

*Section III*

**Multiple-Machine Scheduling**

## **DYNAMIC ANALYSIS OF REPETITIVE DECISION-FREE DISCRETE-EVENT PROCESSES: THE ALGEBRA OF TIMED MARKED GRAPHS AND ALGORITHMIC ISSUES**

Didier DUBOIS

*Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier,  
118 Route de Narbonne, F-31062 Toulouse Cedex, France*

and

Kathryn E. STECKE

*Graduate School of Business Administration, The University of Michigan, Ann Arbor,  
Michigan 48109, USA*

### **Abstract**

A model to analyze certain classes of discrete event dynamic systems is presented. Previous research on timed marked graphs is reviewed and extended. This model is useful to analyze asynchronous and repetitive production processes. In particular, applications to certain classes of flexible manufacturing systems are provided in a companion paper. Here, an algebraic representation of timed marked graphs in terms of recurrence equations is provided. These equations are linear in a nonconventional algebra, that is described. Also, an algorithm to properly characterize the periodic behavior of repetitive production processes is described. This model extends the concepts from PERT/CPM analysis to repetitive production processes.

### **1. Introduction**

Manufacturing automation applications are increasing in industry because of the new abilities to manufacture in low and medium volumes efficiently. The increase in the number of operating options with automation increases the complexity of system planning. These facts have motivated recent research dealing with the modeling, design, performance evaluation, simulation, planning, and control of flexible manufacturing systems. Many models have been suggested to help to describe the behavior of these systems under various points of view and levels of detail and different sets of assumptions. Many of these models are reviewed in Bel and Dubois [3] and Buzacott and Yao [5].

This work concerns one of these models, one that views a manufacturing system as a deterministic, discrete-event, dynamic system (DEDS) that repetitively performs a set of tasks (or activities). This type of model has been introduced by Cuninghame-Green [14,16] and also studied by Cohen et al. [8,10,11]. An algebraic framework for this model exists in the theory of dioids (Gondran [22] and Gondran

and Minoux [24]). These provide a linear formulation, similar to that of linear systems in control theory, under a state space approach. As a consequence, the behavior analysis of deterministic DEDS can be achieved by solving an eigenvalue problem. This class of systems turns out to be identical to a special class of timed Petri nets (Peterson [31], Chretienne [7], and Ajmone Marsan [2]), called *event graphs*. Applications to automated manufacturing have been described by Dubois and Stecke [19,20], and Cohen et al. [8].

This paper is a survey of the modeling, algorithmic, and theoretical issues associated with timed marked graphs. A companion paper (Dubois and Stecke [20]) addresses some solution issues and provides applications of this model to some problems of certain classes of flexible manufacturing systems that exhibit periodic behavior.

The plan of the paper is as follows. Section 2 begins by reviewing definitions and concepts associated with timed marked graphs and timed Petri nets. We specify convenient modeling conventions for timed marked graphs in section 2.2. Section 2.3 reviews the concept of reachability and illustrates some issues with an example. The algebraic representation of timed marked graphs in terms of recurrence equations is provided in section 3.1. These equations are linear in a particular algebra, where "maximum" represents "addition" and "addition" represents "product". This  $\{\max, +\}$  algebra has been termed a path algebra (Carré [6]) in the graph-theoretic literature. Various representation issues are also addressed in section 3, such as outside influences on the performance of repetitive systems represented by timed marked graphs. A convenient choice of state variables provides an efficient representation of the recurrence equations. There are two modes of system evolution. The earliest starting times of activities based on an initial system state can be determined (primal mode). Alternatively, the latest starting times of activities based on system final states, such as due dates, can be determined (dual mode). These are extensions of PERT/CPM ideas to repetitive production environments. These repetitive systems eventually reach steady state with periodic behavior in either mode (Cohen et al. [8]).

The steady state is characterized by solving the recurrence equations in section 4. Some material concerning the periodicity of the system from Cohen et al. [8] is reviewed. This allows the primal mode to be characterized. Other previous results (Cuninghame-Green [15]) are used to characterize the dual mode. With these results, useful parameters such as dynamic analogues to slack times and critical paths of PERT/CPM are calculated. These marked graphs also model the evolution of limited nonconsumable resource usage. This algebraic analysis allows computation of critical resources, whose shortage limits potential system performance.

Finally, section 5 addresses algorithmic issues in solving  $\{\max, +\}$  equations. The computation of eigenvalues and critical circuits consists of shortest path calculations. The connection between the eigenvalue problem and the search for an optimum mean value circuit in the associated graph for path algebras such as the  $\{\max, +\}$  algebra was first exhibited by Gondran and Minoux [23]. There is a low order of computational complexity. The eigenvalue is important to characterize the

period of the system, i.e. the smallest time horizon such that the initial state is identical to the final state. Having characterized the behavior of the system over such a time horizon by calculating earliest and latest starting times for all activities, many performance evaluation measures can be easily obtained. Section 5.2 provides tools to quantify the influence of the number of resources (tokens in the marked graph) on the length of the period. Section 6 indicates some research needs.

## 2. Timed marked graphs

A timed marked graph can be translated into a system of linear equations in the sense of an algebra where the maximum and addition operations, respectively, represent addition and product operations, respectively, in the usual linear algebra. The approach builds on previous work (Cohen et al. [8,10,11]) in the modeling and analysis of discrete-event systems in the field of manufacturing. It improves and extends earlier attempts by Ramchandani [33], and Ramamoorthy and Ho [32] by providing a systematic treatment of timed marked graphs.

### 2.1. A REFRESHER ON MARKED GRAPHS

A *marked graph* is a directed graph  $G$  where some arcs are assigned a number of tokens. It is a special type of Petri net (Peterson [31]), where each place has exactly one input and one output transition. Places in such Petri nets correspond to arcs of the marked graphs, while transitions correspond to vertices. In the following, we use the terms vertex or transition interchangeably, as well as the terms place and arc.

The firing of a vertex is *enabled* as soon as each input arc (place) of this vertex contains at least one token. Vertices without an input arc are enabled unconditionally. The *firing* consists of removing one token from each input arc and adding one token to each output arc of the vertex. The system evolves over time in this manner.

The vector  $M = (r_1, \dots, r_p)$  (where  $r_\alpha$ ,  $\alpha = 1, \dots, p$ , is the number of tokens on arc number  $\alpha$ ) is called a *marking*. Firing transitions when enabled produce sequences of markings. Marked graphs are useful to describe the behavior of concurrent, asynchronous, deterministic processes, such as computer configurations (Diaz [18], Ajmone Marsan et al. [1], and Smigelski et al. [35]), parallel computations (Reiter [34]), and distributed algorithms (Peterson [31]). Marked graphs do capture concurrency phenomena but are decision-free: decisions on how the system should evolve have been made in advance. The consequences of these decisions can then be evaluated.

Let  $V$  be the set of vertices and  $\hat{A}$  be the set of arcs. When an enabled vertex  $i$  fires, the marking  $M$  evolves into  $M'$  such that

$$\begin{aligned} r'_\alpha &= r_\alpha + 1, & \text{if } \alpha \text{ refers to } (i, j) \text{ for some } j; \\ r'_\alpha &= r_\alpha - 1, & \text{if } \alpha \text{ refers to } (j, i) \text{ for some } j; \\ r'_\alpha &= r_\alpha, & \text{otherwise,} \end{aligned}$$

where  $(i, j)$  is the arc from vertex  $i$  to vertex  $j$ , if it exists.  $M'$  is said to be immediately reachable from  $M$  by  $i$ .

More generally, a marking  $M'$  is said to be *reachable* from  $M$  if there is a sequence of firings at vertices  $i_1, \dots, i_n$  and a sequence of markings  $M_0 = M, M_1, \dots, M_{n-1}, M_n = M'$ , such that  $M_{j+1}$  is *immediately reachable* from  $M_j$  by  $i_j$ . The set of *reachable markings* from  $M_0$  is denoted  $R(G, M_0)$  and includes  $M_0$ . A marked graph is *safe* if  $\forall M \in R(G, M_0)$  and  $\forall \alpha \in \dot{A}$ , then  $r_\alpha \leq 1$ ; if this condition is relaxed to allow  $r_\alpha \leq b_\alpha$ , where  $\{b_\alpha | \alpha \in \dot{A}\}$  is a set of finite upper bounds on the number of tokens on each arc  $\alpha$ , the marked graph is said to be *bounded*. A marked graph is *live* for a marking  $M_0$  if  $\forall i \in V$  there exists  $M \in R(G, M_0)$ , such that  $M$  enables vertex  $i$ . Any vertex satisfying this condition is called a *live vertex*.

A *chain* in  $G$  is a sequence of adjacent vertices. A *path* in  $G$  is a chain,  $i_1, \dots, i_n$ , where  $(i_j, i_{j+1}) \in \dot{A}, \forall j = 1, \dots, n - 1$ . When  $i_1 = i_n$ , it is a *circuit*. When all  $i_j$  are different except for  $i_1 = i_n$ , the chain is called a *simple circuit*.  $G$  is said to be *connected* if there is a chain between any two vertices and *strongly connected* if there is a path between any two vertices. A graph containing no circuit is called *acyclic*.

The following results are known for marked graphs that are strongly connected:

- (1) The number of tokens on a circuit does not change after a vertex fires.
- (2) Liveness is equivalent to the existence of at least one token on every circuit.
- (3) Safeness holds if and only if each place belongs to a circuit containing only one token.
- (4)  $M$  is reachable from  $M'$  if and only if the number of tokens on the circuits is the same for  $M$  and  $M'$ .

## 2.2. MODELING CONVENTIONS FOR TIMED MARKED GRAPHS

Marked graphs can capture the time dimension by assigning durations to arcs and/or vertices and are then more useful to model a class of discrete-event dynamic systems. These systems are described by a set of *activities*, each of which requires a set of *resources* in order to be performed. Any resource that is engaged in some activity cannot be simultaneously used for another. In a manufacturing environment, for example, activities include the processing operations of parts on machines as well as transportation, storage, and machine set-up activities. Resources can be parts, pallets, carts, robots, machines, cutting tools, and the like. Each activity requires a certain amount of time, during which some of the resources are captured. A decision-free, discrete-event system is one such that:

- (i) *for each resource*, the set of activities in which it is involved is linearly ordered, so that when this resource is liberated by some activity, it is known which activity captures it next;

- (ii) for each activity, the set of resources it captures is also ordered, so that resources do not compete.

In terms of Petri nets (i.e. see Peterson [31]), activities can be viewed as transitions. The input places of each transition at times contain tokens, which model resources that are required to start the activity. Condition (i) specifies that no place is an input to more than one transition. Condition (ii) means that no place is an output place of more than one transition. Hence, we have the structure of a marked graph. Note that condition (i) specifies a deterministic system after some chosen systematic rule (such as first-come, first-served) has been used to solve the conflicts between resources. Condition (ii) is also necessary to obtain the useful structure of a marked graph.

Conventions to assign durations to the activities of marked graphs are not unique. The marked graph *representation* of a system having the *least number of vertices* is obtained by the following technique:

- Vertices represent activities that involve more than one resource. The duration  $\theta_i$  of activity  $i$  is assigned to the vertex.
- Activities requiring only one resource are modeled by arcs which connect to other activities requiring more resources. The durations  $t_{ij}$  of single-resource activities are assigned to the corresponding arcs  $(i, j)$ . For other arcs,  $t_{ij} = 0$ .

Our firing conventions are described as follows.

- A token that arrives on some arc  $(i, j)$  is *available* only at time  $t_{ij}$  after the arrival date.
- A vertex  $i$  is fired as soon as it is *enabled* (i.e. it has one available token on each input arc), and becomes *active* during  $\theta_i$  after the firing starts; after a duration of  $\theta_i$ , the token move takes place.
- An active vertex can be fired again as soon as another set of available tokens appears on each of the input arcs. This convention prevents vertices from being considered as a special kind of resource.

The third convention implies that a vertex can be fired, while still active, by the arrival of new tokens.

