Petri net with conflicts and (max, plus) algebra for transportation systems

Ahmed Nait Sidi Moh, Marie-Ange Manier, Abdellah El-Moudni, Maxime Wack

PETRI NET WITH CONFLICTS AND (MAX, PLUS) ALGEBRA FOR TRANSPORTATION SYSTEMS

A. Nait-Sidi-Moh, M.-A. Manier,
A. El Moudni and M. Wack

Laboratory Systèmes et Transports,
Université de Technologie de Belfort-Montbéliard,
Site de Belfort 90010 Belfort cedex, France.
Tel : (+33)3 84 58 30 75 Fax : (+33)3 84 58 33 42
E-Mail: ahmed.nait-sidi-moh@utbm.fr

Abstract: In this paper, solving of conflicts of a Petri net model with techniques of (max, plus) algebra is considered. We define a routing policy which enables to solve and arbitrate the associated conflicts with a Petri net. We show how the conflict solving semantic prevents the deadlock in a graphical model while introducing routing functions into modelling. To illustrate the proposed results, a public transportation network is worked out. The aim is to analyze and evaluate the performance of a bus network which is represented by a Petri net with conflicts and a state model in (max, plus) algebra.

Keywords: Transportation system, Modelling, Performance evaluation, Petri Nets, (max, plus) algebra, Routing functions.

1. INTRODUCTION

Many systems may be described through a discrete event dynamic model, like transportation networks, manufacturing facilities, or computer networks (Cohen *et al*, 1985; Nait-Sidi-Moh, 2003; Olsder *et al*, 1998, Goverde *et al*, 1998). They differ from continuous systems as they can be associated with a space of discrete state, and changes of state involved by some events. Their behaviour is mainly characterized by parallelism, synchronization and concurrency (Baccelli *et al*, 1992; Gaubert, 1992).

Among the formalisms used to model the discrete event systems (DES, for short), Petri Nets (PN) are an interesting graphical and mathematical tool (David *et al*, 1992; Proth *et al*, 1995). Nevertheless they sometimes provide complex representations because of the complexity of the modelled systems (problem size,…). Petri Nets include several modelling families. Each one represents a particular aspect of a complex system. For example, it is well known that event graphs allow us to model phenomena like parallelism, concurrency and synchronization. Another class of Petri nets is the Free Choice Petri Nets (FCPNs) (Bouillard, 2005; Sgroi *et al*, 1999). This last class may be used for the reliability, the management and the schedule of tasks. It also may be integrated in a global methodology aiming at designing the systems. Also, we can quote the Petri Nets with conflicts (PNConf) which exhibits clear distinction between the notions of concurrency and choice (David *et al*, 1992; Yen, 2002). In this article, we specifically use this kind of Petri Nets. It is appropriate to obtain a model in which the outcome of a choice depends on the arrival time of token rather than other parameters.

With a PNConf model, one of the main encountered problems is to know which transition will be fired in case of conflict (which involves making choices). Then an adapted strategy must be defined to solve those conflicts. More precisely, a routing policy must be defined when several output transitions exist for a given place. It determines which transition will be fired by a token arriving in a such place. Several routing policies may be envisaged such as race policy, Bernoulli routing, periodic routing, extreme routing (Bouillard, 2005; Nait-Sidi-Moh, *et al.*, 2003).

(Nait-Sidi-Moh, *et al.*, 2003) deals with the use of periodic routing to solve structural conflicts identified in a Petri Net model. This model represented a part of a public transportation network. The problem studied was the evaluation of the connection times during a given travel of passengers. The system behaviour was supposed to be a periodic one (periodic arrival of vehicles at the considered stops). For this study, the routing policy allowed us first to arbitrate those conflicts, and then to solve a (Max, plus) linear state model deduced from the graphical representation. To reach these goals, the explicit expressions of routing functions were determined. After solving *a priori* the structural conflicts, the performance evaluation was made, in terms of passenger waiting times at each connection stop and travel time from a given origin to a given destination point.

This paper deals with an extension of the study presented in (Nait-Sidi-Moh, *et al.*, 2003) to solve a more general problem. Indeed, we have relaxed the periodicity constraint. Then a conflict solving semantic must be found which prevents the deadlock in the graphical representation, and which allows a fine analysis so as to optimize the network dynamics.

