Discrete Event Dyn Syst (2016) 26:561–581
DOI 10.1007/s10626-015-0220-3

Analytic evaluation of the cycle time on networked conflicting timed event graphs in the (Max,+) algebra

W. M. Boussahel[1] **· S. Amari**[2] **· R. Kara**[3]

Received: 9 April 2013 / Accepted: 29 September 2015 / Published online: 16 November 2015

Abstract This work deals with performance evaluation of Conflicting Timed Event Graph (CTEG), a class of Petri net exhibiting phenomena such as synchronization, parallelism and resources sharing. It is well known that the dynamic of Timed Event Graphs (TEG) admits a linear state space representation in the (Max,+) algebra which makes the analysis and control of this class easier. There is also a possibility of associating conflicts with the TEGs by adding conflict places that are mostly considered as safe; this extended class is called CTEG. We first present an analytic evaluation of the cycle time of CTEG as a function of the cycle time of each TEG and of the timers of the conflict places. Finally, in a more general context, we look for a relaxation of the safety hypothesis on the conflict places in order to model and evaluate the cycle time on CTEGs with multiple shared resources.

Keywords Timed event graph · Conflicting timed event graphs · (Max,+) Algebra · Resources sharing · Cycle time

1 Introduction

One of the problems arising from the study of dynamic systems is their temporal behavior; therefore, two types of systems can be considered: the familiar so-called continuous dynamic systems class, described by differential equations where the state of the system changes

✉ W. M. Boussahel
wassim.boussahel@aut.unisaarland.de

S. Amari
samari@ens-cachan.fr

R. Kara
redouk@yahoo.fr

[1] Chair of Automation and Energy Systems, Saarland University, D-66123 Saarbrücken, Germany

[2] LURPA – ENS-Cachan 61, Av. du Président Wilson, 94235 Cachan, France

[3] L2CSP – Mouloud Mammeri University, Hasnaoua Street, 15000 Tizi Ouzou, Algeria

continuously and Discrete Event Systems (DES). The latter are man-made and their states change at discrete instants of time.

Petri nets are a powerful tool for modeling discrete event systems. They are often used to represent phenomena such as synchronization, parallelism and concurrency (Murata 1989). They arise in the domains of manufacturing, delivery services, vehicular traffic, and computer and communication networks. However, the more elaborated their semantics are, the more complex their analysis is. Timed Event Graphs (TEGs) are one of the most widely used classes of Petri nets because their dynamics are easily represented in the form of linear equations in the (Max,+) algebra. Such a representation is well suited to deal with control problems and performance evaluation, especially since the linearity of the models allows for a very advanced analysis similar to classical linear systems. (Baccelli et al. 1992).

Unfortunately, TEGs are Petri nets with places displaying at most one upstream transition and one downstream transition. Thereby, they are not suitable for modeling conflicts phenomena or resources sharing. Systems sharing common resources are frequently encountered; for example, the production systems that display a cyclic allocation of the shared resources (Hillion and Proth 1989). Other examples include networked control systems (Georges 2006; Addad et al. 2011). A communication network is a system with multiple shared resources (processors, routers, switches, …) and the use of TEGs is only relevant in cases where conflicts (e.g. the access to the switches) can be solved in advance and where waiting times for the access to resources can be calculated a priori, and then integrated into the TEGs. For an analytical study of more complex control systems, modeling conflicts of access to shared resources is necessary, which invalidates the modelling using only TEGs.

In the literature, different methods have been developed to address these problems of modeling and analyzing the conflicts. In Hillion and Proth (1989), the authors investigated the job-shop systems, a specific type of production system composed of a variety of jobs that must be produced using machines. The job is duplicated as many times as there are types of pieces to be produced, which allows them to develop an equivalent TEG of three types of circuits, each one containing a token. Therefore, there is no longer a possible conflict in the different machines, which allows them to get TEGs, in which only synchronization phenomena exist. In Trouillet et al. (2007), the authors are interested in flexible manufacturing systems that are characterized by shared -possibly multiple- resources, a Petri net model is developed where the lines of the operating range are duplicated as many times as there are desired products in a production cycle. Equivalent TEGs are built by duplicating common transitions.

Another form of modeling based on the use of (Max,+) automata is able to represent timed safe Petri nets (Gaubert and Mairesse 1999). These models are particularly interesting for successively evaluating a large number of schedules in safe choice free Petri nets. For example, in the case of job-shop, for each scheduling, it is sufficient to calculate the height of the 'Heaps of Pieces' without having to reconsider the model, while the classical approach consists of transforming the job-shop into TEGs as many times as there are schedules to consider. Indeed, the heaps theory allows us to get the cycle time of a certain class of Petri nets in an elegant way, but up to now, this work is not generalized to non-safe Petri nets. In the paper by Van Den Boom and De Schutter (2006), the authors propose modeling a certain class of discrete event systems with switching max-plus linear systems. In switching Max-Plus linear systems, we can switch between different modes of operation. Each mode is described by a Max-Plus linear state space model. The switching permits changing the structure of the system, breaking synchronization and changing the order of events. The model is then used to formulate a control approach strategy. More precisely, a model predictive control (MPC) design approach has been derived for switching max-plus-linear systems. The MPC optimization problem is solved using linear programming. An example of two switching modes was used for 'just-in-time control' in (Alsaba et al. 2006).

In (Boutin et al. 2009), an approach based on dioids of intervals is proposed, which represents production systems with a behavior at the limits and offers the guaranteed maximal and minimal production rates. The approach provides quantitative information about the system with a close approximation, regardless of any control action that may be applied. In (Correia et al. 2009), an approach is used in urban traffic systems. Shared resources considered in the sense that each resource can be reserved by a single entity; it means that the allocation of the resources is specific. The model can be decoupled into a TEG without shared resources, and a system of constraints in the form of inequalities, related to the availability of the resources, is obtained. In (Nait et al. 2006), the concept of virtual and real firing of a transition was introduced for transportation systems and thus represents the system in the (Max, +) algebra.

Finally, in (Addad et al. 2010, 2012), an algebraic (Max,+) approach was presented for the modeling of conflicting networked timed event graphs. Forms similar to the state space representation of TEGs are given for some conflict arbitration policies, thus allowing easier analysis. Nevertheless, the method presents three drawbacks: first, we have to reconsider all the calculation steps, if the sequence changes. Second, the approach treats only a certain type of sequences. When a TEG is disabled it is not activated another time in the sequence. Third, the approach is dedicated only to CTEG with safe conflicting places.

