one copy of a and one copy of b are consumed and one copy of c and two copies of d are produced. A number of possible extensions of rules exists.

Two crucial features of P systems have to be mentioned at this point. As mentioned earlier, in membranes everything is close to everything else. Therefore, if one instance of an object can be processed by two or more rules, the rule to

be applied is chosen *non-deterministically*. All rules have the same probability to be chosen. The rules also have to be used *in maximally parallel manner*.

More specifically, the objects are assigned to rules, non-deterministically choosing the objects and the rules, until no further assignment is possible. An evolution step in a given region of membrane system consists of finding the maximal applicable multiset of rules, removing from region all objects specified in the left hand of the chosen rules and producing the objects on the right hand side of the rules.

After giving short introduction to basic notions of P systems, let us continue with more detailed survey on traffic flow modelling and representation of transportation system by the means of P systems. In Sect. 2, a definition of basic model – transitional P system, is given. Sect. 3 presents possible extensions of such model, which are incorporated into a model, described in Sect. 5. Sect. 4 gives a short introduction into the traffic flow modelling. Finally, an XML scheme designed to store the configuration of developed system is presented in Sect. 6. We will conclude with some final remarks in Sect. 7.

2 Transition P system

P systems based on application of *multiset-rewriting rules* are called *transition P system*. Formally, transition P system is a construct of the form:

$$\Pi = (O, C, \mu, w_1, w_2, \dots, w_m, R_1, R_2, \dots, R_m, i_o), \quad (1)$$

where:

- O is the finite and non-empty alphabet of objects,
- $C \subset O$ is the set of catalysts,
- μ is a membrane structure, consisting of m membranes, labeled $1, 2, \dots, m$; one says, that the membrane structure, and hence the system, is of degree m ,
- w_1, w_2, \dots, w_m are strings over O representing multisets of objects present in regions $1, 2, \dots, m$ of membrane structure,
- R_1, R_2, \dots, R_m is finite set of evolution rules associated with regions $1, 2, \dots, m$ of membrane structure,
- i_o is either one of the labels $1, 2, \dots, m$ and then the respective region is the *output region* of the system, or it is 0 and then the result of the computation is collected in the environment of the system.

A sequence of transitions of P system constitutes a *computation*. A computation is successful if it halts, it reaches a configuration where no rule can be applied to the existing objects, and output region i_o still exists [11].

The rules are of form $u \rightarrow v$, where $u \in O$ and $v \in (O \times Tar)$, where $Tar = \{here, in, out\}$. Target indications Tar extend transition P system in following way: rule $ab \rightarrow c_{here}d_{in}e_{out}$ consumes one instance of each a and b and produces one copy of c in current membrane, one copy of d in a child of current membrane and one copy of e in the parent of current membrane. If current

membrane is skin membrane, object e is send to environment of the system. If current membrane does not have a child, rule can not be applied.

Another extension comes from the existence of *catalysts*. Catalysts are objects, which participate in a chemical reaction, but are not consumed or produced by it. They just enable the application of rule. Rule with catalysts takes following form: $ac \rightarrow bc$, with object c being the catalyst.

3 Possible Extensions of P systems

In this section we will mention some elementary extensions of P system, which however constitute only a fracture of possibilities. We refer reader to The P systems Webpage [16] for complete list of publications and further information. Already in the text, three types of rules were mentioned. Evolution rules were briefly covered in previous sections.

Communication rules were introduced in [10]. Basic idea of communicating P systems is, that computation is achieved only by transporting object between membranes. Direct inspiration from biology are *symport* and *antiport*. When two chemical pass through membrane only together, in the same direction, the process is called *symport*. When the two chemicals pass only with help of each other, but in opposite directions, the process is called *antiport* [10]. Symport rules take a form (ab, in) or (ab, out) , and antiport rules take a form $(a, out; b, in)$, where a, b are object from alphabet of all possible objects. Meaning of rules is following: for symport rule (ab, in) or (ab, out) , if objects a, b are present in current membrane, they are sent together into child (or parent, in second case) of the membrane. For antiport rule $(a, out; b, in)$, if a is present in current membrane and b is present in the parent of a membrane, than a exits current membrane and b enters it. Universality of P systems with symport and antiport have been proven [10] and simplified version of communication, *conditional uniport* have been studied [14]. Comprehensive review of communication strategies in P systems can be found in [15].