It is possible to construct graphs having time only on vertices. To see this, every timed arc can be transformed into a one input arc/one output arc vertex, and then the time is specified on the vertex. (In the terminology of PERT/CPM, this is analogous to the "activity-on-node" representation.) Similarly, it is possible to specify time only on arcs. To do this, set every  $\theta_j$  to zero and every  $t_{ij}$  to  $t_{ij} + \theta_j$ . In this case, vertices model events (which are starting points of activities) and we have an "activity-on-arc" type of representation.

For modeling purposes, the latter convention has the drawback of confusing resource-sharing activities and the simpler activities that might follow them. It is clearer to separate them. On the other hand, the activity-on-arc convention is more straightforward and efficient for further computation with the model.

### 2.3. REACHABILITY IN TIMED MARKED GRAPHS

Let  $G$  be a timed marked graph. When  $G$  is acyclic, then the dynamic analysis of  $G$  reduces to the critical path method. Given the earliest starting times on the activities without predecessors, earliest ending times on the activities without successors are calculated. Considering the latest of these ending times as a due date, latest starting times on other activities are thus obtained. Activities with no slack time are called critical and cannot be delayed. Critical paths of activities are thus obtained, and tokens on these paths are critical resources.

The more interesting types of systems are those in which graph  $G$  contains circuits. Then some activities are performed repetitively. A systematic study of these types of decision-free, discrete-event systems is not readily available to date, although some authors have obtained some insight into such problems (see, for example, Ramchandani [33], and Ramamoorthy and Ho [32]). Timed Petri nets have been studied in some detail (e.g. Chretienne [7]).

Note that the concept of reachability of markings is not adequate to specify a timed marked graph. Indeed, the reachability set  $R(G, M_0)$  consists of all possible markings that can potentially be reached from  $M_0$ , regardless of activity durations. From the previous section, starting with a bounded initial marking  $M_0$  on a strongly connected graph  $G$ , it is clear that  $R(G, M_0)$  is finite. When several vertices  $i, j, k, \dots$ , are enabled under marking  $M_0$ , then there is no information about which vertex should fire first, and all markings  $M_i, M_j, M_k, \dots$ , obtained by firing vertices  $i, j, k, \dots$ , are calculated and put into  $R(G, M_0)$ . With a timed marked graph, the vertices enabled by  $M_0$  are ordered by their firing dates, which depend upon the availability dates of tokens on input arcs. As a consequence, it is necessary to distinguish between  $R(G, M_0)$  and the set of markings which will eventually be obtained from  $M_0$ .

Let  $T = \{\theta_i, i \in V\} \cup \{t_{ij}, (i, j) \in \dot{A}\}$ . Denote by  $R(G, T, M_0)$  the set of *eventually reached* markings from  $M_0$ , i.e. markings that are actually obtained at some point in time, given the graph structure  $G$  and the time allocation  $T$ . Clearly,  $\forall \text{set } T, R(G, M_0) \supseteq R(G, T, M_0)$ . However, the inclusion may be strict, as the following example shows.

#### Example 1

In fig. 1, crosses are vertices, arrows are arcs, and dots on arrows are tokens. The arcs are numbered lexicographically, i.e. if arc  $\alpha = (i, j)$  and  $\alpha' = (i', j')$ , then  $\alpha < \alpha'$  means that either  $i < i'$  or  $i = i'$  and  $j < j'$ . Then, from fig. 1, the initial marking is  $M_0 = (1, 1, 0, 1, 0)$ . The set  $R(G, M_0)$  is pictured in fig. 2. The times that are associated with the arcs are always positive. Arrows contain the names (numbers) of the vertices that are firing over time.

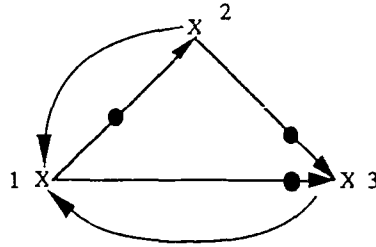


Fig. 1.  $M_0 = (1, 1, 0, 1, 0)$ .

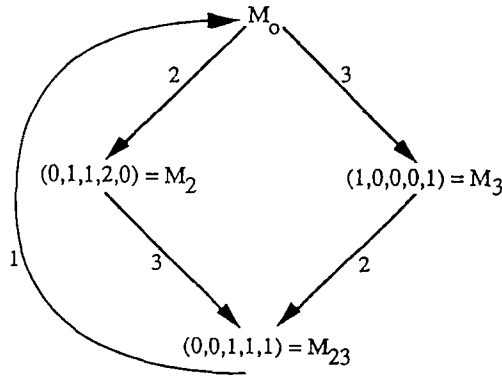


Fig. 2.  $R(G, M_0)$ .

Let  $t_{12} > t_{13} > 0$  and  $t_{ij} = 0$ , otherwise. In the initial state  $M_0$ , the tokens are allocated at time  $t = 0$  and are not available yet. Then the sequence of firing vertices is 3, 2, 1, 3, 2, 1, . . . , i.e.  $R(G, T, M_0) = \{M_0, M_3, M_{23}\}$ . If  $t_{12} < t_{13}$ , then this firing sequence would be 2, 3, 1, 2, 3, 1, . . . and  $R(G, T', M_0) = \{M_0, M_2, M_{23}\}$ , where  $T'$  is the set of new time allocations.  $\square$

Assume now that  $G$  is strongly connected and that  $R(G, M_0)$  is a directed graph whose nodes are markings and arcs express immediate reachability. Moreover, when  $G$  is live for  $M_0$  and  $M_0$  is bounded, this reachability graph contains circuits.  $R(G, T, M_0)$  is one path in  $R(G, M_0)$ . It contains a circuit if  $G$  is live for  $M_0$ , since the number of possible markings is finite and the system evolves forever. Indeed, the liveness of  $G$  for  $M_0$  prevents any blocking from occurring.

However,  $R(G, M_0)$  contains enough markings so that for any  $T$ , and any set of live markings  $M_0$ , all vertices in  $G$  eventually fire:



## PROPOSITION 1

If  $G$  is strongly connected and live for a bounded marking  $M_0$ , then  $\forall T$  and  $\forall i \in V$ , there exists a marking  $M_i \in R(G, T, M_0)$  such that  $M_i$  fires  $i$ .

*Proof*

If vertex  $i$  never fires, then no token on any arc  $(j, i)$  shall ever cross  $i$ . Hence, no vertex  $k$  such that  $(i, k)$  is an arc is eventually enabled. That is, no vertex in  $V$  is eventually enabled. Hence, the immediate reachability relation of  $R(G, T, M_0)$  makes it a path with no circuit. Hence,  $M_0$  is not live, which contradicts the assumption.  $\square$

Note that strong connectedness is required to ensure that all vertices in  $V$  are eventually enabled. In section 3, graphs  $G$  are considered in which every vertex has at least one input and one output arc. Graphs that contain vertices without input arcs (sources) are examined in section 3.3, as well as graphs having vertices with no output arcs (sinks).

### 3. Algebraic representation

Given an initial marking  $M_0$  and a set of availability dates for tokens, it is easy to simulate the behavior of a decision-free, discrete-event system (viewed as a marked graph), by firing the vertices at the proper points in time. We now indicate how to represent this evolution algebraically.

#### 3.1. DERIVING RECURRENCE EQUATIONS

Let  $x_i(n)$  be the earliest firing date of transition  $i$  for the  $n$ th time. Let  $a_k$ ,  $k = 1, \dots, R$ , where  $R = \sum_{\alpha=1}^p r_\alpha$ , be the availability date of each token in the initial marking  $M_0$ . Let  $M_0$  be a live marking. Without loss of generality, we can assume that  $\theta_i = 0$ ,  $\forall i \in V$  (by adopting the activity-on-arc convention).

Let  $\Gamma^-(i) = \{j \mid (j, i) \in A\}$  be the set of predecessors of vertex  $i$ . Then the following inequality holds:

$$x_i(n) \geq x_j(n - r_{ji}) + t_{ji}, \quad \forall n > r_{ji} \text{ and for } j \in \Gamma^-(i). \quad (1)$$

Initially, there are  $r_{ji}$  tokens on arc  $(j, i)$ . However, transition  $i$ 's firing for the  $n$ th time is conditioned *only* by the first availability of the  $n$ th token. This token corresponds to firing transition  $j$  for exactly  $(n - r_{ji})$  times.

Values  $x_j(l)$ ,  $l = 1, \dots, r_{ij}$ , are calculated from the knowledge of the availability dates of the  $r_{ij}$  tokens in the initial marking  $M_0$ . The initial marking is used as a landmark for numbering the transition firings. From eq. (1), for all transitions and

noticing that no other constraint impacts the value of  $x_i(n)$ , we obtain the following result:

PROPOSITION 2

$$\forall i \in V, x_i(n) = \max_{j \in \Gamma^-(i)} x_j(n - r_{ji}) + t_{ji}. \tag{2}$$

We adopt the following notational conventions:

- $X(n) = (x_1(n), \dots, x_{|V|}(n))$  is the vector of  $n$ th earliest firing dates of vertices.
- Let  $\oplus$  and  $*$  be scalar operations, such that  $a \oplus b = \max\{a, b\}$  and  $a * b = a + b, \forall a, b \in \mathbb{R}$ .
- $\oplus$  and  $*$  can be extended to square matrices  $A$  and  $B$  as:  $(A \oplus B)_{ij} = \max\{A_{ij}, B_{ij}\}$  and  $(A * B)_{ij} = \max_k \{A_{ik} + b_{kj}\}$ , where  $A_{ij}$  is the  $(i, j)$ th entry of  $A$ . We write  $AB$  instead of  $A * B$ .

It is easy to see that the following are true:

- $\oplus$  equips  $\mathbb{R}$  with a commutative semi-group structure;
- $\oplus$  is an idempotent operation ( $a \oplus a = a$ );
- the identity is  $-\infty$ , denoted by  $\varepsilon$ ;
- $*$  is distributive over  $\oplus$ :  $[a * (b \oplus c) = (a * b) \oplus (a * c)]$ ;
- the identity of the commutative group  $(\mathbb{R}, *)$  is denoted  $e (= 0)$ ;
- $\varepsilon$  is an absorbing element for  $*$  [ $\varepsilon * a = \varepsilon, \forall a$ ].

$(\mathbb{R}, \oplus, *)$  is an algebraic structure known as a dioïd (see Kuntzmann [27] and Gondran [22]). Note that most of these properties (except for the commutativity property of  $*$ ) carry over to the set of real square matrices. The identity of  $*$  is then  $E$ , such that  $E_{ij} = \varepsilon$ , if  $i \neq j$  and  $E_{ii} = e, \forall i$ . The absorbing element is  $\theta$ , such that  $\theta_{ij} = \varepsilon, \forall i$  and  $j$ .

Let us introduce a sequence  $F_0, F_1, \dots, F_\Delta$  of  $|V| \times |V|$  matrices defined by:

$$(F_r)_{ij} = \begin{cases} t_{ij}, & \text{if there exists an arc } (i, j) \in \mathring{A} \text{ such that } r_{ij} = r; \\ \varepsilon, & \text{otherwise.} \end{cases}$$

With these conventions, eq. (2) can be written as (Cohen et al. [11]):

$$X(n) = X(n)F_0 \oplus X(n-1)F_1 \oplus \dots \oplus X(n-\Delta)F_\Delta, \quad \forall n > \Delta. \tag{3}$$

The vectors  $X(1), \dots, X(\Delta)$  are easily calculated from the availability date of tokens in the initial marking.

Some comments are needed to discuss this representation:

- (i)  $\Delta$  is the largest number of tokens initially on any arc.
- (ii) The earliest firing date of a transition  $i$  for the first time is conditioned by the availability date of the tokens on arcs  $(j, i)$ , i.e.  $x_i(1) \geq a_k$  for token  $k$  on arc  $(j, i)$  in the initial marking.
- (iii) The decision to use the vertex firing dates as the state variables is justified by the fact that, in general, the number of vertices is smaller than the number of arcs.
- (iv) The  $x_i(n), i \in V, n = 1, \dots, \Delta$  are generally not independent, so that eq. (3) does not provide a minimal representation.