It is worth mentioning that the cycle time is an important performance indicator in the analysis, sizing and optimization of the cyclic DES. Its evaluation is essential in the study of such systems. This has led us to work in this direction and we can summarize our contribution on this paper as follows. This contribution is divided into two parts. In the first part, we calculate a cycle time of a CTEG as a function of the cycle time of each TEG that constitutes the network. In this part, we perform a duplication of the conflict places to calculate a modified cycle time of each TEG. This allows the integration of the conflict in each TEG. This conflict place is supposed to be safe in this part. In the second part, we relax the hypothesis on the safety of the conflict place and propose a methodology for multi-resources CTEGs. It is noteworthy that in the cited studies, the authors often considered a conflict place which contained at most one token, which is a rather strong assumption, except in (Corréïa et al. 2009) where the issue was a non-safe marking, but the authors studied the conflict locally on each user resource and the equations were written without taking the conflict into account.

The remainder of this paper is organized as follows: section 2 recalls the algebraic modeling of Timed Event Graph (TEG) in the (Max,+) algebra as well as the modeling of Conflicting Timed Event Graphs (CTEG) presented in (Addad et al. 2010, 2012). Section 3 presents the first part of the work which is a presentation of the approach used to calculate the cycle time as a function of the eigenvalues of the TEGs and the temporizations of the conflict places. Section 4 is the second part of this contribution, which is an approach for the relaxation of the hypothesis on the marking of the conflict place. Section 5 concludes the paper by presenting perspectives for future works.

2 Algebraic notions

2.1 Timed event graphs and (Max,+) algebra

In this section, we present the (Max,+) linear representation of TEGs. We recall that TEGs are Petri nets such that each place has at most one upstream and one downstream transition. In the sequel, delays are associated only with places. If the delays are not depicted, they are supposed to be null. The temporal behavior of a given TEG is obtained by associating to each transition t_i a dater function $x_i(k)$, $k \geq 0$, which represents the date of occurrence of the k^{th} event. We denote by $u_j(k)$ the dater function of the source transition t_{u_j}.

The temporal behavior of the TEG G_1 of (Fig. 1) is given in the following equations assuming that the firing is performed at maximal speed,

$$\begin{cases} x_1(k) = \max\{1 + u_1(k), 2 + x_2(k-1)\}, \\ x_2(k) \quad = \quad 3 \quad + \quad x_1(k) \quad . \end{cases}$$

The equations above can be written using the (Max,+) algebra operators, the maximum (denoted by $\oplus$) and usual addition (denoted by $\otimes$) defined on the set $\mathbb{R}\cup\{-\infty\}$ where $\mathbb{R}$ denotes the set of real numbers. The zero element (identity of the $\oplus$ law) is $\varepsilon=-\infty$ and the unit element (identity of the $\otimes$ law) is $e=0$. The structure $\mathbb{R}_{\max}=(\mathbb{R}\cup\{-\infty\}, ,)$ is called (Max, +) algebra.

The (Max, +) equations of the example given in Fig. 1 are:

$$\begin{cases} x_1(k) = 1\otimes u_1(k)\oplus 2\otimes x_2(k-1), \\ x_2(k) = 3\otimes x_1(k), \end{cases}$$

which can be written in the well-known implicit state space representation:

$$X(k) = A_0\otimes X(k)\oplus A_1\otimes X(k-1)\oplus B_0\otimes U(k), \tag{1}$$

with: $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $U=u_1$, $A_0 = \begin{pmatrix} \varepsilon & \varepsilon \\ 3 & \varepsilon \end{pmatrix}$, $A_1 = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix}$ and $B_0 = \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}$.

Theorem 1 (Baccelli et al. 1992) The minimal solution of the equation $X=A\otimes X\oplus B$, with $A\in\mathbb{R}_{max}^{n\times n}$ and $B\in\mathbb{R}_{max}^{n}$, is $X=A^*\otimes B$ with $A^*=\oplus_{i\in\mathbb{N}}A^i$, the Kleene star of A. The solution to $X=A\otimes X\oplus B$, only exists when the precedence graph of A has no circuit of positive weight.

Using the result of Theorem 1, we obtain the following explicit state space representation of the system (1),

$$X(k) = A\otimes X(k-1)\oplus B\otimes U(k), \tag{2}$$

with $A=A_0^*\otimes A_1$ and $B=A_0^*\otimes B_0$.

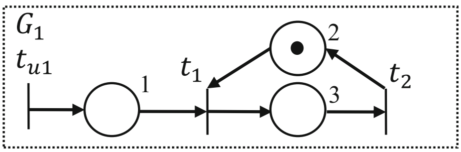


Fig. 1 A Timed Event Graph (TEG)

For the previous example: $A_0^* = \begin{pmatrix} e & \varepsilon \\ 3 & e \end{pmatrix}$, $A = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 5 \end{pmatrix}$ and $B = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$.

The recurrent Eq. (2) is similar to the state representation of classical linear systems. It is often used for the performance evaluation and control synthesis of discrete event systems (Cottenceau et al. 1999; Lahaye et al. 2004; Houssin et al. 2007; Amari et al. 2012).

The cycle time of a strongly connected timed event graph gives a quantitative index of the modelled system and represents a mean time between two consecutive firings of each transition at the steady state . In (Max,+) algebra, the cycle time gives us a matrix cyclicity which can be translated in spectral analysis (eigenvalue determination). More precisely, if a system is cyclic (with cycle c), the matrix A verifies,

$$\exists k_0, \forall k \geq k_0,\ A^{k_0 \otimes c} = \lambda^c \otimes A^{k_0},$$

with k0 being a natural number from which the system becomes periodic (the end of the transient), and λ being the spectral radius of the matrix A which is given by:

$$\lambda = \bigoplus_{j=1}^{n} \mathrm{tr}\left(A^j\right)^{\frac{1}{j}},$$

n being the order of the matrix, and tr being the trace of the matrix A which can be evaluated by:

$$\mathrm{tr}(A) = \bigoplus_{i=1}^{n} (A)_{ii}.$$

The formula providing the eigenvalue can be written in standard algebra as follows:

$$\lambda = \max_{j=1}^{n} \left(\frac{1}{j} \max_{l} \left(A^j\right)_{ll} \right).$$

From this equation, it is clear that the power in (Max, +) algebra corresponds to multiplication in standard algebra.

Note that, if the matrix A is irreducible, the eigenvalue is unique and is equal to the spectral radius. A matrix A is irreducible if its associated precedence graph is strongly connected. We can remark that this condition is sufficient but not necessary.