Third type of rules are *rules for handling membranes*. Dissolution of membranes has already been mentioned, but other ways to obtain dynamical membrane structure, evolving during the course of computation, have been presented. Most simple of those is assigning *electrical polarization* $+, -, 0$ to each membrane. Polarization replaces target indicators *in, out, here*. Polarization of objects is introduced by the rules and polarized objects can enter only membranes with opposite polarization. For example, $ab \rightarrow c^+d^-$ means, that one instance of c enters inner membrane with negative polarization and one instance of d enters inner positive membrane. Rules can also be used to change the polarization of membranes during the computation.

Two issues seem essential, when P systems are used to simulate real-world phenomena rather than for computation. Non-determinism is first of the issues. Some chemical reactions are more likely to occur than others. First attempt to solve this is by assigning priorities to rules. Firstly, set of rules with the highest priority is chosen and according to the principle of maximal parallelism, all rules

which can be applied, are applied. Then, the rules with second highest priority are selected and the procedure repeats, etc.

Second approach is to assign probabilities to all rules. Probability can be introduced to P system on different levels [8], but here we will mention only probability on the level of rule selection. Different approaches have been proposed [2, 6, 12]. Basic idea is to associate each rule with a constant k , so the rule takes a form $u \xrightarrow{k} v$, where u, v are multisets of objects and k can be interpreted either as a probability, or as a “stoichiometric coefficient”, using which the true probability is calculated.

Last extension, which we will mention at this point, is representation of time of P systems. In real world, every biochemical reaction takes some time. Representation of time in P systems is similar to representation of probability. A constant t is assigned to each rule, so the rule takes the form $u \xrightarrow{t} v$, where u, v are multisets of objects and t is number of time units, which must pass to complete the application of the rule [5]. In the first time step, multiset u is consumed and removed from the current membrane. After $t - 1$ more time steps, multiset v is introduced into the system. Time can also be introduced into P systems as a lifetime of objects or even membranes [1].

For the sake of brevity, we will not discuss more extensions of P systems, although many possibilities were explored within this framework. P system based model of traffic flow will take advantage of three main features mentioned in this section – membrane polarization, probability and time-dependence.

4 Traffic flow simulation

Hoogendoorn and Bovy [7] distinguish three main levels of modelling of transportation systems:

- **microscopic simulation model** describes both the space-time behaviour of the systems’ entities (i.e. vehicles and drivers) as well as their interactions at a high level of detail (individually). Sometimes submicroscopic models are mentioned separately.
- **mesoscopic model** does not distinguish nor trace individual vehicles, but specifies the behaviour of individuals, for instance in probabilistic terms. To this end, traffic is represented by (small) groups of traffic entities, the activities and interactions of which are described at a low detail level.
- **macroscopic flow model** describe traffic at a high level of aggregation as a flow without distinguishing its constituent parts. For instance, the traffic stream is represented in an aggregate manner using characteristics as flow-rate, density, and velocity. Individual vehicle manoeuvres, such as a lane-change, are usually not explicitly represented [7].

While microscopic models describe accurately behaviour of small-scaled systems such as lanes or intersections, their use to simulate spatially extensive areas, such are cities or agglomerations, are limited. Macroscopic models, on the other

hand, are capable of simulation on low-scale level, but lack detailed description of individuals. Mesoscopic models are a reasonable compromise between detail and robustness, which is desirable for certain applications, such as public transport modelling [13].

In this study we propose a simulation model for traffic flow, aiming to describe accurately vehicle behaviour at the intersections, but ignoring detailed description of cars behaviour at road segments between intersections. Such model should be individual based, representing vehicles as agents instead of populations, but at the same time should be robust enough to enable simulation of large-scaled road networks. Such model could be used for example for optimization of traffic lights at signalized intersections.