Among the formalisms used to represent the analytical behaviour of discrete event systems, dioid algebra presents an adequate tool (Baccelli *et al*, 1992; Gaubert, 1992, Olsder *et al*, 1998). With this algebra, the evolution of the system is then described by linear state equations. The interpretation is the following one: each variable is a "dater" in (Max, plus) algebra: each function $x_i(k)$ represents the k^{th} firing date of transition x_i; in (Min, +) algebra, each variable is a "counter": each function $x_i(t)$ represents the number of firings of x_i at time t.

2. MOTIVATION AND PAPER CONTRIBUTION

Let us model the studied system with the Petri Net PN= $\{P, T, A, W, M_0, \triangle\}$ where $\{P, T, A, W, M_0\}$ is the classical Petri Net, and $\triangle = (\triangle_t)_{t\in T}$ is the set of routing functions. The routing strategy proposed in this article is given by the introduction of the routing function $\triangle$ in the modelling. Hereafter, some elements that we will need to use in this paper are defined.

- $\forall$ P$\in$ P, $P^\bullet$ = {t / A(P, t) $\geq$ 1, t$\in$ T} is the set of output transitions of P,
- P_{conf} is a subset of P ($P_{conf} \subseteq P$) such that: $\forall$ P$\in P_{conf}$, $|P^\bullet| > 1$ (each place of P_{conf} has at least two output transitions). A place P such that $|P^\bullet| > 1$ is in **conflict** situation. Each internal transition t of a PNConf verifies $|{}^\bullet t| \geq 1$ (i.e. each internal transition may have more than one input place). It is not the case for Free Choice Petri Nets where each transition has only one input place ($|{}^\bullet t| = 1$).
- For each place P $\in P_{conf}$ and for each transition t $\in P^\bullet$, we define a routing function that we associate with t. This function is defined by :

$$\triangle_t : \mathbb{N}^* \to \{e, \varepsilon\}$$
$$k \mapsto \triangle_t(k)$$

  - $\triangle_t(k) = e$ implies that the transition t is fired by the k^{th} token arriving in P.
  - $\triangle_t(k) = \varepsilon$ implies that the transition t will not be fired by the k^{th} token arriving in P. This token takes part in the firing of another transition t'$\in P^\bullet$.

According to the definition of $\triangle$, we have the following relation : $\forall$ P$\in P_{conf}$, if $\exists$ $t_1 \in P^\bullet$ such that $\triangle_{t1}(k) = e$, then $\forall$ t $\in P^\bullet$ with t $\neq t_1$, $\triangle_t(k) = \varepsilon$, we say that the firing of t_1 is **real**, and the firing of t is **virtual,** we affect then to t(k) a virtual date which is ε (t(k) = ε).

Figure 1 illustrates a way to represent conflicts in the Petri Nets. In this graph, we identify two transitions in conflict situation: x_1 and x_2. The firing of x_1 is preferred to the firing of x_2, on condition that the places P_1 and P_2 simultaneously contain one token. Otherwise, if only P_2 contains one token, x_2 will be fired. Nevertheless, we associate a k^{th} firing to each transition even if only one of them is really fired (introduction of the function $\triangle$). This firing is performed by the k^{th} token arriving in place P_2. If the transition x_2 (respectively x_1) is effectively fired, then a virtual firing of transition x_1 (respectively x_2) is generated at time ε (respectively ε). In the graphical model of the figure 1, real or virtual firings of transitions x_1 and x_2 may be expressed respectively by associated functions $\triangle_{x1}$ and $\triangle_{x2}$. Each function takes part to the proposed solving strategy of conflicts.

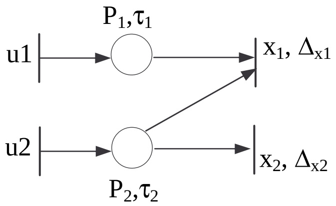

Fig. 1. Timed Petri Net with conflicts.

Remark 1:

- If x_1 is virtually fired then the place P_1 does not contain no token;
- If x_1 is really fired then the place P_1 contains at least one token. In the case where P_1 contains more than one token, the real firing of x_1 withdraws all tokens of P_1. This means, in the bus network, that all batches of passengers waiting at the considered connection stop get on the bus and carry out the connection. In this case, the associated weight to the arc A(P_1, x_1) becomes m(P_1) which is the marking of the place P_1. Let us consider that the capacity of the buses is infinite.