### 3.2. EVOLUTION OF THE SYSTEM

In order to be able to simulate the evolution of the system over time, it is necessary to express  $X(n)$  in terms of  $X(n')$  for  $n' < n$ . That is, we need to solve eq. (3) with the  $X(n)$  as unknowns and viewing (3) under the form  $X(n) = X(n)F^0 \oplus M$ . To do this, we introduce the directed graph  $G(A)$  associated with a square matrix  $A$ . The graph is defined by:

- $V(A)$  = the set of rows of  $A$ ;
- $\mathring{A}(A) = \{(i, j) \mid A_{ij} > \varepsilon\}$ ;
- $A_{ij}$  = the weight on arc  $(i, j)$ .

Moreover, the weight of a path is the sum (in the usual sense, i.e. the "product" in the algebra) of the weights on the path's arcs. The length of the path is the number of its arcs. The following result is known (Gondran and Minoux [24]):

#### PROPOSITION 3

Equation  $X = XA \oplus B$  has a solution if and only if no circuit of  $G(A)$  has a weight greater than  $\varepsilon = 0$ . Then, the unique solution is:

$$X = BA^*, \text{ where } A^* = E \oplus A \oplus A^2 \oplus \dots \oplus A^n \oplus \dots, \text{ and } A^2 = AA, A^3 = AAA, \dots$$

The matrix  $A^*$  may be generated by repeatedly squaring  $E \oplus A$  until (after a finite number of squarings) convergence is observed. Other, more efficient methods exist (Carré [6]).  $A^*$  may also be generated by substituting  $XA \oplus B$  for  $X$  infinitely many times into the right-hand side of eq. (3). In our context, the entries  $A_{ij}$  are positive, since they are the durations associated with activities. Circuits in  $G(A)$  have positive weights, if any. In this case, the existence condition in proposition 2 reduces to the non-existence of circuits in  $G(A)$  (except for 0-weight ones). Moreover, only a finite number of computations is required to obtain  $A^*$ , since  $A^*$  can be obtained

by any longest path algorithm, e.g. Floyd's algorithm [21]. More precisely, it is enough to compute  $\{A^n \mid n \leq |V(A)|\}$ , where  $|V(A)|$  is the number of rows in  $A$ .

Now eq. (3) can be written as:

$$X(n) = X(n - 1)F_1(F_0)^* \oplus \dots \oplus X(n - \Delta)F_\Delta(F_0)^*, \tag{4}$$

provided that  $G(F_0)$  contains no circuit except 0-weight circuits. This can be seen once it is noticed that  $G(F_0)$  is a subgraph of  $G$  obtained by deleting all arcs  $(i, j)$  with  $r_{ij} > 0$ . Hence, eq. (4) holds, for example, when  $G$  is strongly connected and the initial marking  $M_0$  is live. Then, every circuit in  $G$  contains at least one token and every circuit is broken in  $G(F^0)$ .

In general,  $G$  can be decomposed into strongly connected components. To do this, consider the equivalence relation  $C$  on  $V = iCj \Leftrightarrow$  there is a circuit containing  $i$  and  $j$ ; the strongly connected components are the equivalence classes of  $C$ . They form a partial ordering and can include isolated vertices. Now eq. (4) holds when the initial marking restricted to each strongly connected component is live for this strongly connected component, that is, when it is not an isolated node.

Note that when  $G$  is strongly connected and  $M_0$  is not live, there is something wrong in the system and it cannot be feasible. Indeed, there is a circuit  $\gamma$  that does not contain a token. Because the number of tokens on each circuit always remains the same as the marking evolves, then no vertex on circuit  $\gamma$  will ever be enabled, even if this circuit has 0-weight. Hence, we must rule out the latter case, even though  $(F_0)^*$  still exists.

### 3.3. COMMUNICATION OUTSIDE THE SYSTEM

So far, only isolated marked graphs have been described. However, it is possible to also consider the case when some tokens are supplied from outside, and dually, when some tokens are released from within the system.

Let  $U$  be the set of vertices without predecessors, which then act as token sources. Let  $Y$  be the set of vertices without successors, which can act as token sinks. Let  $u_i(n)$  be the firing date of vertex  $i \in U$  for the  $n$ th time. Let  $B$  be the  $|U| \times |V|$  matrix describing the link between input vertices and interior vertices:

$$B_{ij} = \begin{cases} e, & \text{if arc } (i, j) \text{ exists;} \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Then eq. (3) becomes:

$$X(n) = \sum_{j=0}^{\Delta} X(n-j)F_j \oplus U(n)B, \tag{5}$$

where  $U(n)$  is the vector  $(u_1(n), u_2(n), \dots, u_{|U|}(n))$ . More generally, if there are tokens in the input arcs (i.e.  $(i, j)$  with  $i \in U$ ), eq. (5) becomes (Cohen et al. [11]):

$$X(n) = \sum_{j=0}^{\Delta} X(n-j)F_j \oplus \sum_{k=0}^K U(n-k)B_k,$$

where  $B_k$  pertains to input arcs containing  $k$  tokens in the initial marking, i.e.  $(B_k)_{ij} = e$ , if  $(i, j)$  exists and  $r_{ij} = k$ ; and  $=\varepsilon$ , otherwise.

Similarly,  $y_i(n)$  denotes the firing date of vertex  $i \in Y$  for the  $n$ th time.  $C$  is the  $|V| \times |Y|$  matrix expressing the links between  $V$  and  $Y$ , i.e.  $C_{ij} = e$ , if  $(i, j)$  exists and  $=\varepsilon$ , otherwise. Then, introducing  $Y(n) = (y_1(n), y_2(n), \dots, y_{|Y|}(n))$ , the output map of the system is described by  $Y(n) = X(n)C$  or, more generally,

$$y(n) = \sum_{l=0}^L X(n-l)C_l,$$

$$(C_l)_{ij} = \begin{cases} e, & \text{if } i = j \text{ and } r_{ij} = l; \\ \varepsilon, & \text{otherwise.} \end{cases}$$

The set  $U$  can be viewed as a set of control vertices and  $Y$  as a set of observation vertices. A timed marked graph can be defined as *controllable*, if each transition of  $V$  can be sent tokens by some input  $i \in U$ . More formally, for any  $j \in V$ , there is an  $i \in U$  such that a path exists from  $i$  to  $j$ . Otherwise, if no path exists from any  $i \in U$  to some  $j \in V$  and tokens come only from inputs, then vertex  $j$  is not live.

Dually, a timed marked graph is *observable*, if for any  $i \in V$  there is an output vertex  $j \in Y$  such that a path exists from  $i$  to  $j$ . If there is some  $i \in V$  that is linked to no  $j \in Y$  by a path, the tokens in the input places of  $i$  are trapped in the system.

We note that controllability (or observability) can be equivalently defined by the existence of a path from some input (or each strongly connected component) to each strongly connected component (or some output) of  $G$ .

One example of a controllable and observable marked graph is a graph that consists of sources, interior vertices, and sinks and is an acyclic graph. Another example is a graph  $G$  consisting of interior vertices that is strongly connected. Otherwise, the marked graph may fail to satisfy the controllability and observability properties.

### 3.4. THE CHOICE OF A STATE SPACE

In order to be able to solve the recurrent equations, one possibility is to transform eq. (4) into a first-order system. To achieve this, the most straightforward technique is to increase the size of the state vector  $X(n)$ , i.e. let  $\chi(n-1) = (X(n-1), X(n-2), \dots, X(n-\Delta))$ . Then eq. (4) becomes:

$$\chi(n) = \chi(n - 1)H, \tag{6}$$

where

$$H = \left[ \begin{array}{c|c} F_1(F_0)^* & E \\ F_2(F_0)^* & \epsilon \\ \vdots & \epsilon \\ \vdots & E \\ \hline F_\Delta(F_0)^* & \epsilon \end{array} \right]$$

The problem with eq. (6) is that the vector  $\chi$  is too large. However, many of its components are dependent. It would be interesting and useful to reduce the number of variables necessary to describe the behavior of the timed marked graph.

Assume  $G$  is strongly connected and that the initial marking  $M_0$  is live. Let  $\dot{A} \supseteq \dot{A}^*$  be the set of marked arcs,  $I = \{j \in V \mid (i, j) \in \dot{A}^*\}$ , and  $F = \{j \in V \mid (j, i) \in \dot{A}^*\}$ .  $I$  ( $F$ ) is called the set of *initial vertices* (set of *final vertices*), respectively. Note that the matrix associated with  $G_0 = (V, \dot{A} - \dot{A}^*)$  is  $F_0$ .

Two cases are now considered:

*Case (a):* For every arc  $(i, j) \in \dot{A}^*$ ,  $(i, j)$  contains only one token.

In this case, eq. (4) reduces to:

$$X(n) = X(n - 1)F_1(F_0)^*. \tag{7}$$

Using  $I$  and  $F$  as input and output vertices, respectively, associated with subvectors  $\hat{X}(n)$  and  $\underline{X}(n)$  of  $X(n)$ , we have:

$$X(n) = \hat{X}(n) B(F_0)^*, \tag{8}$$

where  $B$  expresses the canonical injection of  $I$  in  $V$ , that is

$$(B)_{ij} = \begin{cases} 0, & \text{if } \hat{X}_i(n) \text{ and } X_j(n) \text{ correspond to the same vertex in } G, \\ -\infty, & \text{otherwise.} \end{cases}$$

Equation (8) says that only the knowledge of  $\hat{X}(n)$  (components of  $X(n)$  from  $I$ ) is required to reconstruct  $X(n)$ .

Similarly, we have:  $\underline{X}(n) = X(n)C$ , where  $C$  is the matrix associated with the restriction of  $V$  to  $F$ . Hence, from eq. (8) we obtain:

$$\underline{X}(n) = \hat{X}(n) B(F_0)^* C. \tag{9}$$

Now the feedback equation is:

$$\hat{X}(n) = \underline{X}(n - 1)K = X(n - 1)CF_1B,$$

where  $K$  is an  $|F| \times |G|$  matrix with  $K_{ij} = t_{ij}$ , if  $(i, j) \in \hat{A}^*$ ; and  $= -\infty$ , otherwise.  $K$  is built from  $F_1$  by selecting rows corresponding to  $F$  and columns corresponding to  $I$ . Using (9), we obtain the following reduced form of eq. (7):

$$\hat{X}(n) = \hat{X}(n - 1)B(F_0)^*CK \tag{10}$$

or

$$\underline{X}(n) = \underline{X}(n - 1)KB(F_0)^*C. \tag{11}$$

Case (b): Marked arcs may contain more than one token.

If arc  $(i, j) \in \hat{A}^*$  contains  $r_{ij} > 1$  tokens, then we can add  $r_{ij} - 1$  vertices to the graph  $G$  and  $r_{ij} - 1$  arcs will bear one token each, as shown in fig. 3. In

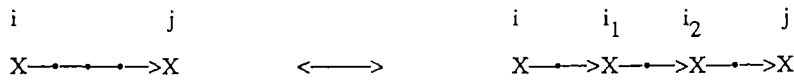


Fig. 3. Arc  $(i, j)$  of graph  $G$  containing 3 tokens.

particular, let  $i_1, i_2, \dots, i_k$  for  $k = r_{ij} - 1$ , be the new vertices, and  $(i, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k), (i_k, j)$  be the new arcs, each marked with exactly one token (see fig. 3).  $\hat{A}^*$  is then completed by adding these arcs and deleting arc  $(i, j)$ . In addition, the new vertices  $i_1, \dots, i_k$  are added to both  $F$  and  $I$ .