For example, the eigenvalue of the matrix A of the TEG of Fig. 1 is,

$$\begin{aligned} A &= \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 5 \end{pmatrix} \Rightarrow \mathrm{tr}(A) = \varepsilon \oplus 5 = 5, \\ A^2 &= \begin{pmatrix} \varepsilon & 7 \\ \varepsilon & 10 \end{pmatrix} \Rightarrow \mathrm{tr}\left(A^2\right) = \varepsilon \oplus 10 = 10, \\ \lambda &= \mathrm{tr}(A) \oplus \frac{1}{2} \mathrm{tr}\left(A^2\right) = 5 \oplus \frac{10}{2} = 5. \end{aligned}$$

In (Max,+) algebra, the value $\mathrm{tr}\left(A^k\right)^{\frac{1}{k}}$ corresponds to the maximum cycle time of all the circuits of length k of a given TEG. The eigenvalue of A gives the cycle time of the corresponding timed event graph.

Remark 1 Within the context of cyclic production systems, the cycle time corresponds to the production period or to the inverse of the throughput.

2.2 Conflicting timed event graphs in the (Max,+) algebra

We recall in this part, the modeling of conflicting TEGs used in (Addad et al. 2012). A conflicting timed event graph CTEG is a particular class of timed Petri nets defined by a set $G=\{G_1, G_2,\ldots,G_N\}$ of TEGs connected to each other by a set $\tilde{R}=\{\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m\}$ of shared places. For a concise definition, we recall some hypotheses and notations.

The transitions of each TEG G_i, $i=1,\ldots,N$, are denoted t_j^i, $j=1,\ldots,q_i$, where q_i represents the number of transitions in the TEG G_i. The conflicting place $\tilde{p}_l$ is connected to a given TEG G_i by a unique downstream transition and by a unique upstream transition. The conflicting places are safe. Recall that a place is called safe if, for any evolution of a given Petri net, its marking does not exceed one token. A shared resource is used by, at most, one user at a time; in other terms, when the place $\tilde{p}_l$ is not marked, only one elementary circuit $t_j^i\ldots t_f^i\tilde{p}_l\, t_j^i$, among all others that involve place $\tilde{p}_l$, contains it at a time. The temporization associated with a conflict place $\tilde{p}_l$ is denoted $\tilde{\tau}_l$. A periodic allocation of the tokens of the conflicting places results in a periodic functioning of the considered CTEG. An example of a CTEG is given by the Fig. 2.

According to these hypotheses, the conflict places cannot contain more than one token. This token is allocated to the TEGs according to a specific sequence σ.

In (Addad et al. 2012) the authors associated a dater with the token of the conflict place, which will be the availability date of this token for the k^{th} time. If the k^{th} incoming token of the conflict place is consumed by the TEG *G1* and if this latter has already consumed ($k1$-1) tokens from the conflict place, the dynamics of the TEG *G1* are given by:

$$\begin{cases} x_1^1\,(k_1) \;=\; 1\otimes u^1\,(k_1)\oplus 1\otimes x_2^1\,(k_1-1)\oplus\tilde{\psi}(k)\,, \\ x_2^1(k_1)=4\otimes x_1^1(k_1)=5\otimes u^1(k_1)\oplus 5\otimes x_2^1(k_1-1)\oplus 4\otimes\tilde{\psi}(k). \end{cases} \tag{3}$$

Such that $\tilde{\psi}(k)$ represents the availability of the k^{th} token of the conflict place. The transition t_1^1 is concerned with the conflict, then $\tilde{\psi}(k)$ appears in the system of Eq. (3). The system (3) in the matrix form is:

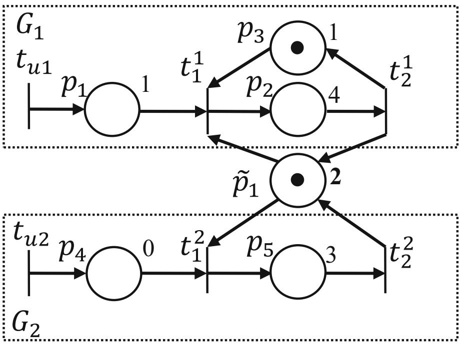


Fig. 2 Two conflicting TEGs (CTEG)

$$X_1(k_1) = \begin{pmatrix} \varepsilon & 1 \\ \varepsilon & 5 \end{pmatrix} \otimes X_1(k_1 - 1) \oplus \begin{pmatrix} 1 \\ 5 \end{pmatrix} \otimes U_1(k_1) \oplus \begin{pmatrix} e \\ 4 \end{pmatrix} \otimes \tilde{\psi}(k).$$

Furthermore, if the k^{th} token of the conflict place is consumed by the TEG G_1, the $(k+1)^{\text{th}}$ token is generated with the firing, for the $k_1{}^{th}$ time, of the upstream transition of the conflict place. Thus, we can write:

$$\tilde{\psi}(k+1) = 2 \otimes x_2^1(k_1). \tag{4}$$

To summarize the approach, if the k^{th} token of the conflict place is consumed by the TEG G_i, the following (Max,+) recurrent equations are verified for k defined as:

$$k = \sum_{i=1}^{N} k_i - 1, \quad 1 \le i \le N,$$

where N is the total number of TEGs.

$$\begin{cases} X_i(k_i) = A_i \otimes X_i(k_i - 1) \oplus B_i \otimes U_i(k_i) \oplus F_i \otimes \psi(k), \\ \Psi(k+1) = G_i \otimes X_i(k_i). \end{cases} \tag{5}$$

These recurrent (Max,+) equations describe the system regardless of the arbitration policy that is used.

By following this procedure for the TEG G_2, we find the characteristic matrices of the CTEGs for this system:

$$A_1 = \begin{pmatrix} \varepsilon & 1 \\ \varepsilon & 5 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix}, \quad B_2 = \begin{pmatrix} e \\ 3 \end{pmatrix},$$
$$F_1 = (e \quad 4)^t, \quad G_1 = (\varepsilon \quad 2), \quad F_2 = (e \quad 3)^t, \quad G_2 = (\varepsilon \quad 2).$$

3 Evaluation of the cycle time on Conflicting Timed Event Graphs

3.1 Evaluation of an upper bound of the cycle time: duplication of the conflict places

Let us consider the CTEG in Fig. 2. We explain our approach which consists of the evaluation of an upper bound of the cycle time under the hypothesis that all conflicting places are safe. The token of the conflict place is allocated according to a cyclic sequence of TEGs.

Let us denote this sequence by σ. It consists of the order of the allocation.

$$\sigma = G_{i_{(0)}} G_{i_{(1)}} \ldots G_{i_{(K-1)}},$$

where $i_{(0)} \ldots i_{(K-1)}$ are the indices of the TEG G_i ($1 \le i \le N$), that are part of the CTEG. The token of the conflict place is allocated K times.