From this remark, figure 2 gives the new PN obtained from the PN of figure 1.

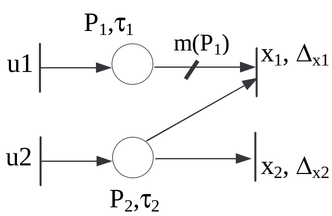

Fig. 2. New Timed PN with conflicts (some weights different from 1).

By taking into account the routing functions $\triangle_{x1}$ and $\triangle_{x2}$, respectively associated with x_1 and x_2, the firing evolution of these two transitions is expressed by the following way.

$$x_1(k) = \max [\tau_1 + u_1(k') + \Delta_{x1}(k),\ \tau_2 + u_2(k) + \Delta_{x1}(k)]$$
$$= \tau_1 \otimes u_1(k') \otimes \Delta_{x1}(k) \oplus \tau_2 \otimes u_2(k) \otimes \Delta_{x1}(k) \quad (1)$$

k' is given according to the marking $m(P_1)$ of place P_1 just before a real firing of x_1.

Since the k^{th} token that fires the transition u_2 takes part in the firing of x_1 (really or virtually) then the expression of $x_1(k)$ can be written as follows:

$$x_1(k) = \tau_2 \otimes u_2(k) \otimes \Delta_{x1}(k) \quad (2)$$

For the k^{th} token arriving in P_2, if x_1 is fired rather than x_2 (i.e. P_1 also contains one token), then we obtain for $\Delta x_1(k) = e$

$$x_1(k) = \tau_2 \otimes u_2(k)$$

Otherwise, $\Delta x_1(k) = \varepsilon$,

$$x_1(k) = \varepsilon$$

In the same way we deduce the evolution equation of the transition x_2. We finally obtain the following expressions:

$$\begin{cases} x_1(k) = \tau_2 \otimes u_2(k) \otimes \Delta x_1(k) \\ x_2(k) = \tau_2 \otimes u_2(k) \otimes \Delta x_2(k) \end{cases} \quad (3)$$

The algorithm given hereafter allows us to solve these equations and then to evaluate the states of the system. We note that this algorithm may be used to solve a system of several equations in the case of a large model which contains several conflicts. In the considered case (figure 1), to use the proposed algorithm, some data must be known: initial conditions $x_1(0)$ and $x_2(0)$, and the various firing times of transitions u_1 and u_2 (labelled $u_1(k)$ and $u_2(k)$).

Algorithm:

% $u_1(k)$ and $u_2(k)$ are known for all k

% $x_1(0) = x_2(0) = \varepsilon$

For k = 1: N

If $m(P) \geq 1 \quad \forall P \in {}^\bullet x_1$

Then $\Delta x_1(k) = e$ and $\Delta x_2(k) = \varepsilon$

$x1(k) = \tau_2 \otimes u_2(k)$

$x_2(k) = \varepsilon$

Elseif $m(P_2) \geq 1$ and $m(P_1) = 0$

Then $\Delta x_1(k) = \varepsilon$ and $\Delta x_2(k) = e$

$x_1(k) = \varepsilon$

$x_2(k) = \tau_2 \otimes u_2(k)$

Else

"Neither x_1 nor x_2 will be fired until arriving of tokens in P_1 and/or P_2"

Endif

Endfor

----

3. THE STUDIED SYSTEM

We illustrate the approach proposed by studying a public transportation network. We consider then a bus network composed of four lines (Figure 3). Each line L_i is linked with at least another lines L_j by the connection stops. Each line L_i is represented by a departure stop (Ds_i), connection stop(s) $Cs_{i,j}$, and an arrival terminus (As_i). Also, each line contains other simple stops that we do not consider here since we are just interested at the connection stops.

For each line L_i ($1 \leq i \leq 4$), the following data are supposed to be fixed:

- the parameter λ_i which is the necessary time (average time) for a bus of line L_i to perform one turn;
- the number of buses circulating on the line L_i;
- the travel times of buses between all network stops. They include the mean time needed for passengers to go up and/or get off the bus and also the time spent at intermediate stops.