5 P system based model of traffic flow

In this section we describe P system model of traffic flow. This model aims to capture detailed behaviour of cars at the intersection, while ignoring behaviour at road segments.

5.1 Definition of a system

Within P systems paradigm, road network can be described as a graph with membranes at the nodes. This representation is similar to the one of communication P systems [15]. There are three types of membranes, based on their function. *White holes* act as generators of cars. If the system represents, say, a city, than white holes will be the roads, bringing cars to the city. Opposite reactions take place in *black holes*, where the cars leave the system. White and black holes does not necessarily have to be only at the periphery of the system – also residential, industrial or business areas within the city can act as white or black holes, and they can therefore be placed arbitrarily in the graph. The third type of nodes is an *intersection*. At intersection, cars are distributed to other parts of the network.

Within this paradigm, cars are represented as objects. For the sake of brevity, we will consider only one type of object – an arbitrary vehicle. In praxis, however, different types of vehicles (cars, buses, motorcycles etc.) can be represented by different objects.

The behaviour of objects is described by a set of rules. In classical P systems, rules enable objects to be created, destroyed or qualitatively changed. In the proposed system, each of these is encapsulated in one type of membranes. In white holes, objects are created, in black holes, they are removed from the system, and at the intersections, they are changed (meaning they can change their state from moving to stopping or vice versa). At the intersections, objects are communicated to other membranes in the system.

Clearly, the behaviour of cars is not strictly deterministic, therefore stochasticity must be introduced at some point. There are two ways to represent the behaviour of a car in the network.

- When an object is introduced to the system at white hole A , a target destination B is chosen from set of available black holes. The shortest path between nodes A and B is calculated and vector of intersection, which must be passed by an object is assigned to it. For each white hole, a set of probabilities is assigned to all available black holes. Therefore, some paths are preferred by the cars then others.
- For each intersection, a set of probabilities is assigned to all intersecting roads. When a car is present at the intersection, a road is chosen and car is send to next node of a graph. Each road is associated with a certain probability of being chosen, therefore some links are more preferred by cars than others.

First approach is preferable in case, where the preferences of cars in the network are well known. If we lack the knowledge about drivers preferences and know only vehicle intensities at the roads, second approach seems more suitable.

Also, travel time have to be taken into account. Time necessary to travel from one node of a graph to another is represented by a cost of the link. Graph is oriented, therefore travel time between two nodes can differ in both directions.

Last issue is associated with representation of traffic lights. At intersection, only cars approaching from a certain direction are allowed to pass. Within P systems framework, membrane polarizations are suitable tool for selective allowance of objects into a membrane. The polarization of each intersections represents red/green lights and periodically changes, therefore simulating real world situation.

5.2 Formal definition of a model

Traffic P system is a construct:

$$Traffic\Pi = (O, \mu, syn, (s_{(i,j,t)})_{(i,j) \in syn; t \in \mathbb{N}}, R), \quad (2)$$

where:

- $O = \{veh, veh_s\}$ is an alphabet of objects, veh representing moving vehicle and veh_s representing stopping vehicle,
- μ is a membrane structure, consisting of m membranes, labeled WH, BK, INT ; representing white holes, black holes and intersections,
- $syn \subseteq \{(i, j, t) | i, j \in \{0, 1, 2, \dots, m\}, i \neq j, t \in \mathbb{N}\}$ is a subset of synapses – links between the membranes, where i and j are membranes from the set μ and t is time constant associated with each synapse.
- R is a set of rules associated with membranes from μ . The rules differ for membranes labeled as white holes, black holes and intersections.

- White holes – vehicles veh are generated and sent to one of neighboring membranes.

$$[]_{WH} \xrightarrow{t_i} [veh^{WH}]_k,$$

where $\{k, WH, t_i\} \in syn$ and application of a rule takes t_i time steps. A label of white hole will be associated with a vehicle veh when approaching intersection k , which will be used to decide, whether a vehicle should stop or pass the intersection.