We duplicate the conflict place as shown in Fig. 3:

This duplication is always possible since the conflict place belongs to a conservative component of the considered net. The temporal parameter α_i is introduced to show an additive delay that may be taken by the token. This parameter is defined as:

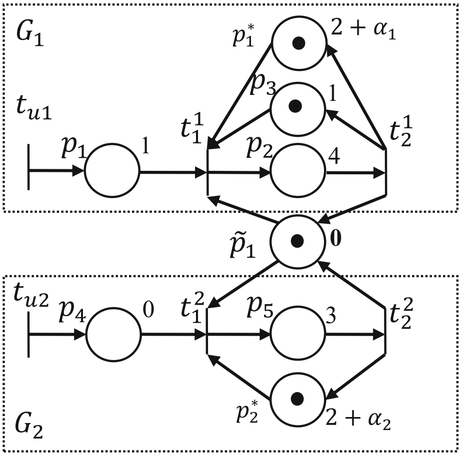


Fig. 3 Duplication of the conflict place

$$\begin{cases} a_i = 0 & \text{if the token is available for the TEG } G_i, \\ a_i > 0 & \text{otherwise} \end{cases}$$

After duplicating the conflict place, one property of the obtained CTEGs is the null delay of the conflict place token. This allows the different TEGs to communicate, that is to say, if the token is consumed by a TEG, it is taken back directly by the next TEG without a waiting time. The obtained TEGs (which are the initial TEGs with the duplicated conflict places) will have different state space matrices which are noted in the sequel A_{mi} (for a TEG G_i) and then different eigenvalues. The obtained eigenvalue of a TEG Gi is denoted by λ_i^* and baptized: modified eigenvalue.

Remark 2 The original CTEG may have some TEGs that are not strongly connected (no uniqueness of the eigenvalue). After duplication, these TEGs become strongly connected and then have a unique eigenvalue. For example, let us consider the TEG G_2 in Fig. 2. The eigenvalue of this TEG using the matrix A_2 has a value of zero. That is to say that without conflict, the TEG has no cycle. However, the matrix of the new TEG G_2 is:

$$A_{m2} = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 5 \end{pmatrix}.$$

The modified eigenvalue is equal to 5 time units. This means that duplication highlights the real cycle that involves the integration of conflict in the TEG.

Property 1 The upper bound λ_{ub} of a given CTEG with a repetitive sequence σ is given by the following formula,

$$\lambda_{ub} = \sum\nolimits_i n_i \cdot \lambda_i^* + \sum\nolimits_i \Delta t_i. \tag{6}$$

The indices i take their value from 1 to N, according to the indices of the TEGs that appear in the sequence and where n_i is the number of times that the TEG G_i appears in the sequence σ. Δt_i is an additional time due to the duplicated timed places. In other terms, the temporizations of the conflict places can delay the execution of a given TEG.

Proof To prove this, we break down a sequence σ into elementary sequences that contains only one TEG G_i n_i times, for instance, $\sigma = G_1^{n_1} \ldots G_l^{n_l} \ldots G_k^{n_k} \ldots$ We are interested in the determination of the time execution of the sequence σ (i.e. the period T). To do this, we compute the duration of each token allocation. We can easily observe that the tokens of the TEG G_i were waiting for the token allocation, so the tokens have remained temporarily in their places and can be available to fire transitions. Furthermore, the switching between TEGs may generate a time delay due to the duplicated places. More precisely, in the case where a token is allocated to a TEG G_i knowing that G_i is preceded by itself in the sequence, there is no additional delay between the last fired transition and the first transition of the considered TEG (i.e. Δt_i=0). In the case where a token is allocated to a TEG G_i knowing that G_i is preceded by another TEG G_k the first transition of G_i will fire after $\Delta t_i = \max \{\tilde{\tau}_l\}$, with $\tilde{\tau}_l$ representing the temporization of the input conflicted places of the first transition of the TEG G_i.

It follows directly that $T_i \leq n_i.\lambda_i^* + \Delta t_i$. Summing up the obtained inequalities, we obtain:

$$T = \sum\nolimits_i T_i \leq \sum\nolimits_i \left(n_i \cdot \lambda_i^* + \Delta t_i\right) = \lambda_{ub.}$$

3.2 Calculation of the exact cycle time of CTEGs with one safe conflict place

We use the duplication method to calculate an exact cycle time for CTEGs with one conflict place. The main idea of this method is to calculate the delay of the available tokens of the TEGs that are waiting for the allocation of the conflict place token. This is done by going through the sequence, step by step. A step begins when the token is allocated to a TEG and ends when it is released by the same TEG. The TEG that takes the token is characterized by its modified eigenvalue at the step k.

Let us consider a sequence of K steps:

$$\sigma = G_{i_{(0)}} G_{i_{(1)}} \ldots G_{i_{(K-1)}},$$

where $i_{(k)}$ represents the indices of the TEGs at step k.

At step k, the TEG G_i can be found in three locations in the sequence:

– At the beginning of the sequence:

$$\sigma = G_{i_{(0)}} \ldots,$$

– Preceded by itself:

$$\sigma = \ldots G_{i_{(k-1)}} G_{i_{(k)}} \ldots,$$

– Change in the sequence:

$$\sigma = \ldots G_{j_{(k-1)}} G_{i_{(k)}} \ldots (i \neq j).$$

At step k, these three locations can be grouped into two cases:

1st case G_i at the beginning of the sequence or preceded by itself. We associate with each upstream place of the transition related to the conflict a delay noted $t_{j(k)}^{pi}$. It corresponds to the time spent by a token in the place p_i belonging to the TEG G_j at step k in order to enable the downstream transition. The considered parameters are shown in Figs. 4 and 5. We calculate the modified eigenvalue at step k of the TEG that was allocated the conflict token. Also, at the end of this step we compute the tokens delays of the TEGs that are waiting for the allocation.

Lemma 1 The delay of tokens of the upstream place p_i of the transitions related to the conflict of the TEG G_j waiting an allocation knowing that G_j is preceded by itself in the sequence is given by:

$$t^{pi}_{j(k+1)} = \begin{cases} t^{pi}_{j(0)}, & \text{if the place is safe,} \\ \left(t^{pi}_{j(k)} - \lambda^*_{j(k)}\right) \oplus e & \text{otherwise,} \end{cases} \tag{7}$$

where: $t^{pi}_{j(k)}$ is the delay of the place p_i of the TEG G_j at step k and $t^{pi}_{j(0)}=d_i$, the temporization of the place p_i.