To mathematically quantify case (b), we begin with:

$$x_j(n) = \max\{x_i(n - r_{ij}) + t_{ij}, a\},$$

where  $a$  contains terms pertaining to other vertices preceding  $j$ . As we add vertices, we obtain:

$$x_j(n) = \max\{x_{i_k j}(n - 1) + t_{i_k j}, a\},$$

$$x_{i_k}(n) = x_{i_{k-1} i_k}(n - 1) + t_{i_{k-1} i_k}$$

⋮

$$x_{i_1}(n) = x_{i i_1}(n - 1) + t_{i i_1},$$

where

$$t_{ijk} = t_{ik-1ik} = \dots = 0 \quad \text{and} \quad t_{ii1} = t_{ij}.$$

In other words, we have introduced  $r_{ij} - 1$  new variables (the new vertices) into the algebraic representation.

This technique enables matrices  $F_2, F_3, \dots, F_\Delta$  in eq. (4) to vanish. They are transformed into new rows for matrix  $F_1$ , by using these new variables. Then we are in the situation described by case (a).

This method of transforming higher-order delays into first-order delays points out the fact that the number of independent state variables is bounded from above by the number of tokens. It is then useful to consider as variables the times when each token returns to its initial arc.

Let  $\theta$  be the set of tokens and  $z_i(n)$  be the time when token  $i \in \theta$  is available again on its initial arc for the  $n$ th time. Using the technique of adding vertices, we can always assume without loss of generality that in the initial marking, each arc bears at most one token.

In eq. (10), let  $B$  (or  $C$ ) become a  $|\theta| \times |V|$  (or  $|V| \times |\theta|$ ) matrix, where

$$B_{ij} = \begin{cases} 0, & \text{if token } i \text{ is on some input arc of vertex } j; \\ -\infty, & \text{otherwise.} \end{cases}$$

$$C_{ji} = \begin{cases} 0, & \text{if token } i \text{ is on some output arc of vertex } j; \\ -\infty, & \text{otherwise.} \end{cases}$$

Let  $K$  be a diagonal matrix, with  $K_{ii} = t_{jk}$ , if token  $i$  is on arc  $(j, k)$ . Then eq. (10) becomes:

$$Z(n) = Z(n - 1)B(F_0)^*CK. \tag{12}$$

Now each line or row of matrix  $B(F_0)^*CK$  corresponds to a resource. This property will prove to be useful when we subsequently discuss performance evaluation issues.

### 3.5. AN ILLUSTRATIVE EXAMPLE

An example is now provided to demonstrate the algebraic representations and manipulations. The Petri net (marked graph) of fig. 4 consists of 4 vertices, 7 arcs, and an initial marking  $M_0$  of 4 tokens distributed among the arcs. The arcs  $(i, j)$  are labeled with the durations of the activities  $t_{ij}$ .

The algebraic representation of this graph,  $x_i(n)$ ,  $i = 1, \dots, 4$ , is provided by eq. (2).



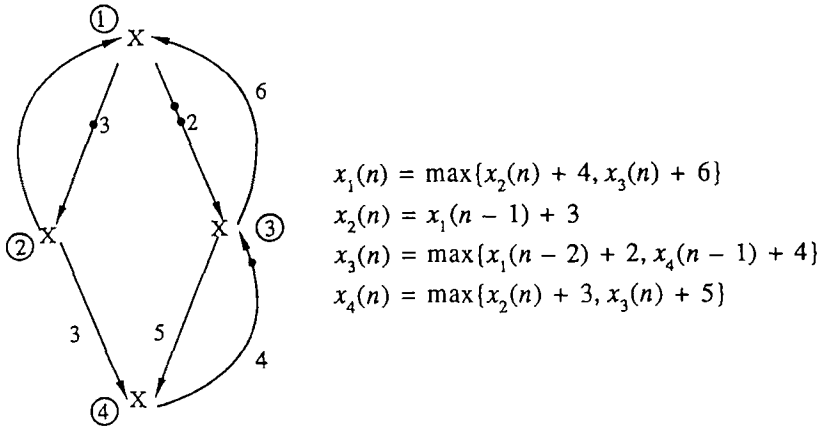


Fig. 4. Petri net example with algebraic representation.

The equations of fig. 4 directly define the matrices  $F_0, F_1,$  and  $F_2$  as defined by eq. (3):

$$F_0 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ 4 & \epsilon & \epsilon & 3 \\ 6 & \epsilon & \epsilon & 5 \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

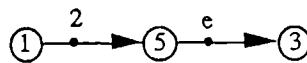
$$F_1 = \begin{bmatrix} \epsilon & 3 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 4 & \epsilon \end{bmatrix}$$

$$F_2 = \begin{bmatrix} \epsilon & \epsilon & 2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

$(F_0)^*$  is computed as:

$$(F_0)^* = \begin{bmatrix} e & \epsilon & \epsilon & \epsilon \\ 4 & e & \epsilon & 3 \\ 6 & \epsilon & e & 5 \\ \epsilon & \epsilon & \epsilon & e \end{bmatrix}$$

Arc (1, 3) is the only arc containing two tokens. It is transformed into two arcs:



With this transformation, the sets of initial and final vertices are:

$$I = \{2, 3, 5\} \text{ and } F = \{1, 4, 5\}.$$

The expanded matrices  $(F_0)^*$  and  $F_1$  are:

$$(F_0)^* = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 4 & e & \varepsilon & 3 & \varepsilon \\ 6 & \varepsilon & e & 5 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \quad F_1 = \begin{bmatrix} \varepsilon & 3 & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & 4 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix}$$

Matrices  $B$ ,  $C$ , and  $K$  are defined:

$$B = \begin{bmatrix} \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \quad C = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix} \quad K = \begin{bmatrix} 3 & \varepsilon & 2 \\ \varepsilon & 4 & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}$$

Multiplying these matrices provides:

$$B(F_0)^*C = \begin{bmatrix} 4 & 3 & \varepsilon \\ 6 & 5 & \varepsilon \\ \varepsilon & \varepsilon & e \end{bmatrix} \quad \text{and} \quad B(F_0)^*CK = \begin{bmatrix} 7 & 7 & 6 \\ 9 & 9 & 8 \\ \varepsilon & e & \varepsilon \end{bmatrix}.$$

With  $\hat{X}(n) = (x_2(n), x_3(n), x_5(n))$ , we obtain from eq. (9):

$$\begin{aligned} x_2(n) &= \max\{x_2(n-1) + 7, x_3(n-1) + 9\}, \\ x_3(n) &= \max\{x_2(n-1) + 7, x_3(n-1) + 9, x_5(n-1)\}, \\ x_5(n) &= \max\{x_2(n-1) + 6, x_3(n-1) + 8\}. \end{aligned}$$

If we use the token space, tokens are numbered as follows:

- No. 1 on arc (1, 2)
- No. 2 on arc (1, 5)
- No. 3 on arc (4, 3)
- No. 4 on arc (5, 3).

These token markings define matrices  $B, C, K$ :

$$B = \begin{bmatrix} \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix} \quad
 C = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \quad
 K = \begin{bmatrix} 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 4 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix}$$

Multiplying the matrices provides:

$$B(F_0)^* C = \begin{bmatrix} 4 & 4 & 3 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \\ 6 & 6 & 5 & \varepsilon \\ 6 & 6 & 5 & \varepsilon \end{bmatrix} \quad \text{and} \quad B(F_0)^* CK = \begin{bmatrix} 7 & 6 & 7 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 9 & 8 & 9 & \varepsilon \\ 9 & 8 & 9 & \varepsilon \end{bmatrix}$$

Note that in the token space,  $B(F_0)^* CK$  has 2 rows and 2 columns which are identical, thus indicating the redundancy of the token comeback variables  $z_i(n)$ , i.e. the token space does not provide minimal representation. In particular, the input vertices (or the output vertices) lead to a smaller state space. The problem of finding the minimal representation is a topic for further research. We conjecture that it might be solved by finding minimal cuts in the graph. The reason is that one needs to find a minimal set of vertices that break all of the circuits in the marked graph (e.g. here vertices 1 and 3). Moreover, the initial marking and the number of tokens must also affect the minimal representation. Then, it seems to us that minimal cuts should play a role in finding the minimal representation.

More precisely, in order to obtain a closed form such as (10), (11) or (12), we must make sure that by deleting the initially marked arcs, a circuit-free graph is obtained. This is possible only if the initial marking  $M_0$  is live. However, any marking  $M$  in the reachability graph  $R(G, T, M_0)$  can serve as an initial marking for the closed form representation at steps  $n > n_M$  such that  $M$  is attained after  $n_M$  firings of the vertices.

Let  $A^*(M)$  be the set of marked arcs for marking  $M$ ,  $I(M)$  be the input vertices for marking  $M$ ,  $F(M)$  be the output vertices for marking  $M$ , and  $\tau$  be the number of

tokens in the network. Then, the dimension (i.e. the number of independent variables) of the minimal representation of  $(G, T, M_0)$  is conjectured to be:  $\max\{\tau, \min[|I(M)|, M \in R(G, T, M_0)], \min[|F(M)|, M \in R(G, T, M_0)]\}$ , where  $|A|$  denotes the cardinality of  $A$ . It may correspond to the token space, the input vertex space, or the output vertex space.

### 3.6. THE DUAL SYSTEM

Let  $v_r$  denote the latest time that token  $r$  is available in  $G$ . In other words, we define a final marking  $M_\infty$  (due dates, for example) to be reached at time  $t_\infty$ , with  $v_r \leq t_\infty, r \in \theta$ . From these data, it is possible to simulate or evolve backwards to determine the behavior of the timed marked graph  $G$  and to calculate the latest firing times  $q_i$  of vertices which satisfy a final marking constraint. The following equality holds (Cuninghame-Green [16] and Cohen et al. [8]):

$$\forall i, \quad q_i(n) = \min_{j \in \Gamma^+(i)} \{q_j(n + r_{ij}) - t_{ij}\}. \tag{13}$$

$\Gamma^+(i) = \{j \mid (i, j) \in A\}$  and is the set of successors of vertex  $i$ . Equation (13) is the dual of eq. (2). Denoting  $Q_i(n) = -q_i(n)$ , eq. (13) is transformed into:

$$\forall i, \quad Q_i(n) = \max_{j \in \Gamma^+(i)} \{Q_j(n + r_{ij}) + t_{ij}\}.$$

We have returned to the  $\{\max, +\}$  algebra. Dually of eq. (3), we have:

$$Q(n) = F_0 Q(n) \oplus F_1 Q(n + 1) \oplus \dots \oplus F_\Delta Q(n + \Delta), \tag{14}$$

where  $\Delta = \max_{ij} \{r_{ij}\}$  and  $Q(n)$  is a column vector containing the variable  $Q_i(n)$ . Matrix multiplication is now on the right-hand side because the successors of node  $i$  are used in eq. (13). The backwards simulation from (14) requires first its solution in terms of  $Q(n)$ :

$$Q(n) = (F_0)^* F_1 Q(n + 1) \oplus \dots \oplus (F_0)^* F_\Delta Q(n + \Delta), \tag{15}$$

and then the knowledge of  $Q(n), Q(n + 1), \dots, Q(n + \Delta - 1)$ , for some  $n$ .

Moreover, by introducing token variables  $\zeta_r(n)$  that represent the opposite of the latest arrival date of token  $r$  on its final arc (as in the final marking), it is possible to transform eq. (15), when  $\Delta = 1$ , into the dual of (12):

$$\zeta(n) = B(F_0)^* CK \zeta(n + 1), \tag{16}$$

where  $B$  is the token-to-vertex interface matrix,  $C$  is the vertex-to-token interface matrix, and  $K$  is a feedback matrix. When  $\Delta > 1$ , it is possible to transform to eq. (16) by adding a sufficient number of dummy vertices, as in section 3.4, case (b).

#### 4. Solving {max, +} recurrence equations

Section 3 provides a linear representation of timed marked graphs in a non-standard algebra. The usefulness of this representation depends upon whether enough can be developed about this algebra to provide interesting results and analyses from the linear representation. This algebra has been studied by several researchers, such as Cuninghame-Green [14,15], Gondran and Minoux [24], and Cohen et al. [8]. Results on more general algebraic structures can be found in Zimmermann [36], and Gondran and Minoux [25]. In particular, Zimmermann discusses eigenvalue problems in ordered algebraic structures. Here, we summarize and extend some of the results contained in the latter references concerning the study of {max, +} systems of recurrence equations.