- Black holes – vehicles veh are removed from the system.

$$[veh]_{BH} \xrightarrow{t=1} []_{WH},$$

application of such rule takes 1 time unit.

- Intersection – several rules are associated with intersections. Here we describe a situation, where car makes a decision about its target destination at each intersection.

Rule:

$$[veh^k]_m \xrightarrow{t_i, p_i} [veh^m]_n$$

describes a situation, where car coming from membrane k approaches an intersection m with polarization k . Since the polarization and label of veh correspond, a veh can pass the intersection and proceed to membrane n , with probability p_i . veh will be assigned with label of passed membrane m . Application of rule will take t_i time units.

Rule:

$$[veh^k]_m \xrightarrow{t=1, p=1} [veh_s^l]_m$$

describes a situation, where moving car veh coming from membrane k approaches an intersection m with polarization l . Since the polarization and label of veh do not agree, the vehicle veh must stop and therefore changes to veh_s , with probability 1 and taking 1 time step.

Rule:

$$[veh_s^k]_m \xrightarrow{t_i, p_i} [veh]_n$$

describes a situation, where stopping car veh_s from membrane k is in membrane m with polarization k ; it can pass the intersection and continue to membrane n . This is enabled by changing polarizations of membranes.

Global clock for the system are assumed. The polarization of membranes changes according to a given schedule. All other parameters of the system - creation rate at white holes, probabilities and time constants associated with the synapses can also be time-dependent.

6 XML structure for proposed model

We used XML language for creating topology, creating rules and setting initialization values. The reason why we chose this way is simple, because we can simply and clearly describe all necessary attributes. A code sample is shown in Listing 1.1, where we divide file to 3 parts.

1. First part stores the topology of P-system. Connections between membranes are defined, I/O membranes are chosen etc.
2. Second part describes the initial configuration of membranes.
3. The third part contains the rules for membranes. It consists of left side, right side and additional information about conditions of use, delays, etc. Left side contains items needed for rule to be applied, right side describes the result of rule application.

Listing 1.1. Example XML file

```

<?xml version="1.0" encoding="utf-8"?>
<Program Version="1.0" >
  <Topology>
    <Membrane ID="1" >
      <Type Membrane="White" />
      <LinkTo ID="3" ValuePath="5" />
    </Membrane>
    <Membrane ID="4" >
      <Type Membrane="Black" />
      <LinkTo ID="3" ValuePath="5" />
    </Membrane>
    <Membrane ID="3" >
      <Type Membrane="Green" />
      <PolarizationComing IDs="1,4" Time="10" />
      <PolarizationComing IDs="2,5" Time="5" />
      <LinkTo ID="2" ValuePath="10" />
      <LinkTo ID="4" ValuePath="5" />
      <LinkTo ID="5" ValuePath="5" />
    </Membrane>
  </Topology>
  <Initialization >
    <Membrane ID="1" >
      <Item NameID="a" Value="0" />
      <Item NameID="b" Value="0" />
    </Membrane>
  </Initialization >
  <Rules>
    <Selected>
      <Membrane ID="3" >
        <LeftSide>
          <Item NameID="a" Value="1" />
        </LeftSide>
        <RightSide>
          <Item NameID="a" Value="1" />
        </RightSide>
        <Additional>
          <Probability Value="0.5" />
          <DuelID Value="4" />
          <Polarization Value="true" />
        </Additional>
      </Membrane >
    </Selected>
  </Rules>
</Program >

```

```
</Additional>
</Membrane>
</Selected>
<Default>
  <Membrane Typ="White">
    <LeftSide>
      <Item NameID="a" Value="1" />
    </LeftSide>
    <RightSide>
      <Item NameID="a" Value="1" />
    </RightSide>
  </Membrane>
</Default>
</Rules>
</Program>
```

7 Concluding Remarks and Future Work

A P system for traffic flow simulation was defined and an XML scheme designed to store the configuration of such model was presented. A basic variant of P system was described with additional features – stochasticity, time-dependence, selective membrane permeability – described in more detail.