Proof to prove this lemma we have to consider two cases.

1st case: a safe place. If the place is safe, all transitions fired the previous cycle and the tokens have only been in the places they are currently in since their upstream transition fired, and should stay there at least for the default holding time $t^{pi}_{j(0)}$.
2nd case: a non-safe place. Actually, the upstream places of the transition related to the conflict may contain more than one token. It follows directly that after a cycle the tokens of this place that have not participated in the previous cycle will wait $(t^{pi}_{j(k)} - \lambda^*_{j(k)})$. When this delay is negative, the formula follows immediately.

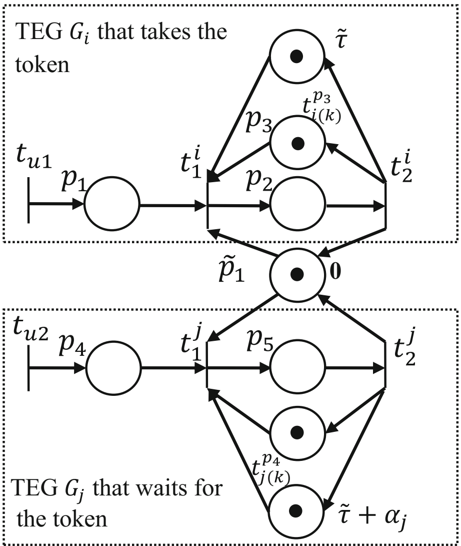


Fig. 4 Calculation of the exact cycle time (1st case)

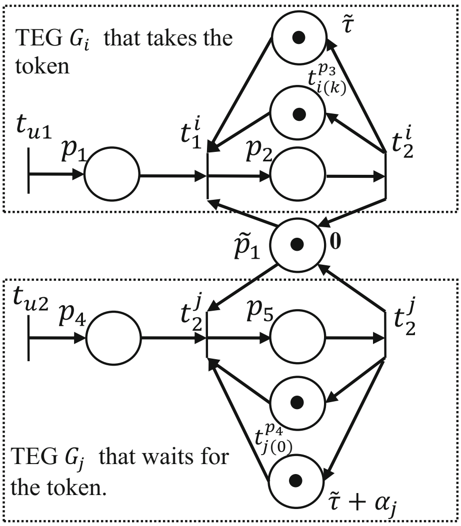


Fig. 5 Calculation of the exact cycle time (2nd case)

2nd case change in the sequence:

Let G_j and $G_{(j-1)}$ be these two TEGs such that G_j is preceded by $G_{(j-1)}$ in the sequence σ. the following Lemma characterizes the waiting time of a token in a place p_i concerned with the conflict.

Lemma 2 The token delay of a place p_i belonging to a TEG G_j waiting for an allocation, knowing that the TEG G_j is preceded by the TEG $G_{(j-1)}$ in the sequence, is given by:

$$t_{j(k+1)}^{pi} = \left(t_{j(k')}^{p_i} - \sum_{i=k'}^{k-1} \lambda_{i(i+1)}^{*} \right) \oplus e, \tag{8}$$

with k' being the last step where the TEG G_j appears in the sequence and $t_j\ (0)^{p_i} = d_i$, the temporization of the place p_i.

Proof The proof is obvious since the TEGs with duplicated places are periodic.

At the end of the step, we sum up the modified eigenvalues that have been calculated at each step.

$$\lambda_{exact} = \sum_i \lambda_{i(k)}^{*}. \tag{9}$$

We finally get the exact cycle time.

Remark 3 In (Addad et al. 2012) the authors are interested in particular periodic sequences. More precisely, if a TEG G_i is disabled in the sequence it is not activated another time in the sequence. For example, the sequence $\sigma = G_1 G_1 G_2 G_1 G_2$ cannot be treated with this method. In our approach, it is not the case; we can activate a TEG G_i in sequence as many times as we want.

3.3 Examples of application

3.3.1 Example 1 (One conflict place)

Let us consider the CTEGs of Fig. 6a, with one shared resource. The CTEGs with the duplicated conflict place is shown in Fig. 6b. We first evaluate an upper bound of the cycle time by using the relation (6). Then we calculate the exact cycle time by using the iterative procedure of section 3.2

We consider the following sequence:

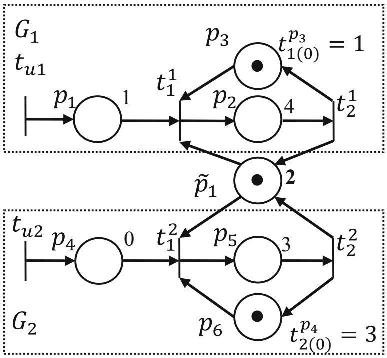
$$\sigma = G_1 G_2 G_2.$$

The modified eigenvalues are:

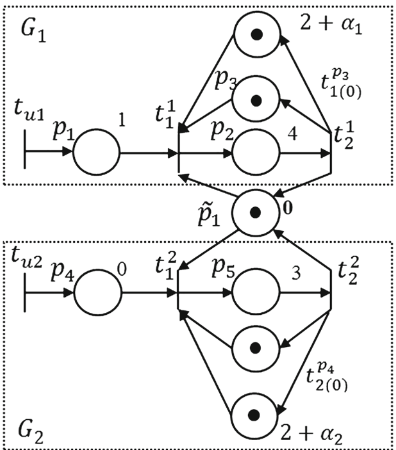
$$A_{m1} = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 6 \end{pmatrix} \Rightarrow \lambda_1^* = 6,$$

$$A_{m2} = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 6 \end{pmatrix} \Rightarrow \lambda_2^* = 6.$$

Fig. 6 **a**. CTEG (one conflict place). **b**. CTEG with the duplicated conflict place



a. CTEG (one conflict place)



b. CTEG with the duplicated conflict place

Let us calculate, using the relation (6), an upper bound for the cycle time:

$$\lambda_{ub} = \lambda_1^* + 2\lambda_2^* + \Delta t = 6 + 12 + 2 = 20.$$

Let us calculate the exact cycle time by using the procedure of section 3.2. The matrices of the TEGs at the step k are:

$$\begin{cases} A_{m1(k)} = \begin{pmatrix} \varepsilon & t_{1(k)}^{p_3} \oplus 2 \\ \varepsilon & 4\left(t_{1(k)}^{p_3} \oplus 2\right) \end{pmatrix}, \\ A_{m2(k)} = \begin{pmatrix} \varepsilon & t_{2(k)}^{p_4} \oplus 2 \\ \varepsilon & 3\left(t_{2(k)}^{p_4} \oplus 2\right) \end{pmatrix}. \end{cases}$$