##### 4.1. ALGEBRAIC SETTING

Let  $M$  be a square  $m \times m$  matrix and  $G(M)$  the associated graph as described in section 3.2.  $M$  is said to be *irreducible* if  $G(M)$  is strongly connected, which is assumed here. A study of recurrent equations of the form  $Z(n) = Z(n-1)M$  naturally leads to a study of eigenvalues and eigenvectors. An eigenvalue is a quantity  $\lambda$  such that  $Z(n-1)M = \lambda Z(n-1)$ , where  $\lambda Z(n-1)$  is obtained here by adding  $\lambda$  to each component of  $Z(n-1)$ . This behavior implies that  $Z(n)$  is obtained from  $Z(n-1)$  by a simple translation in time, without moving the relative location of events corresponding to the components of  $Z(n-1)$ . Hence, any real number  $\lambda$  and vector  $X = (x_1, \dots, x_m)$  will be called an eigenvalue and eigenvector of  $M$ , respectively, on the left, if:

$$\lambda X = XM. \quad (17)$$

It can be proven (see Gondran and Minoux [23], section 3, lemma 1) that  $\lambda > \varepsilon$  and  $x_i > \varepsilon, \forall i$ . Moreover, given  $M$ ,  $\lambda$  is unique and has a nice graph-theoretical interpretation. Namely,  $\lambda$  is the *maximal average weight of circuits* in  $G(M)$ . The *average weight*  $\bar{W}(\gamma)$  of a circuit  $\gamma$  in  $G(M)$  is defined as the ratio of the weight of the circuit by its length  $l(\gamma)$ :

$$\bar{W}(\gamma) = \frac{W(\gamma)}{l(\gamma)}. \quad (18)$$

In the notation of the algebra, where maximum represents addition and addition represents product, one would write eq. (18) as

$$\overline{W}(\gamma) = {}^{l(\gamma)}\sqrt{W(\gamma)}$$

(see Ramchandani [33]).

The maximum of  $\overline{W}(\gamma)$  exists and is attainable due to the finiteness of the graph  $G(M)$  (and  $M_{ij} < +\infty, \forall i, j$ ). It is easy to show that only simple circuits (with length  $l(\gamma) \leq m$ ) need be considered. Let  $W^* = \max_{\gamma} \{\overline{W}(\gamma), l(\gamma) \leq m\}$  be the maximal average weight of circuits. We have  $W^* = \lambda$ , an eigenvalue of  $M$ . Any circuit  $\gamma_0$  such that  $\overline{W}(\gamma_0) = \lambda$  is called a critical circuit. The graph  $G_0(M) = (V_0(M), \dot{A}_0(M))$ , obtained as the union of all critical circuits, is called the critical graph associated with  $M$ .

The set of eigenvectors of  $M$  has been characterized as follows (see Gondran and Minoux [23], and Cohen et al. [8]): Let

$$M^+ = M \cdot M^* = M \oplus M^2 \oplus \dots \oplus M^n \oplus \dots$$

$M^+$ , as  $M^*$ , exists only if no positive weight circuit exists. In that case,

$$M^* = E \oplus M \oplus M^2 \oplus \dots \oplus M^{m-1}.$$

Hence,  $M^+ = M \oplus M^2 \oplus \dots \oplus M^m$ . Given  $M$  with eigenvalue  $\lambda$ , define  $M_\lambda$  such that  $(M_\lambda)_{ij} = M_{ij} - \lambda, \forall i, j$ , which can be written:  $M_\lambda = \lambda^{-1}M$ . Since  $\lambda$  is the maximal average weight of circuits in  $G(M)$ , the most weighted circuit in  $G(M_\lambda)$  has weight 0 and  $(M_\lambda)^+$  exists. It is proven in Cuninghame-Green [16] (and also shown in Gondran and Minoux [23], and Cohen et al. [8]) that the set  $V_0$  of rows of  $(M_\lambda)^+$ , corresponding to vertices of the critical graph, are eigenvectors of  $M$ , and are enough to generate the eigenspace associated with  $M$ . That is, if  $y$  is an eigenvector of  $M$ , there is a set of real numbers  $\{\mu_i | i \in V_0\}$  such that

$$y = \sum_{i \in V_0} \mu_i (M_\lambda)_i^+, \tag{19}$$

where the  $\sum$  means a maximum and  $M_i$  denotes row  $i$  of  $M$ . Moreover,  $\mu_i = y_i, \forall i \in V_0$ . This is because any eigenvector of  $M$  is also an eigenvector of  $M^+$ .

The structure of the eigenspace depends upon the number of connected components  $C_1, \dots, C_k$  of the critical graph  $G_0(M)$ . It is shown that (Cuninghame-Green [16], and Cohen et al. [8]):

- If  $i$  and  $j$  are vertices in  $V_0$  belonging to the same connected component, then  $(M_\lambda)_i^+$  and  $(M_\lambda)_j^+$  are collinear. That is, there exists a scalar  $\mu$  such that  $(M_\lambda)_i^+ = \mu (M_\lambda)_j^+$ .

Hence, choosing one vertex  $i_l$  in each connected component  $C_l$ , we obtain a basis of  $k$  independent generators of the eigenspace;  $k$  is called the *rank* of matrix  $M$ . Note

that the  $\{\max, +\}$  algebra does not behave as a regular linear algebra with respect to independence. In particular, if  $V_1, \dots, V_k$  are such that:

$$V = \sum_{i=1}^k \mu_k V_k,$$

then we cannot conclude that there exists  $\alpha_1, \alpha_2, \dots, \alpha_{i-1}, \alpha_{i+1}, \alpha_k, \alpha$ , such that:

$$V_i = \sum_{\substack{j=1 \\ j \neq i}} \alpha_j V_j \oplus \alpha V,$$

Subspaces in the  $\{\max, +\}$  algebra are more akin to convex polygons, where any inner point  $x$  is a convex combination ( $\sum_{i=1}^n \lambda_i x_i$ ) of vertices  $x_1, \dots, x_n$ , where  $\sum \lambda_i = 1$  and all  $\lambda_i \geq 0$ . When  $x = \sum_{i=1}^n \lambda_i x_i$ , we cannot conclude that  $x_i = \sum_{j \neq i} \alpha_j x_j + \alpha x$ , because the  $\lambda$ 's must be positive. In the  $\{\max, +\}$  algebra, all numbers are "positive" because "max" does not allow inverses. That is, there do not exist numbers "-a" such that  $\max\{a, "-a"\} = \varepsilon$ . The  $k$  basic vectors extracted from  $(M_\lambda)^+$  play the role of the vertices of a convex polygon. The question of independence in the  $\{\max, +\}$  algebra requires theoretical investigations which are beyond the scope of this paper (however, see Cuninghame-Green [16], Gondran and Minoux [24,25], Cohen et al. [10], Moller [29], Oldser and Roos [30]).

The notion of a projector also exists in the algebra. A *projector* is a matrix  $\underline{P}$  such that  $\underline{P}^2 = \underline{P}$ . Given a matrix  $M$ , there is a projector  $P_M$  such that for every  $V$ ,  $VP_M$  is an eigenvector of  $M$  as long as there exists an  $n_0 \in \mathbb{N}$  such that:

$$\forall n \geq n_0, (M_\lambda)^n = (M_\lambda)^{n_0} \tag{20}$$

and  $P_M = \lim_{n \rightarrow +\infty} (M_\lambda)^n$ , a limit which is attained in a finite number of steps. In that case,

$$VP_M M = V \lim_{n \rightarrow +\infty} \lambda (M_\lambda)^{n+1} = \lambda VP_M.$$

#### 4.2. PERIODICITY

More generally, a matrix  $M$  is said to be *order-d-periodical* if there are integers  $d$  and  $n_0$  such that:

$$\forall n \geq n_0, (M_\lambda)^{n+d} = (M_\lambda)^n, \tag{21}$$

thus generalizing eq. (20).

In the following, it is assumed that  $\forall n, M^n$  is irreducible. This condition holds, for example, if  $G(M)$  contains a loop, i.e. there exists an  $i$  such that  $M_{ii} > \varepsilon$ . The following results are proven in Cohen et al. [8]:

- If the critical graph is a circuit of length  $d$ , then  $M$  is order- $d$ -periodical.
- If the critical graph has only one connected component, then  $d$  is the g.c.d. (greatest common divisor) of the lengths of the simple circuits (critical circuits) in  $G_0(M)$ . Then  $d$  is called the *order* of  $G_0(M)$  and  $M$  is order- $d$ -periodical.
- If  $G_0(M)$  has  $K$  connected components  $C_l, l = 1, \dots, K$ , let  $d_l$  be the order of  $C_l$ . Then  $M$  is order- $d$ -periodical with  $d = \text{l.c.m.}$  (least common multiple) of  $(d_1, \dots, d_K)$ .

If  $M$  is order- $d$ -periodical, the following quantities exist:

$$Q_i = \lim_{n \rightarrow +\infty} (M_\lambda)^{dn+i}, \quad i = 0, \dots, d-1.$$

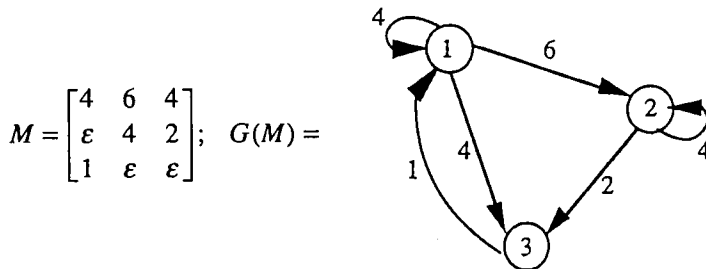
Clearly,  $Q_i = (M_\lambda)^i Q_0 = Q_0 (M_\lambda)^i$ .  $Q_0 = P_{M^d}$  is the projector of  $M^d$ .  $Q_0$  can be obtained from the knowledge of a basis of generators of the eigenspace on the left (extracted from the rows of  $(M_\lambda^d)^+$ , as explained in section 4.1) and a basis of generators of the eigenspace on the right, extracted in the same way from the columns of  $(M_\lambda^d)^+$ . More specifically, let  $V_1, \dots, V_K$  be a basis of left eigenvectors of  $M^d$ , where  $K$  refers to strongly connected component number  $k$  of  $G_0(M^d)$ , and let  $W_1, \dots, W_K$  be a basis of right eigenvectors of  $M^d$ , i.e.  $V_k = (M_\lambda^d)^+_k$  and  $W_k = (M_\lambda^d)^+_{.k}$ . The outer product  $W_k \otimes V_k$  produces an  $m \times m$  matrix, where entry  $i, j$  contains the term:  $(W_k)_i \cdot (V_k)_j$ . It is proven in Cuninghame-Green [16] and shown in Cohen et al. [8] that:

$$Q_0 = \sum_{k=1}^K W_k \otimes V_k.$$

We now demonstrate with an example.

*Example 2*

We have the following matrix  $M$  and its associated graph  $G(M)$ . The weights of each arc are labeled accordingly.