We suppose that all buses work independently, i.e. the buses of different lines do not wait for each other; each of them leaves the connection stop just after the passengers got on/off it; this working mode is called *working without synchronization*. It is not the same working in the railway as a train that arrives first at the connection station often has to wait for a connecting train to carry out the connection (Olsder *et al*, 1998, Vries *et al*, 1998).

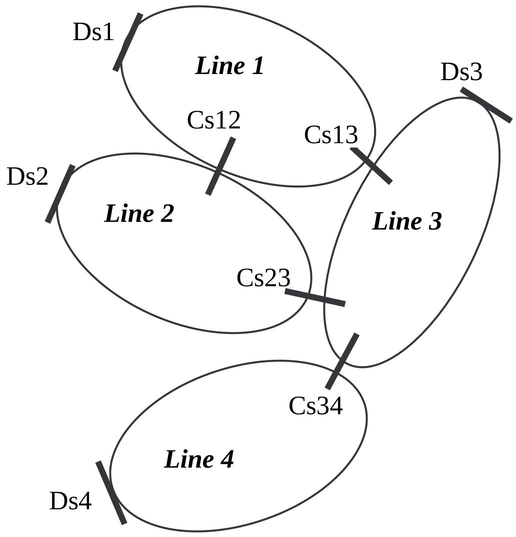

Fig. 3. The studied network.

We assume that the departure times of buses from their departure stops are scheduled according to a given timetable. In this study, we are interested at each connection stop between two lines L_i and L_j to the connection management of two kinds of passengers: passengers who make the connection from L_i to L_j and passengers who move from L_j to L_i.

The Petri net with conflicts model that represents the considered network is given in the following section (figure 4). This model illustrates the network working without any deadlock of buses.

4. APPLICATION TO THE CONSIDERED BUS NETOWRK

4.1 Modeling with Petri net

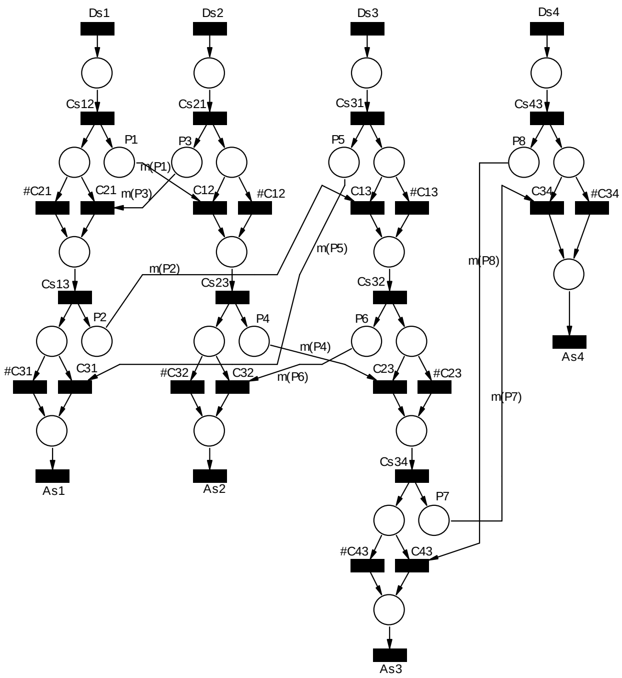

Fig. 4. The Petri net model of the considered network.

Legend:

The various notations of this Petri net are given in the following way: for each i and j ($1 \le i, j \le 4$):

- Dsi: departure station of the line L_i;
- Csij, Csji, Cij and #Cij represent the connection stop between the lines L_i and L_j. Moreover Cij is also associated with the connection carried out from L_i towards L_j ; #Cij is fired if no connection is carried out;
- Asi: arrival stop of the line L_i;
- $m(P_k)$ represents the associated weight to the downstream arc of the place P_k.

The figure 4 describes the behaviour of the considered network (figure 3). In this model, we identify the four lines which are linked by the connection stops. Each line L_i ($1 \le i \le 4$) is represented by a sequence of places and transitions where input transitions (Dsi) model departure stops, internal transitions (Csij, Cij, #Cij) model connection stops and output transitions (Asi) represent arrival stops. In the level of each connection stop, one finds the given structure in figure 2 where the stop is modelled by two transitions Cij (the connection with other line is carried out) and #Cij (the connection did not take place). This situation and the choice of the transition that will be fired involve the conflict in the level of each connection stop. As we already said in the second section, the parameters m(Pi) associated with some arcs model the number of batches of passengers who want to make the connection with another line.