– *Step 0*: G_1 at the beginning of the sequence (1st case)

$$A_{m1(0)} = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 6 \end{pmatrix} \Rightarrow \lambda_{1(0)}^* = 6,$$
$$t_{2(1)}^{p_4} = \left(t_{2(0)}^{p_4} - \lambda_{1(0)}^*\right) \oplus e = (3-6) \oplus e = e.$$

– *Step 1*: change in the sequence (2nd case)

$$A_{m2(1)} = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 5 \end{pmatrix} \Rightarrow \lambda_{2(1)}^* = 5,$$
$$t_{2(2)}^{p_4} = \left(t_{2(1)}^{p_4} - \lambda_{2(1)}^*\right) \oplus e = (0-5) \oplus e = e.$$

– *Step 2*: G_2 preceded by itself (1st case)

$$A_{2(2)} = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & 6 \end{pmatrix} \Rightarrow \lambda_{2(2)}^* = 6,$$
$$t_{2(3)}^{p_3} = \left(t_{2(1)}^{p_3} - \left(\lambda_{2(1)}^* + \lambda_{2(2)}^*\right)\right) \oplus e = (0-10) \oplus e = e.$$

– End of iterations:

$$\lambda_{exact} = \lambda_{1(0)}^* + \lambda_{2(1)}^* + \lambda_{2(2)}^* = 17.$$

This result is the exact cycle time of the CTEG.

3.3.2 Example 2 (Two conflict places)

In this example, we evaluate an upper bound for the cycle time of CTEG. Let us consider the system presented in (Gaubert and Mairesse 1999) and studied again by (Addad et al. 2010). After duplicating the conflict places, the CTEGs in Fig. 7 are obtained. The tokens are allocated according to the following sequence m times for G_1 and m times for G_2:

$$\sigma = (G_1)^m (G_2)^m.$$

Let us calculate an upper bound using the matrix of the system with the duplicated places, in order to compare it to the exact value. We use the characteristic matrices of the CTEGs of Fig. 7.

The parameters of the TEG G_1 are:

$$A_{m1} = \begin{pmatrix} \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & 3 \end{pmatrix} \Rightarrow \lambda_1^* = 3\ ,$$

$$\begin{cases} X_1(1) = (e \quad 2 \quad 3)^t \\ X_1(2) = (3 \quad 5 \quad 6)^t \end{cases} \Rightarrow (\Delta t)_1 = e,$$

the parameters of the TEG G_2 are:

$$A_{m2} = \begin{pmatrix} \varepsilon & 2 & 1 \\ \varepsilon & 3 & 2 \\ \varepsilon & 5 & 4 \end{pmatrix} \Rightarrow \lambda_2^* = 4\ ,$$

$$\begin{cases} X_2(1) = (2 \quad 3 \quad 5)^t \\ X_2(2) = (6 \quad 7 \quad 9)^t \end{cases} \Rightarrow (\Delta t)_2 = 2.$$

Using the relation (6):

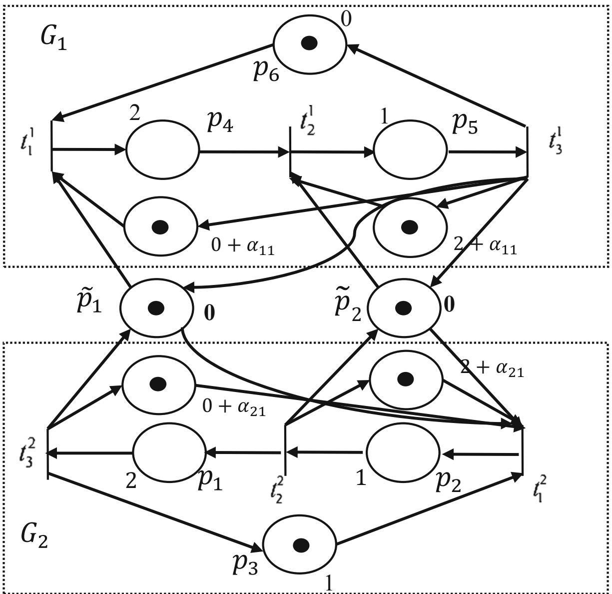
$$\lambda_{ub} = m\lambda_1^* + m\lambda_2^* + (\Delta t)_1 + (\Delta t)_2 = 7m + 2.$$

Remark 5 Addad et al. (2010) calculated the exact eigenvalue by using the global matrix given throughout the considered sequence. The calculation of the eigenvalue gives:

$$\lambda_{exact} = 7m + 1.$$

The drawback of the approach presented in (Addad et al. 2010) is the complexity of the calculations of the global matrix as well as the specificity of the sequence. If the sequence is changed, we have to reconsider the calculations to get the new global matrix.

Fig. 7 CTEGs (two conflict places)



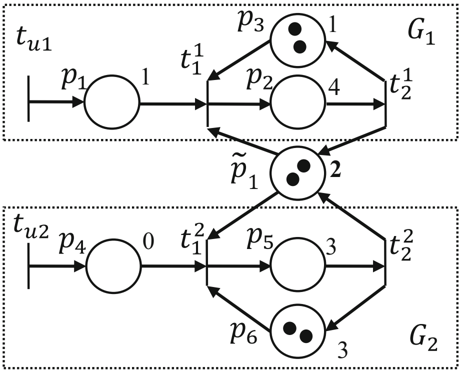


Fig. 8 Two conflicting TEGs with one non-safe conflict place

We compare the two approaches and we calculate the error on the upper bound compared to the exact value:

$$\Delta\lambda = \lambda_{ub} - {}^{ub}\lambda_{exact} = 1.$$

The calculation of the upper bound overcomes these calculations and gives a satisfactory result.

4 Modeling and evaluation of the cycle time of multi-resources Conflicting Timed Event Graphs

In this section, we relax the hypothesis concerning the token of the conflict place. Indeed, most of the approaches concerning the modelling of the conflicts consider the case of mono-resources (or safe conflict place). More particularly, this assumption is often taken into account in the study of the CTEGs.

Hypothesis H.1: The conflicting timed event graph contains at least one non-safe conflict place.