By inspection, it is clear that the circuits (and their weights and lengths) are:

1-1	weight 4	length: 1
2-2	weight 4	length: 1
1-3-1	weight 5	length: 2
1-2-3-1	weight 9	length: 3

This implies that the eigenvalue  $\lambda = 4$ . Hence, the critical graph consists of two components  $C_1$  and  $C_2$  corresponding to loop 1-1 and loop 2-2. We have  $M_\lambda$ :

$$M_4 = \begin{bmatrix} 0 & 2 & 0 \\ \varepsilon & 0 & -2 \\ -3 & \varepsilon & \varepsilon \end{bmatrix}, \quad (M_4)^2 = \begin{bmatrix} 0 & 2 & 0 \\ -5 & 0 & -2 \\ -3 & -1 & -3 \end{bmatrix} = (M_4)^3 = (M_4)^4.$$

Hence,  $(M_4)^+ = (M_4)^2$ . The eigenspace on the left of  $M$  is generated by the two row vectors  $(0, 2, 0)$  and  $(-5, 0, -2)$  (rows 1 and 2 of  $(M_4)^+$ ). So, the eigenspace is:

$$E = \{(x, y, z) \mid \text{there exists } \lambda_1 \text{ and } \lambda_2, \text{ where } x = \max\{\lambda_1, \lambda_2 - 5\}, \\ y = \max\{\lambda_1 + 2, \lambda_2\}, z = \max\{\lambda_1, \lambda_2 - 2\}\}.$$

Notice that the third line of  $(M_4)^+$  is collinear to the first row.  $C_1$  and  $C_2$  are of order 1. Therefore,  $M$  is order-1-periodical.  $Q_0$  is obtained from rows 1 and 2, and columns 1 and 2 of  $(M_4)^+$ :

$$Q_0 = \begin{pmatrix} 0 \\ -5 \\ -3 \end{pmatrix} (0 \ 2 \ 0) \oplus \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix} (-5 \ 0 \ -2) = \begin{pmatrix} 0 & 2 & 0 \\ -5 & 0 & -2 \\ -3 & -1 & -3 \end{pmatrix} (= (M_4)^+ \text{ here}).$$

If  $(x, y, z)$  is any vector, then

$$(x, y, z)Q_0 = (\max\{x, y - 5, z - 3\}, \max\{x + 2, y, z - 1\}, \max\{x, y - 2, z - 3\}) \\ \text{which belongs to } E(\lambda_1 = \max\{x, z - 3\}, \lambda_2 = y).$$

It is indeed an eigenvector.

### 4.3. SYSTEMS OF LINEAR EQUATIONS

The algebraic setting presented in section 3.1 and the concept of order- $d$ -periodicity enable  $\{\max, +\}$  recurrence equations to be solved. A basic result is now stated, from Cohen et al. [8].

PROPOSITION 4

Let  $M$  be an irreducible order- $d$ -periodical matrix with eigenvalue  $\lambda$ . Let  $\{X(n)\}_{n \geq 0}$ ,  $\{Q(n)\}_{n \leq N}$  be sequences of vectors which are solutions of the recurrent systems:

$$X(n) = X(n - 1)M, \quad X(0) = X \quad (\text{primal system}),$$

$$Q(n) = MQ(n + 1), \quad Q(N) = -Q \quad (\text{dual system}).$$

Then there exists  $n_0$  such that

$$X(n + d) = \lambda^d X(n) \quad \forall n \geq n_0, \tag{22}$$

$$Q(n) = \lambda^d Q(n + d) \quad \forall n \leq N - n_0. \tag{23}$$

The duality between  $X(n)$  and  $Q(n)$  is acknowledged in the following identity:

$$X(n)Q(n) = X(n')Q(n'), \quad \text{for } 0 \leq n \leq N \quad \text{and} \quad 0 \leq n' \leq N. \quad \square$$

The behavior of  $X(n)$  over the long term is more precisely described as follows:

*Case (i):* If the critical graph  $G_0(M)$  has only one connected component of order 1 (e.g.  $G_0(M)$  is a loop), then

$$\forall n \geq n_0, \quad X(n) = \lambda^n X Q_0.$$

$Q_0$  is a rank 1 projector. If  $X$  and  $X'$  are two initial states, then  $XQ_0$  and  $X'Q_0$  are collinear. In other words,  $\forall X$ , the sequence  $S(n) = X(n)\lambda^{-n}$  converges to  $XQ_0$ . The steady state is unique.

*Case (ii):* If the critical graph  $G_0(M)$  has only one connected component of order  $d$  (e.g.  $G_0(M)$  is a circuit of length  $d$ ), then

$$\forall n \geq n_0, \quad X(n) = \lambda^n X Q_i, \quad \text{if } n - d(n : d)^* = i, \quad i = 0, \dots, d - 1.$$

$Q_0$  is a rank  $d$  projector.

*Case (iii):* If the critical graph  $G_0(M)$  has  $k$  connected components of order  $d_k$ ,  $k = 1, \dots, K$ , then

\*The notation  $n:d$  is the Euclidean division and  $n - d(n:d)$  is the remainder of this division.

$$\forall n \geq n_0, X(n) = \lambda^n X Q_i, \text{ if } n - d(n : d) = i, \quad i = 0, \dots, d - 1,$$

with  $d = \text{l.c.m.}(d_1, \dots, d_k)$  and the rank  $Q_0 = \sum_{i=1}^k d_i$ .

These results are very useful for understanding the behavior of discrete-event systems (that can be described by timed marked graphs) in the long run, from the knowledge of their initial state. Such systems achieve a complex periodical behavior: the earliest and latest firing times of events repeated  $n$  and  $n + d$  times are the same up to a translation of amount  $\lambda^d$ ,  $\forall n > n_0$ . A limiting cycle of states of the system is eventually reached. This limiting cycle is unique in case (i), and it becomes stable. The period of the system is closely related to the eigenvalue of its associated matrix. The effective calculation of the time characteristics of the limiting cycle is easily obtained from the knowledge of the eigenvectors of the matrix. More specifically, the earliest starting times in steady state are obtained as  $\{X^i = X Q_i \mid i = 0, \dots, d - 1\}$ , up to a translation. It is called the *earliest steady state*.

To get the latest starting times in steady state, we must first recall a result obtained by Cuninghame-Green [15] on solving  $\{\max, +\}$  systems of equations of the form

$$XM = V, \tag{24}$$

where  $V$  is some given vector of dates. The problem is to determine how to choose starting times contained in  $X$  to ensure that the next cycle is undertaken at preassigned times. Equation (24) can be relaxed to

$$XM \lesssim V, \tag{25}$$

where  $\lesssim$  is the usual vector-ordering:  $V \lesssim V', \Leftrightarrow V_i \leq V'_i, \forall i$ . Equation (25) views  $V$  as a due date constraint.

Now, the set of latest starting times is described by  $\hat{X}$  satisfying

$$\hat{X} = \sup\{X \mid XM \lesssim V\},$$

where  $\sup$  is in the sense of  $\lesssim$  (vector-maximization).  $\hat{X}$  can be characterized as follows.

PROPOSITION 5

$\hat{X}$  exists and is defined by

$${}^t\hat{X} = (M \cdot {}^t(V^{-1}))^{-1},$$

where  ${}^tV$  is the transposition of  $V$ ,  $(V^{-1})_i = -V_i$ , and  $(M^{-1})_{ij} = -M_{ij}$ . Moreover, if eq. (24) has a solution, then  $\hat{X}$  is this solution.  $\square$

Now, the latest steady state of the system can be obtained by considering the earliest steady state as a due date constraint, i.e. by defining the sequence of vectors:

$$\begin{aligned} \hat{Y}(N-n), \quad n \geq 0, \quad \text{such that } \hat{Y}(N) &= X Q_0 \quad \text{and } \forall n \leq N, \\ \hat{Y}(N-n) &= \sup\{Y(N-n) \mid Y(N-n)(M_\lambda)^n \leq X Q_0\} \\ &= {}^t((M_\lambda)^n \cdot {}^t(X Q_0)^{-1})^{-1}, \quad \text{using theorem 4} \\ &= {}^t(Q_i {}^t(X Q_0)^{-1})^{-1}, \quad \text{for } n \geq n_0 \quad \text{and } n - d(n:d) = i. \end{aligned} \quad (26)$$

Hence, the latest steady state is defined by the row vectors

$$\bar{Y}^i = {}^t(Q_{d-i} {}^t(X Q_0)^{-1})^{-1}, \quad i = 1, \dots, d.$$

The following property can be established:

PROPOSITION 6

For every  $i = 1, \dots, d$ ,  $\bar{Y}^i \geq \underline{X}^{d-i}$ .

*Proof*

Note that  $\forall n \geq n_0$ , denoting  $p = n:d$  and  $i = n - dp$ ,

$$\underline{X}^{d-i} \in \{Y(N-n) \mid Y(N-n)(M_\lambda)^n \leq \underline{X}^0 \triangleq X Q_0\}.$$

Indeed,  $\underline{X}^{d-i}(M_\lambda)^{dp+i} = X Q_0 (M_\lambda)^{d-i} (M_\lambda)^i Q_0 = X Q_0$ , since  $Q_0^2 = Q_0 = Q_0 (M_\lambda)^d$ . However,  $\underline{X}^{d-i}$  is not the greatest element of the above defined set. (This maximizing element is  $\bar{Y}^i$ .) □

Proposition 6 says that the earliest occurrence times of events take place before the latest occurrence times.

Since each component denotes the firing time of some event, the *slack times* ( $s_j$ ) associated with the occurrence of events are defined by:

$$\forall j, \quad s_j = (\bar{Y}^i)_j - (\underline{X}^{d-i})_j. \quad (27)$$

If  $j$  belongs to the critical graph of  $M$ , then  $s_j = 0$  can be established.

Example 3

Consider the matrix  $M$  in example 2, with the initial state  $X = (0, 0, 0)$ .  $M$  is order-1-periodical. The steady state is:  $4^n Q_0 = 4^n(0, 2, 0) = (4n, 4n + 2, 4n)$ .  $\underline{X}^0 = (0, 2, 0)$ ,  $X(1) = XM = (4, 6, 4)$ , and  $X(2) = (8, 10, 8)$ . The steady state is reached ( $n_0 = 1$ ):

$$\begin{aligned}
 {}^tY^{-1} &= (Q_0 {}^t(\underline{X}^0)^{-1})^{-1} = \left( \begin{bmatrix} 0 & 2 & 0 \\ -5 & 0 & -2 \\ -3 & 1 & -3 \end{bmatrix} \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix} \right)^{-1} \\
 &= \begin{pmatrix} 0 \\ -2 \\ -3 \end{pmatrix}^{-1} = \begin{pmatrix} 0 \\ 2 \\ 3 \end{pmatrix}.
 \end{aligned}$$

Thus, the slack values are  $s_1 = 0, s_2 = 0$  (nodes 1 and 2 are critical nodes), and  $s_3 = 3$  (node 3 is not critical). It can be checked that  $(0, 2, 3)M = (4, 6, 4)$ , but that  $(0, 2, 3 + \epsilon)M = (4 + \epsilon, 6, 4)$ . Then 3 is a latest occurrence date for the event represented by vertex 3, if vertices 1 and 2 fire at  $t = 0$  and 2, respectively.

4.4. REDUCIBLE MATRICES

In this section, we relax the assumption that the graph  $G(M)$  associated with matrix  $M$  is strongly connected. In this case, it is possible to compute strongly connected components  $\{G_1, \dots, G_p\}$ , which are partially ordered by the relation  $\geq$ :  $G_i \geq G_j \Leftrightarrow$  there is a path from  $G_i$  to  $G_j$ . Then,  $G_i \geq G_j$  means that  $G_i$  is upstream from  $G_j$ . In terms of marked graphs,  $G_i$  sends tokens to  $G_j$ , but not conversely. Each graph  $G_i$  corresponds to a matrix  $M(i)$ , which is irreducible and has its own eigenvalue  $\lambda_i$ , unless  $G_i$  is reduced to an isolated node with no loop. In the latter case,  $G_i$  is said to be *degenerate*.

Let us assume for the moment that  $G(M)$  has only two strongly connected components  $G_1$  and  $G_2$  which are not degenerate.  $M$  can be arranged as an upper block-triangular matrix with the following structure:

$$M = \begin{bmatrix} M(1) & M(12) \\ \theta & M(2) \end{bmatrix}, \quad M(12) \neq \theta.$$

It is assumed that  $G_1 \geq G_2$ .

PROPOSITION 7

$M$  has a left eigenvalue equal to  $\lambda_2$ . Moreover, if  $\lambda_1 > \lambda_2$ , then  $\lambda_1$  is also a left eigenvalue.  $M$  has no other eigenvalue.