4.2 State representation in (max, +) algebra

In this section, we present a (max, +)-linear model for the considered urban bus network. With this aim, we assign variables to model transitions and temporisations to certain places (necessary times to move from a stop to another) (see figure 5). The obtained model with these new notations is given figure 5.

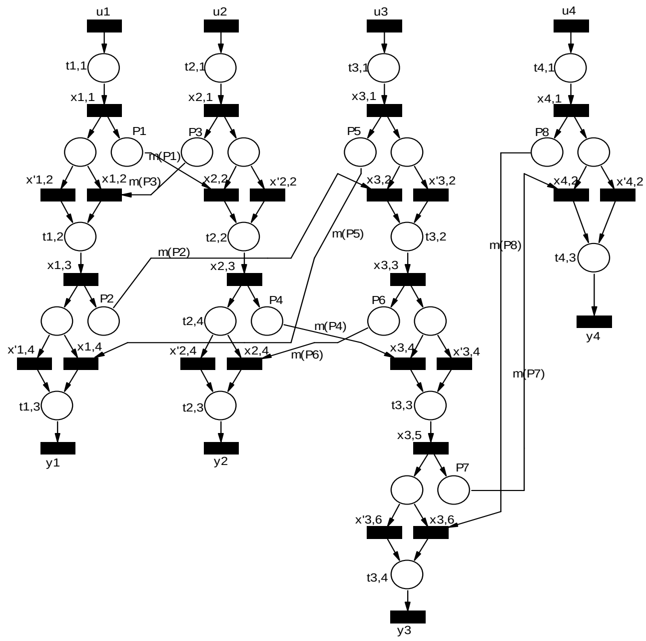

Fig. 5. The Petri net model: definition of variables and temporisations.

Remark 2: even if it does not appear on the graphical model (Fig. 5), we associate a routing function $\Delta x_{i,j}$ (respectively $\Delta x'_{i,j}$) to each transition $x_{i,j}$ (respectively $x'_{i,j}$) in a conflict situation, that represent a connection stop,. These functions enable to arbitrate the conflicts in the PN model.

From this model and by using the theory of linear systems in dioid algebra, we obtain the (max, +)-linear model describing the network. Because of the conflicts associated to the Petri net model, the associated (max, +) model is a non stationary system. This means that, at a given time, the firing numbers of the downstream and upstream transitions of some places are not the same ones.
Hereafter we give the expression of a dater $x_{i,j}(k)$ by using the technique introduced in the equation (1). We consider for example $x_{1,2}(k)$. For all k,

$x_{1,2}(k) = x_{1,1}(k) \otimes \Delta x_{1,2}(k) \oplus x_{2,1}(k_1) \otimes \Delta x_{1,2}(k)$

where k_1 is expressed according to the number of tokens $m(P_3)$ in the place P_3.
The connection will be done just when the bus arrives at the connection stop. Then one can deduce for all k, $x_{1,1}(k) \geq x_{2,1}(k_1)$. So, we obtain:

$x_{1,2}(k) = x_{1,1}(k) \otimes \Delta x_{1,2}(k)$

In the same way, we obtain all equations of the model. The transportation system can be then modelled as a state representation in $\mathbb{R}_{max}$ by:

$$\begin{aligned} X(k) &= A \otimes \Delta \otimes X(k) \oplus B \otimes U(k) \\ Y(k) &= C \otimes X(k), \end{aligned} \qquad (4)$$

$X(k) \in \mathbb{R}^{24}_{max}$ is the state vector that gathers all defined internal variables $x_{i,j}(k)$. $x_{i,j}(k)$ denote the departure times of buses at stops $x_{i,j}$. $U(k) \in \mathbb{R}^{4}_{max}$ is the input vector whose components are the input variables $u_i(k)$. $u_i(k)$ denote the departure times of buses at their departure stop, $Y(k) \in \mathbb{R}^{4}_{max}$ is the output vector which gather the output transitions $y_i(k)$ denote the arrival times of buses at arrival stations. $A \in \mathbb{R}^{24x24}_{max}$, $B \in \mathbb{R}^{24x4}_{max}$ and $C \in \mathbb{R}^{4x24}_{max}$ are the characteristic matrices whose components are the moving times on the network. $\Delta \in \mathbb{R}^{24x24}_{max}$ is the routing matrix whose components are the routing functions.