Presented model was designed to simulate the behaviour of vehicles at the intersections and enables elegant simulation of cars behaviour at the intersections, while ignoring detailed description of behaviour at road segments. This robustness is desirable i.e. for traffic light optimization.

On the webpages of the authors is possible to find the application, which we develop and use for experiments. (<http://wh.cs.vsb.cz/voj189>)

In the future, the presented model will be further developed to be able to describe the traffic system in more realistic manner – cars delays, complex intersections and destination preferences are of primal interest.

Acknowledgement

This work was supported by SGS, VŠB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing. This work was also supported by SoftComp project (CZ.1.07/2.3.00/20.0072). SoftComp project is co-nanced by ESF and Czech state budget.

References

1. Aman, B., and Ciobanu, G. Adding Lifetime to Objects and Membranes in P Systems. *International Journal of Computers Communications and Control* 5 (3). 2010. Pages 268-279.
2. Bernardini, F., and Manca, V. Dynamical aspects of P systems. *BioSystems* 70 (2). 2003. Pages 85-93.

3. Bernardini, F., and Gheorghe, M., and Krasnogor, N., and Muniyandi R.C., and Perez Jimenez, M.J., and Romero-Campero, F.-J. On P Systems as a Modelling Tool for Biological Systems. *Pre-Proc. of the sixth Workshop on Membrane Computing*. Vienna, Austria, 2005. Pages 114-133.
4. Cavaliere, M. Evolution-Communication P Systems. Proceeding WMC-CdeA '02 Revised Papers from the International Workshop on Membrane Computing. 2003. Pages 134-145.
5. Cavaliere, M., and Sburlan, D. Time-independent P systems. *Membrane Computing. International Workshop WMC5*. Milano, Italy, 2005. Pages 239-258.
6. Cazzaniga, P., and Pescini, D., and Romero-Campero, F.-J., and Besozzi, D., and Mauri, M. Stochastic Approaches in P Systems for Simulating Biological Systems. *Fourth Brainstorming Week on Membrane Computing*. Seville, Spain, 2006. Pages 145-164.
7. Hoogendoorn, S.P., and Bovy, P.H.L. State-of-the-art of Vehicular Traffic Flow Modelling. *Delft University of Technology, Delft, The*. 2001. Pages 283-303.
8. Obtulowicz, A., and Păun, Gh. (In search of) Probabilistic P systems. *Biosystems* 70 (2). 2003. Pages 107-121.
9. Păun, Gh. Computing with Membranes. *Technical report, Turku Center for Computer Science-TUCS*. Turku, Finland, 1998.
10. Păun, A., and Păun, Gh. The Power of Communication: P Systems with Symport/Antiport. *New Generation Computation* 20 (3). 2002. Pages 295-306.
11. Păun, Gh. Introduction to Membrane Computing. *First brainstorming Workshop on Uncertainty in Membrane Computing*. Palma de Mallorca, Spain, 2004. Pages 1-42.
12. Pescini, D., and Besozzi, D., and Mauri, G., and Zandron, C. Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science* 17 (1). 2006. Pages 440-447.
13. Peeta, S., and Ziliaskopoulos, A.K. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics* 1. 2001. Pages 1233-1265.
14. Verlan, S., and Bernardini, F., and Gheorghe, M., and Margenstern, M. Computational completeness of tissue p systems with conditional uniport. *Proceedings of the 7th international conference on Membrane Computing*. Springer-Verlag, 2006, Berlin/Heidelberg. Pages 521-535.
15. Verlan, S., and Bernardini, F., and Gheorghe, M., and Margenstern, M. Generalized communicating P systems. *Theoretical Computer Science* 404 (1-2). 2008. Pages 170-184.
16. The P Systems Web Page. <http://ppage.psyste.ms.eu/>. Last revision: 2012-03-18.
17. P system - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/P_system. Last revision: 2012-03-18.