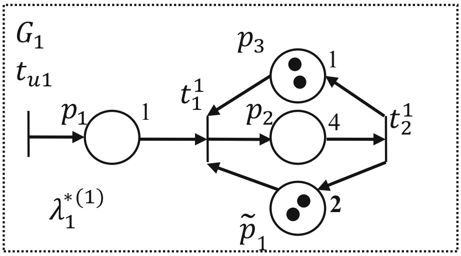
To illustrate our approach, we consider the CTEG of Fig. 8, with two tokens in the conflict place. If both tokens are allocated simultaneously, the system can be divided into three operation modes:

- Mode 1: the two tokens are allocated to TEG G_1
- Mode 2: the two tokens are allocated to TEG G_2
- Mode 3: one token is allocated to each TEG

Each mode will represent a timed event graph without conflict. We denote by $\lambda_i^{*(l)}$ the eigenvalue of the TEG G_i corresponding to the mode l. These three modes are illustrated as follows:

Mode 1: Fig. 9
Mode 2: Fig. 10
Mode 3: Fig. 11

Fig. 9 TEG representing mode 1



We consider that the initial TEG can commute between several functioning modes. We assume the case of q modes. In each mode $l=1,\ldots,q$ the system is described by a linear (Max,+) system of the form studied by (Alsaba et al. 2006; Van den Boom and De Schutter 2006):

$$\begin{cases} x(k) = A^{(l)} \otimes x(k-1) \oplus B^{(l)} \otimes u(k), \\ y(k) = C^{(l)} \otimes x(k), \end{cases} \tag{10}$$

where $A^{(l)}$, $B^{(l)}$, $C^{(l)}$ are the characteristic matrices corresponding to the l^{th} operating mode.

The switching instants of time are determined by a mechanism known as commutation or switching. A variable $S(k)$ is defined and could depend on a certain application, previous states, previous modes, an input variable or on an output variable of decision.

$S(k) \in \mathbb{R}^{n_S}$ where n_S is the dimension vector of the commutation S. In our case, for example, the decision will be external, depending on the imposed sequence for the tokens.

$\mathbb{R}^{n_S}$ is partitioned into q subsets $Z^{(i)}$, $i=1,\ldots,q$. The mode $l(k)$ is obtained by determining which subset the switching variable $S(k)$ belongs to: if $S(k) \in Z^{(i)}$, then $l(k)=i$. An example of a system with three switching linear (Max,+) modes is represented in Fig. 12.

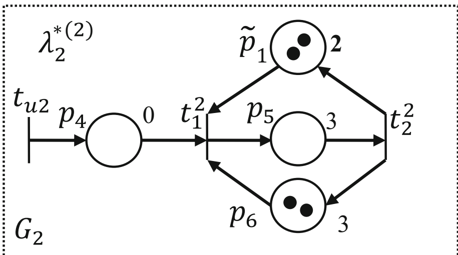
In addition, if we introduce the hypothesis that we need to wait for the arrival of the two conflict place tokens, we can finally formalize the system according to these switching modes.

4.1 Summary of evaluation of upper bound method

For CTEGs with a non-safe conflict place, we have to consider their various operating modes. Thus, we consider that the system can switch between its various modes as we await the arrival of the tokens in the conflict place to change from one mode to another. Each mode is represented by a linear (Max,+) model without conflict. One specific mode is characterized by the eigenvalue of the TEG that represents the mode.

Regarding the sequence, we assume that the sum of capacities of all TEGs is less or equal to the number of tokens in the conflict place. Then, we divide the sequence into equal parts where

Fig. 10 TEG representing mode 2



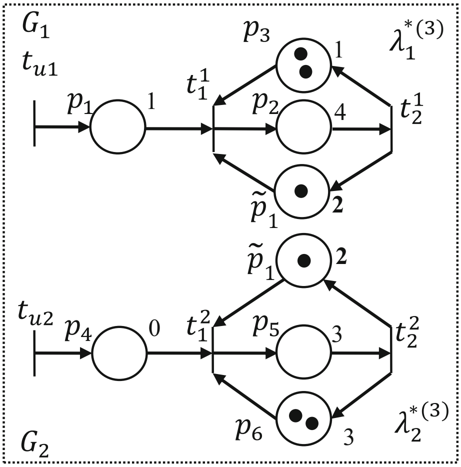


Fig. 11 TEGs representing mode 3

each part will contain a number of TEGs equal to the number of the conflict place tokens. Therefore, each part represents a specific operation mode. Thus, the division of the sequence depends on the initial marking of the conflict place. This division allows us to evaluate the upper bound of the cycle time, which is being done by summing up the eigenvalues of each part.

These steps summarize our approach and allow us to evaluate an upper bound of the cycle time. The steps are given in the following algorithm:

- *Step 0:* CTEG with a non-safe conflict place + sequence σ.
- *Step 1:* Break down the system into subsystems without conflicts (operating modes).
- *Step 2:* Divide the sequence into equal parts (the length of a part is equal to the number of tokens in the conflicting place).
- *Step 2.1:* Find the operating mode of each part.
- *Step 2.2:* Determine the eigenvalue of each part.
- *Step 3:* The upper bound is evaluated by summing the eigenvalues (*Step 2.2*) of each part.

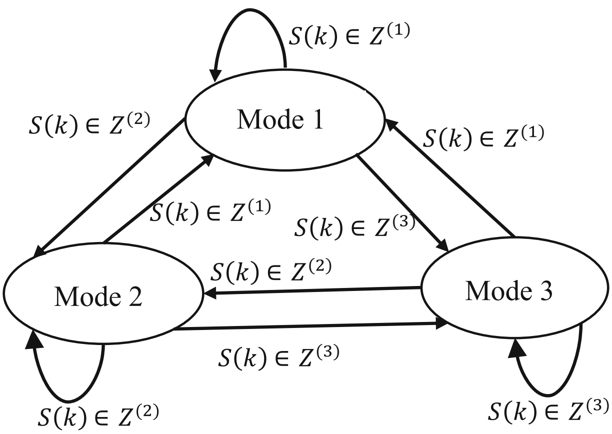


Fig. 12 System with three switching linear (Max,+) modes

4.2 Example (cycle time of CTEG, Fig. 8)

To illustrate this methodology, let us consider for example, that the two tokens are allocated according to the following sequence:

$$\sigma = G_1 G_2 G_2 G_1 G_1 G_1 G_2 G_2.$$

We divide the sequence by pairs of TEGs (2 tokens in the conflicting place); each one of them could be identified with one of the modes described before.

Pair	Mode
G_1G_2	Mode 3
G_2G_1	Mode 3
G_1G_1	Mode 1
G_2G_2	Mode 2

Thus, to switch from the first to the second pair (or from the second to the third pair), there is a waiting time that corresponds to the generation of both tokens in the conflict place. This waiting time will be evaluated by the difference between the modified eigenvalues of the two TEGs with one token in the conflict place:

$$\left| \lambda_1^{*(3)} - \lambda_2^{*(3)} \right|, \tag{11}$$

where $\lambda_i^{*(l)}$ is the modified eigenvalue of the TEG G_i specific to the mode l.