4.3 Timetable evaluation problem

In this paper, we focus our study on the use of routing functions to evaluate the transit times of buses at the network stops, the waiting times of passengers at the connection stops and the journey times of passengers. In the same context, similar studies are already achieved using the theory of (max, +) algebra (Nait *et al*, 2006, Houssin *et al*, 2006).

In this section, we present the timetable evaluation problem by solving the state model (4). This evaluation problem will be focused on the determination of various daters related to each stop of the network. To reach this objective, we define, for each step, the elements of the routing matrix Δ. For each step k, the value of each routing function $\Delta x_{i,j}(k)$ (elements of the matrix Δ) is defined observing the marking of each place of the model. The application of the algorithm given in the section 2 for all transitions in conflict situation allows us to solve the equations of the (max, +)-linear system (4).

By applying the proposed algorithm, the obtained results for the two transitions $x_{i,j}$ and $x'_{i,j}$ (connection stop with or without connection between buses) may be expressed as follows: for example, for $1 \leq k \leq N_{max}$, (N_{max} the number of journeys to perform by a bus), if $N_{max} = 10$,

$x_{i,j} = \{v1, v2, \varepsilon, v4, \varepsilon, \varepsilon, v7, v8, \varepsilon, v10\}$

$x'_{i,j} = \{\varepsilon, \varepsilon, v3, \varepsilon, v5, v6, \varepsilon, \varepsilon, v9, \varepsilon\}$

From the values of $x_{i,j}$ we remark that only six components are different from zero (ε) which means that six connections are carried out. For other components, which take the values "ε," the connection is not really carried out (we say that the connection is virtually carried out). The vector $x'_{i,j}$ is the opposite of $x_{i,j}$: each component equals to ε means when the connection is ensured, and each component differs from ε if connection is not carried out. The two vectors $x_{i,j}$ and $x'_{i,j}$ are complementary ones: for all k,

$$\begin{aligned} x_{i,j} \cup x'_{i,j} &= \{x_{i,j}(1) \oplus x'_{i,j}(1), \ldots, x_{i,j}(10) \oplus x'_{i,j}(10)\} \\ &= \{v1, v2, v3, v4, v5, v6, v7, v8, v9, v10\} \end{aligned}$$

The real components of $x_{i,j}$ and $x'_{i,j}$ are the components which are different from ε. So, we obtain: for all k ($1 \leq k \leq N_{max}=10$):

$x_{i,j} = \{v1, v2, v4, v7, v8, v10\} = \{x_{i,j}(1), x_{i,j}(2), x_{i,j}(3), x_{i,j}(4), x_{i,j}(5), x_{i,j}(6)\}$ which means that $x_{i,j}$ is really fired six times.

$x'_{i,j} = \{v3, v5, v6, v9\} = \{x'_{i,j}(1), x'_{i,j}(2), x'_{i,j}(3), x'_{i,j}(4)\}$ which means that the transition $x'_{i,j}$ is really fired four times.

In the same way, we define all real firing times of various model transitions. We note that the number of firings of all transitions, during a working period of buses, is not in general the same one.

After defining all arrival times of buses at various connection stops, one can evaluate the connection times of passengers. The expressions of these connection times are given as follows.

For example, the connection time for the passengers who make the connection from line 1 to line 2 is given by:

For $k \geq 1$,
$T_{1,2}(k, j_k) = x_{2,2}(k) - x_{1,1}(j_k)$,

with $j_k = \text{Sup}\ \{l \text{ such that } x_{2,2}(k) \geq x_{1,1}(l)\}$

If $j_k - j_{k-1} = m(P_3) > 1$ (with $j_0 = 0$), then several batches of passengers make the connection. The connection time of each batch is given by: for $k \geq 1$

$T_{1,2}(k, j_{k-1} + 1) = x_{2,2}(k) - x_{1,1}(j_{k-1} + 1)$

$T_{1,2}(k, j_{k-1} + 2) = x_{2,2}(k) - x_{1,1}(j_{k-1} + 2)$

…

$$\begin{aligned} T_{1,2}(k, j_{k-1} + m(P3)) &= x_{2,2}(k) - x_{1,1}(j_{k-1} + m(P_3)) \\ &= x_{2,2}(k) - x_{1,1}(j_k) \end{aligned}$$

In the same way all connection times at each connection stop can be evaluated.