*Proof*

Let  $X^i$  be a left eigenvector of  $M(i)$ ,  $i = 1$  or  $2$ . Now,

$$[\varepsilon, \varepsilon, \dots, \varepsilon X^2]M = [\varepsilon, \varepsilon, \varepsilon, \dots, \varepsilon X^2 M(2)] = \lambda_2 [\varepsilon, \dots, \varepsilon X^2].$$

Hence,  $\lambda_2$  is an eigenvalue of  $M$ .

Let  $X$  be the concatenation of vectors  $X^1$  and  $X^2$  and denoted by:  $X = [X^1 X^2]$ .

Then,

$$[X^1 X^2]M = [X^1 M(1), X^1 M(12) \oplus X^2 M(2)] = \lambda_1 X$$

if and only if:  $X^1 M(12) \oplus X^2 M(2) = \lambda_1 X^2$ . That is,

$$X^1 M(12)_{\lambda_1} \oplus X^2 M(2)_{\lambda_1} = X^2,$$

where  $M_\lambda = \lambda^{-1}M$ . That is,

$$X^2 = X^1 M(12)_{\lambda_1} (M(2)_{\lambda_1})^*.$$

However,  $(M(2)_{\lambda_1})^*$  exists only if  $M(2)_{\lambda_2}$  has only negative weight circuits, i.e.  $\lambda_1 > \lambda_2$ . That  $M$  has no other eigenvalue is obvious from the unicity of the eigenvalues of  $M(1)$  and  $M(2)$ .  $\square$

Note that when  $\lambda_1 > \lambda_2$ , the eigenvectors of  $M$  for  $\lambda_2$  are the eigenvectors of  $M(2)$  completed with  $\varepsilon$ 's, which amounts to cancelling both  $M(1)$  and  $M(12)$  in  $M$ . That is, the system whose marked graph is  $G_2$  is run by its own dynamics. The eigenvectors for  $\lambda_1$  are defined by:

$$\{ X = [X^1 X^2] \mid X^1 \text{ is an eigenvector of } M^1 \text{ and } X^2 = X^1 M(12)_{\lambda_1} (M(2)_{\lambda_1})^* \},$$

as shown in the proof of theorem 4.

When  $\lambda_2 > \lambda_1$ , the system  $G_1$  cycles faster than  $G_2$  and sends tokens to  $G_2$  (through  $G(M(12)) \triangleq G_{12}$ ) at a rate which  $G_2$  cannot follow. Hence, as time elapses, more and more tokens wait to be cancelled by firing the interface vertices between  $G_2$  and  $G_{12}$ . Tokens accumulate in the system and stability is not preserved.

Hence, for a proper behavior of systems whose structure is not strongly connected, it is useful to assume that if  $G_i$  is upstream of  $G_j$ , then  $G_i$  is slower ( $\lambda_i > \lambda_j$ ) than  $G_j$ .

Now the process defined by  $X(n) = X(n - 1)M$  and  $X(0) = X$  can be studied as follows:

$$X(n) = X \begin{bmatrix} M(1)^n & M(1, 2, n) \\ \theta & M(2)^n \end{bmatrix},$$

where  $M(1, 2, n) = M(1)^{n-1}M(1, 2) \oplus M(1, 2, n - 1)M(2)$  and  $M(1, 2, 1) = M(1, 2)$ . That is, if  $X^1(n)$  denotes the part of  $X(n)$  pertaining to  $G_1$ , then  $X^1(n) = X^1M(1)^n$ . In the long run, if  $Q_0(1)$  denotes the projector built on  $M(1)$ , we have  $X^1(n) = X^1Q_i(1)\lambda_1^n$ , for  $n - (n : d_1) = i$ , if  $d_1$  is the order of periodicity of  $M(1)$ .

Now,

$$X^2(n) = X^1(n - 1)M(1, 2) \oplus X^2(n - 1)M(2). \tag{28}$$

For a large enough  $n$ , we can compute a periodical behavior of  $X^2(n)$  by noticing that  $\lambda_1$  is the only interesting eigenvalue of  $M$  when  $G_1$  has been excited as well as  $G_2$ . This means that there exist  $n_0$  and  $n > n_0$  such that

$$X^2(n) = X^1Q_i(1)\lambda_1^n \oplus X^2(n)\lambda_1^{-1}M(2),$$

and we obtain

$$X^2(n) = X^1Q_i(1)\lambda_1^n (M(2)\lambda_1)^*, \text{ for } n - (n : d_1) = i.$$

$X^2$ , the initial state of  $G_2$ , does not appear in steady state. To see this, let  $Q_0(2)$  be the projector associated with  $M(2)$ , and let  $d_2$  be the order of periodicity for  $M(2)$ . Then there exists  $n'_0$  such that  $X^2M(2)^n = X^2Q_j(2) + n\lambda_2$ , where  $n - d_2(n : d_2) = j$ . Now let us compare the  $k$ th component of  $X^2(M(2))^n$  and that of  $X^2(n)$  as given by eq. (28):

$$n \geq n'_0 : (X^2M(2)^n)_k = (X^2Q_j(2))_k + n\lambda_2 \leq \max_{j=0, \dots, d_2-1} \{X^2Q_j(2)\}_k + n\lambda_2 \triangleq a_n.$$

$$n \geq n_0 : (X^2(n))_k \geq (X^1Q_i(1))_k + n\lambda_1 \geq \min_{i=0, \dots, d_1-1} \{X^1Q_i(1)\}_k + n\lambda_1 \triangleq b_n.$$

Now, since  $\lambda_1 > \lambda_2$ , there exists  $n''_0$  such that  $n > n''_0 \Rightarrow b_n > a_n$ , i.e.  $(X^2(n))_k > (X^2M(2)^n)_k$ . Hence, the term  $X^2M(2)^n$  vanishes from  $X^2(n)$ , and does not affect its value in the long run.

### 5. Algorithmic issues

In order to be able to use the results of sections 3 and 4 on practical problems, computational tools are needed. Such tools already exist in the literature, in graph theory (see Dantzig et al. [17], Lawler [28], and Karp [26]). These are applied to the computation of eigenvalues of matrices in the  $\{\max, +\}$  algebra, as suggested by

Chretienne [7] in the context of timed Petri net models. This section recalls a graph-theoretical interpretation of the eigenvalues. An algorithm is provided to compute the eigenvalue and to determine critical circuits of activities. We prove that adding tokens to a marked graph representation decreases the eigenvalue.

$M$  is a square matrix with coefficients that are real numbers or  $\varepsilon = -\infty$ . Let  $G(M)$  be its associated graph whose nodes are rows (or columns) of the matrix. Arcs correspond to the entries that are different from  $\varepsilon$ . Each arc  $(i, j)$  is labeled by  $M_{ij}$ . Let  $\lambda$  be the unique eigenvalue of  $M$ ; this unicity holds when  $G(M)$  is strongly connected, which is assumed here. The eigenvalue is defined on the  $\{\max, +\}$  algebra.

5.1. COMPUTING EIGENVALUES AND CRITICAL CIRCUITS

Recall that  $\lambda$  is the average weight of any circuit in  $G(M)$  with maximal average weight. Finding this maximal average weight is a problem that has been addressed first by Dantzig et al. [17], and Lawler [28]. Karp [26] provides a very efficient algorithm to calculate  $\lambda$  without requiring its algebraic interpretation. He proves that the eigenvalue of  $M$  can be obtained by choosing any vertex  $i$  of  $G(M)$  and calculating:

$$\lambda = \max_{j=1, \dots, m} \min_{0 \leq k \leq m-1} \left\{ \frac{(M^m)_{ij} - (M^k)_{ij}}{m - k} \right\}, \tag{29}$$

where  $(M^k)_{ij}$  is the  $(i, j)$ th entry of the  $k$ th  $\{\max, +\}$  power of  $M$ , i.e. the maximal weight of paths of length  $k$  from  $i$  to  $j$  in  $G(M)$ .

The implementation of eq. (29) requires only the determination and storage of row  $i$  of matrix  $M^k$ , for  $k = 1, \dots, m$ . The computational time is then  $O(m \times \text{the number of arcs in } G(M))$ . Karp's [26] proof indicates that for any critical circuit, there is a node  $j^*$  in this circuit such that

$$\lambda = \min_{0 \leq k \leq m-1} \left\{ \frac{(M^m)_{ij^*} - (M^k)_{ij^*}}{m - k} \right\}$$

for any choice of  $i$ . Moreover, a critical circuit is then trapped in one maximal weighted path in  $m$  arcs from  $i$  to  $j^*$ . If this critical circuit has length  $d$ , then letting  $k^* = m - d$ ,

$$\lambda = \frac{(M^m)_{ij^*} - (M^{k^*})_{ij^*}}{m - k^*}.$$

A critical circuit can be retrieved by a systematic check of the maximal weighted path from  $i$  to  $j^*$  in  $m$  arcs. While computing the terms  $(M^k)_{ij}, \forall j$ , the extremal paths can be stored, and critical circuits can thus be trapped.

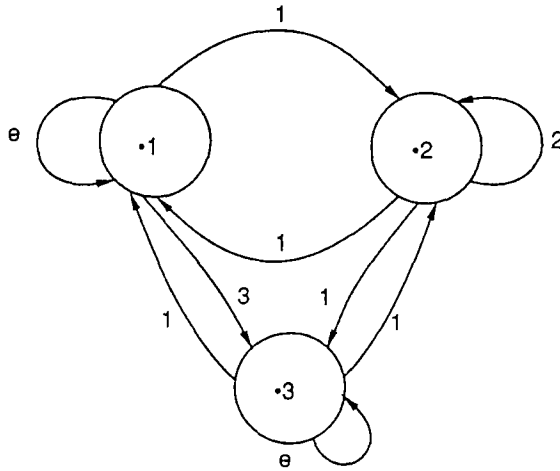


Example 4

Suppose that

$$M = \begin{bmatrix} e & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 1 & e \end{bmatrix}.$$

(Recall that the identity  $e = 0$ ). The associated graph  $G(M)$  is the following:



Choosing  $i = 1$ , the following array  $A$  is built.

	k	0	1	2	3
vertices j					
A =	1	e	e	4 (3)	5 (2)
	2	e	1	4 (3)	6 (2)
	3	e	3	3 (1,3)	7 (1)

Entry  $(j, k)$  contains  $(M^k)_{ij}$ , for  $k > 2$ . Entry  $(j, k)$  also contains a list  $(i_1, \dots, i_q)$  of vertices which are the predecessors of  $j$  on each maximal weighted path from 1 to  $j$  in  $k$  arcs. Note that  $M^0 = E$ . Column  $k$  of the array is obtained as

$$\forall j, A_{jk} = \max_{l=1, \dots, 3} \{A_{l, k-1} + M_{lj}\},$$

which says that  $(M^k)_j = (M^{k-1})_j M$ . Now array  $B$ , containing as entry  $(j, k)$  the term

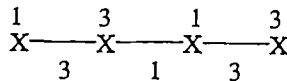
$$\frac{(M^3)_{ij} - (M^k)_{ij}}{3 - k}, \quad k = 0, 1, \text{ and } 2,$$

is obtained as:

B =

	k	0	1	2	
j					
1		5/3	5/2	1	1
2		$+\infty$	5/2	2	2
3		$+\infty$	2	4	2

The circled entries are the minimum in their corresponding row. These values are placed on the right-hand side of the array. The maximum of these minimum values is placed in a square. It is  $2 (= \lambda)$ . There are two rows that provide this  $\lambda$ :  $j = 2$  and  $3$ . For  $j = 3$  and  $k = 1$ , we can expect to capture a critical circuit  $\lambda$  of length  $3 - 1 = 2$  in the maximal weighted path from 1 to 3 in 3 arcs. From array  $A$  and the knowledge of the predecessors, we obtain the path  $(1, 3, 1, 3)$  containing the circuit  $\gamma = (1, 3, 1)$ :



Hence,  $\gamma = (1, 3, 1)$  is critical. Indeed, its weight is 4 and its length is 2. For  $j = 2$  and  $k = 2$ , we capture a critical circuit of length  $3 - 2 = 1$  in the maximal

weighted path from 1 to 2 retrieved from array  $A$ . This critical circuit is the critical loop (2, 2) contained in the path (1, 3, 2, 2).