However, in a mode where both tokens are consumed simultaneously by the same TEG (mode 1 and mode 2), the waiting time is null because the tokens are delivered at the same time. Regarding the upper bound of the cycle time, it can be evaluated by summing up the partial eigenvalues, corresponding to each pair of TEGs (which is each mode).

In our case, we sum up the eigenvalues corresponding to the 4 pairs of the sequence:

$$\begin{aligned} \lambda_{ub} &= \left(\lambda_1^{*(3)} \oplus \lambda_2^{*(3)}\right) \otimes \left(\lambda_2^{*(3)} \oplus \lambda_1^{*(3)}\right) \otimes \lambda_1^{*(1)} \otimes \lambda_2^{*(2)}, \\ \lambda_{ub} &= (6 \oplus 5) \otimes (5 \oplus 6) \otimes 3 \otimes 3 = 18. \end{aligned}$$

So, if the difference between the modified eigenvalues of both TEGs has a small value (corresponding to Mode 3 in our example), this approach can satisfactorily meet the needs of the sequence.

5 Conclusion

In this paper, we investigated the problem of the evaluation of the cycle time of conflicting timed event graphs (CTEG). This evaluation is a function of the eigenvalues of the TEGs and of the timers of the safe conflict places. We duplicated the safe conflict places in order to correct the eigenvalues; it leads to the evaluation of an upper bound of the cycle time in the case of one or more shared resources. We also calculated an exact cycle time in the case of one shared resource by using an iterative process. This is being done by calculating during each step the modified eigenvalue of the TEG to which the token was allocated and, meanwhile, by calculating the temporizations of the available tokens of the TEGs that are awaiting the allocation. Two application examples have been presented to illustrate the approaches that we presented. In a more general context, we looked for the hypothesis relaxation on the conflict places. We hence considered non-safe conflict places. We proposed a methodology by introducing a waiting hypothesis on the tokens of the conflict place. We considered that the system can switch between different operation modes. One application example of a CTEG with a two tokens conflict place has been presented. In future research, it would be interesting to deepen this latter modeling problem by proposing an algebraic model, which describes the system in a more optimal way, and by calculating an exact cycle time. It could be also interesting to study CTEGs with more than one non-safe conflict place.

References

Addad B, Amari S, Lesage J-J (2010) Modélisation de réseaux de graphes d'évènement temporisés avec conflits dans l'algèbre (Max,+). IEEE Conférence Internationale Francophone d'Automatique, Paper n°22 Nancy, France

Addad B, Amari S, Lesage J-J (2011) Genetic algorithms for delays evaluation in networked automation systems. Eng Appl Artif Intell 24(3):485–490

Addad B, Amari S, Lesage J-J (2012) Networked conflicting timed event graphs representation in (Max,+) algebra. Discrete Event Dyn Syst 22(4):429–449

Alsaba M, Lahaye S, Boimond J-L (2006) On just in time control of switching max-plus linear systems. In Proceedings of ICINCO'06, Setubal, Portugal, pp. 79–84

Amari S, Demongodin I, Loiseau J-J, Martinez C (2012) Max-plus control design for temporal constraints meeting in timed event graphs. IEEE Trans Autom Control 57(2):462–467

Baccelli F, Cohen G, Olsder G J, Quadrat J-P (1992) Synchronization and linearity: an algebra for discrete event systems. Wiley

Boutin O, Cottenceau B, L'Anton A, Loiseau J-J (2009) Modeling systems with periodic routing functions in dioid (Min,+), 13th IFAC Symposium on Information Control Problems in Manufacturing, Moscow, Russia, pp. 1377–1382

Corréïa A, Abbas TA, Bouyekhf R, El Moudni A (2009) A dioid model for invariant resource sharing problems. IEEE Trans Syst Man Cybern Syst Hum 39(4):770–781

Cottenceau B, Hardouin L, Boimond J-L, Ferrier J-L (1999) Synthesis of greatest linear feedback for timed event graph in dioid. IEEE Trans Autom Control 44(6):1258–1262

Gaubert S, Mairesse J (1999) Modeling and analysis of timed Petri nets using heaps of pieces. IEEE Trans Autom Control 44(4):683–697

Georges J-P (2006) A design process of switched Ethernet architectures according to real-time application constraints. Eng Appl Artif Intell 19(3):335–344

Hillion H, Proth J-M (1989) Performance evaluation of jobshop systems using timed event graphs. IEEE Trans Autom Control 34(1):3–9

Houssin L, Lahaye S, Boimond J-L (2007) Just in time control of constrained (Max,+)-linear systems. Discrete Event Dyn Syst 17(2):159–178

Lahaye S, Boimond J-L, Hardouin L (2004) Linear periodic systems over dioids. Discrete Event Dyn Syst 14(2): 133–152

Murata T (1989) Petri nets: properties analysis and applications. Proc IEEE 77(4):541–580

Nait SM, Manier MA, El Moudni A, Wack M (2006) Petri net with conflicts and (Max,plus) algebra for transportation systems, 11th IFAC Symposium on Transportation Systems, The Netherlands, pp. 548–553

Trouillet B, Korbaa O, Gentina JC (2007) Formal approach for fms cyclic scheduling. IEEE Trans Syst Man Cybern Part C Appl Rev 37(1):126–137

Van Den Boom T-J-J, De Schutter B (2006) Modelling and control of discrete event systems using switching Max-Plus linear systems. Control Eng Pract 14(10):1199–1211

Boussahel Wassim Mohamed received the Engineer degree in Automation from the Ecole Nationale Polytechnique d'Alger, Algiers, Algeria, in 2011, and the Master's degree in Complex Systems Engineering from Ecole Normale Supérieure de Cachan, Cachan, France, in 2012. He is currently pursuing his research at Saarland University in Germany. His research interests include modeling and analysis of discrete event systems in the context of energy efficiency.

Saïd Amari received the Ph.D. degree from the Institute of Research in Communications and Cybernetic of Nantes, Nantes, France, in 2005. He is currently an Associate Professor at the University of Paris XIII. He carries out research at the Automated Production Research Laboratory from the École Normale Supérieure de Cachan. His main research interests are performance evaluation and control of discrete-event systems with Petri nets and Dioid algebra.

Redouane KARA received the Ph.D. degree from the Mouloud Mammeri University, Tizi-Ouzou, Algeria, in 2009. He is currently an Associate Professor at the University of Mouloud Mammeri. He carries out research at the Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), Mouloud Mammeri University. His main research interests are analysis and control of discrete event systems.