4. CONCLUSION

In this paper, we have introduced a new method to model and evaluate performances of a bus network.

Originality of this study is the possibility to use the two complementary tools Petri nets with conflicts and (max, +) algebra to describe the behavior of a bus network and then evaluate its performances. The assignment of weights which differ from 1 to certain arcs of PNCnf model allows several batches of passengers to make a connection and get on the same bus. The introduction of a routing policy enables us to solve the associated conflicts to the graphical model and to prevent its deadlock. By applying a compute algorithm, we have solved the (max, +) model describing the network. From the obtained results, a timetable evaluation problem is studied; also we have proposed an evaluation study of connection times of passengers on the network.

REFERENCES

Baccelli, F., G. Cohen, G.-J. Olsder, J.-P Quadrat, (1992). Synchronisation and linearity, Algebra for discrete Event Systems. *John wiley et sons*.

Bouillard A. (2005). Optimiation et analyse probabiliste de systèmes à événements discrets. Thèse de doctorat de l'école normale supérieure de Lyon. France.

Cohen, G., D. Dubois, J.-P. Quadrat, and M. Viot (1985), A linear-system-theoritic view of discrete-event process and its use for performance evaluation in manufacturing. *IEEE Transaction on Automatic Control*, **vol.30, n°3**, pp. 210-220.

David R. et H. Alla (1992), Du grafcet aux réseaux de Petri. *Hermes*, Paris.

Gaubert S. (1992) Théorie des systèmes linéaires dans les dioïdes. *Ph.D.* Ecole Nationale Supérieure des Mines de Paris, France.

Houssin L., S. Lahaye and J.L. Boimond (2006) Timetable synthesis using (max, +) algebra. In Proceeding of 12th IFAC Symposium Information Control Problems in Manufacturing. Pp. 375-380, May 17 – 19 2006. St-Etienne France.

Nait-Sidi-Moh A., M.-A. Manier, A. El Moudni (2006). Spectral theory in (Max, +) Algebra for performance evaluation in a transportation network. In Proceeding of 12th IFAC Symposium Information Control Problems in Manufacturing. Pp. 375-380, May 17 – 19 2006. St-Etienne France.

Nait-Sidi-Moh A. (2003). Contribution à la modélisation, à l'analyse et à la commande des systèmes de transport public par les réseaux de Petri et l'algèbre (max, plus). Thèse de doctorat de l'université de technologie de Belfort Montbéliard, France.

Nait-Sidi-Moh A., M.-A. Manier, A. El Moudni and H. Manier (2003). Performance Analysis of a Bus Network Based on Petri Nets and (max, +) Algebra. *Systems Analysis Modelling Simulation*, **Vol. 43, no. 5**, pp. 639-669, Taylor and Francis Group, ISSN 0232-9298.

Olsder G.-J., S. Subiono and M.-M. Gettrick (1998). On large scale max-plus algebra model in railway systems. *26ème école de printemps d'informatique, Algèbre (Max,+) et applications en informatique* (Ile de Noirmoutier, Vendée, France), pp. 177-192.

Proth J. M. et X. Xie, (1995). Les réseaux de Petri pour la conception et la gestion des systèmes de production. ISBN 2-225-84649-9, *Masson*, Paris.

Goverde R., P. Bovy and G. J. Olsder (1998). The max-plus algeba approchto transportation problems.

Sgroi M., L. Lavagno, Y. Watanabe and A. Sangiovanni-Vincentelli (1999). Synthesis of embedded software using free-choice Petri nets. Proceedings of automation conference, DAC'99.

De Vries R., B. De Schutter and B. De Moor (1998). On max-algebraic models for transportation Networks. *Proceedings of the International Workshop on Discrete Event Systems (WODES'98)* (Cagliari, Italy), pp. 457-462.

Yen H.-C. (2002). A Valuation-Based Analysis of Conflict-Free Petri Nets, *Systems and Control Letters*, **Vo. 45, No. 5**, pp. 387 - 395.