5.2. THE INFLUENCE OF THE NUMBER OF TOKENS ON  $\lambda$

If we again interpret matrix  $M$  as derived from the analysis of a timed marked graph in the token space, each row of  $M$  corresponds to a token. Then a path  $i_1, i_2, \dots, i_p$  in  $G(M)$  corresponds to one or more paths in the marked graph  $G$ , each path containing  $p$  tokens. In particular, a circuit of length  $p$  in  $G(M)$  is interpreted as a circuit in  $G$  containing  $p$  tokens. Hence, when  $G$  is strongly connected, if  $\tau(\gamma)$  is the number of tokens on circuit  $\gamma \in G$ , we have:

$$\lambda = \max_{\gamma \in G(M)} \frac{w(\gamma)}{l(\gamma)} = \max_{\gamma \in G} \frac{w(\gamma)}{\tau(\gamma)}. \tag{30}$$

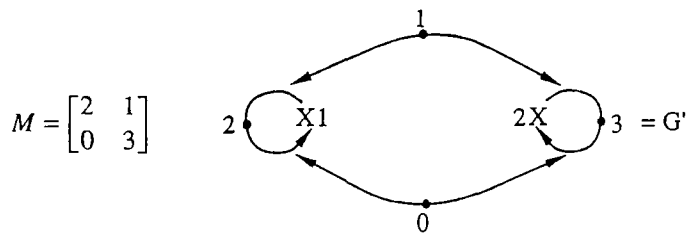
This result has been proved by Ramchandani [33] (see also Ramamoorthy and Ho [32]). However, Ramchandani provided no computationally efficient method for computing  $\lambda$  or for finding critical circuits. Moreover, Ramamoorthy and Ho only provide a polynomial algorithm to determine, given a guessed cycle time  $a$ , whether or not  $a \geq \lambda$ .

Another interesting point is that any matrix  $M$  with nonnegative (and finite) entries can be associated with a marked graph  $G'$ , as follows:

- Associate with each row  $i$ , a vertex and an arc  $(i, j)$ ,  $\forall M_{ij} \geq 0$ .
- If  $M_{ij} \geq 0$ , then there is a token on arc  $(i, j)$ .

We thus obtain a marked graph  $G'$ , as demonstrated in example 5.

*Example 5*



It is clear in example 5 that matrices  $F_0, F_2, F_3, \dots$  are  $\emptyset$ , since every arc has one token on it. We have  $X(n) = \tilde{X}(n) = X(n) = X_{n-1} F_1 = X_{n-1} M$ . There are no redundant entries in  $X(n)$  with respect to each other.

As a consequence, the above remark provides a technique to transform any timed marked graph  $G$  into another equivalent graph  $G'$  such that each arc carries

exactly one token. In particular, first construct  $M = B(F_0)^*CK$ . Then,  $G'$  can be constructed from  $M$ .

If  $M$  has any negative finite entries, then if  $w = \min\{M_{ij} \neq \varepsilon\} < 0$ , it is possible to derive an associated marked graph by considering the matrix  $wM$  and modifying the state variable accordingly (see Cohen et al. [8]). However, in practice, starting from a timed marked graph always leads to  $M_{ij} \geq 0, \forall i, j$ .

Now, given a critical circuit  $\gamma_0$  on  $G(M)$ ,  $\gamma_0$  corresponds to a circuit  $\gamma'_0$  of critical events in the original marked graph  $G$ . Since  $\gamma = w(\gamma_0)/\tau(\gamma'_0)$ , it is clear that by adding a token on the critical circuit  $\gamma'_0$ , the average weight of this circuit becomes  $w(\gamma_0)/(\tau(\gamma'_0) + 1)$ . This may make  $\gamma_0$  no longer critical. On the other hand, there is no other way of decreasing  $\lambda$  (i.e. to improve the performance of the marked graph), since adding tokens on other circuits would leave  $\lambda$  unchanged.

To see this in a more rigorous manner, let  $G$  be a timed marked graph with each arc carrying at most one token. We can always satisfy this condition (see section 3.4). Hence, we have the following:

$$X(n) = X(n)F_0 \oplus X(n - 1)F_1 = X(n - 1)F_1(F_0)^*$$

Now, if we add a token to arc  $(\underline{i}, \underline{j})$  in  $G$ , we must consider two cases:

(a) *Case 1:* Arc  $(\underline{i}, \underline{j})$  had contained no token.

Hence,  $(F_0)_{\underline{i}\underline{j}} = t_{\underline{i}\underline{j}}$  and  $(F_1)_{\underline{i}\underline{j}} = \varepsilon$ . Adding a token yields  $F'_0$  and  $F'_1$  such that

$$\forall (i, j) \neq (\underline{i}, \underline{j}), (F'_0)_{ij} = (F_0)_{ij}, (F'_1)_{ij} = (F_1)_{ij}, (F'_0)_{\underline{i}\underline{j}} = \varepsilon, \text{ and } (F'_1)_{\underline{i}\underline{j}} = t_{\underline{i}\underline{j}}.$$

Now, using matrices  $B, C$ , and  $K$  for working in the token space and vector  $Z(n)$ , where each component pertains to a token (see section 3.4),  $B', C'$ , and  $K'$  are obtained as follows:

$$B' = \begin{bmatrix} B \\ b \end{bmatrix}, \quad \text{where } b \text{ is an additional row such that } b_{\underline{j}} = 0, b_k = \varepsilon, \forall k = \underline{j},$$

$$C' = \begin{bmatrix} C & c \end{bmatrix}, \quad \text{where } c \text{ is an additional column such that } c_{\underline{i}} = 0, c_k = \varepsilon, \forall k = \underline{i},$$

and

$$K' = \begin{bmatrix} K & \varepsilon \\ \varepsilon & t_{\underline{i}\underline{j}} \end{bmatrix}, \quad \text{which has one more row and column than } K.$$

Now we consider  $Z'(n) = [Z(n), z(n)]$ , where  $z(n)$  is the entry pertaining to the new token.

In the token space matrix,

$$\begin{aligned}
 M' &= B'(F'_0)^* C'K' \\
 &= \begin{bmatrix} B \\ b \end{bmatrix} (F'_0)^* \begin{bmatrix} C & c \end{bmatrix} \begin{bmatrix} K & \varepsilon \\ \varepsilon & t_{ij} \end{bmatrix} \\
 &= \begin{bmatrix} B(F'_0)^* \\ (F'_0)^*_j \end{bmatrix} \cdot \begin{bmatrix} CK & \begin{bmatrix} \varepsilon \\ t_{ij} \\ \varepsilon \end{bmatrix} \end{bmatrix} \text{ --- } i\text{th row} \\
 &= \left[ \begin{array}{c|c} B(F'_0)^* CK & t_{ij} B(F'_0)^*_{.i} \\ \hline (F'_0)^*_i CK & t_{ij} (F'_0)^*_{j.i} \end{array} \right] = M'.
 \end{aligned}$$

Since  $G(F_0)$  has no circuit and, moreover,  $(F_0)_{ij} > 0$ , then  $(F_0)^*_{ji} = \varepsilon$ . Deleting arc  $(i, j)$  from  $G(F_0)$  leads to  $(F'_0)^*_{ji} = \varepsilon$  again. Hence,  $M'_{p+1, p+1} = \varepsilon$ . Now, assume that arc  $\underline{i}, \underline{j}$  belongs to a critical circuit in  $G$ , and let  $k$  and  $l$  be two tokens such that

- $l$  is on an input arc of vertex  $i_l$ .
- $k$  is on an input arc of vertex  $i_k$ .
- Arc  $\underline{i}, \underline{j}$  is on a maximal weight path from  $i_l$  to  $i_k$ .
- There is no token on this path, except  $k$ .

Now

$$\begin{aligned}
 ((F'_0)^*_j CK)_k &= \text{maximal weight of a path from } \underline{j} \text{ to } k \text{ in } G \\
 &= ((F_0)^*_j CK)_k = M_{p+1, k},
 \end{aligned}$$

since all paths that originated on  $\underline{j}$  in  $G(F_0)$  remain the same in  $G(F'_0)$ . (Recall that  $G(F_0)$  has no circuits.)

Similarly,  $(t_{ij} B(F'_0)^*_{.i})_l = (t_{ij} B(F_0)^*_{.i})_l = M'_{l, p+1}$ , since no path that ends in  $\underline{i}$  changes when  $\bar{G}(F_0)$  turns into  $\bar{G}(F'_0)$ .

Moreover, arc  $(\underline{i}, \underline{j})$  is on a maximal-weighted path from  $i_l$  to  $i_k$  in  $G(F'_0)$  and so  $(F'_0)^*_{i_l i} + t_{ij} + (F'_0)^*_{j i_k} = (F_0)^*_{i_l i_k}$ . We can conclude (multiplying by  $K$  on the right) that  $M_{lk} = M'_{l, p+1} + M'_{p+1, k}$ , where "+" has its usual meaning.

So, adding a token to an arc  $\underline{i}, \underline{j}$  that belongs to a critical circuit is the same as breaking a critical arc  $l, k$  in  $G(\bar{M})$  (the graph whose nodes are tokens) by adding a node  $p + 1$  (without a loop), while preserving the weight from  $l$  to  $k$ , as pictured below:



Now  $B'$  has  $p + 1$  rows and  $q$  columns. Let  $s$  be the number of the token on  $(i, j)$  in  $G$ . Then the old token is still in the input arc of vertex  $\underline{j}$ . The new token  $\bar{p} + 1$  is at the input of  $q$ . Hence:

$$B' = \begin{array}{c|c|c|c} & & \underline{j} & q \\ \hline & B & & \varepsilon \\ \hline s & & 0 & \\ \hline & & & \\ \hline p+1 & \varepsilon & & 0 \end{array}$$

That is,  $(B')_{ij} = B_{ij}$ , for  $i \leq p$  and  $j \neq q$ ;  $(B')_{p+1,j} = (B')_{i,q} = \varepsilon$ , but  $(B')_{p+1,q} = 0$ .

$$C' = \begin{array}{c|c|c|c} & s & p+1 & \\ \hline C & & C & \varepsilon \\ \hline \underline{i} & & \varepsilon & 0 \\ \hline C & & C & \varepsilon \\ \hline q & \varepsilon & 0 & \varepsilon \end{array}$$

That is,  $(C')_{ij} = C_{ij}$ , for  $i \neq q$  and  $j < p$ . However,  $C'_{is} = \varepsilon$  (instead of 0).  $C'_{qi} = C'_{j,p+1} = \varepsilon$ , except  $C'_{ip+1} = 0$  and  $C'_{qs} = 0$ .

$$K' = \begin{array}{c|c|c} & s & p+1 \\ \hline & K & \\ \hline & & 0 \\ \hline s & & \\ \hline p+1 & & t_{ij} \end{array}$$

That is,  $(K')_{ii} = K_{ii}$ , except that  $K'_{ss} = 0$ .  $K'_{p+1,p+1} = t_{ij}$ .

Now,

$$(F_0^*) = \left[ \begin{array}{c|c} F_0^* & \varepsilon \\ \hline \varepsilon & 0 \end{array} \right]$$

and

$$B'F_0^*C' = \left[ \begin{array}{c|c} B & \varepsilon \\ \hline \varepsilon & 0 \end{array} \right] \left[ \begin{array}{c|c} F_0^* & \varepsilon \\ \hline \varepsilon & 0 \end{array} \right] C'$$

$$= \left[ \begin{array}{c|c|c|c} & s & & p+1 \\ BF_0^*C & \varepsilon & BF_0^*C & (BF_0^*)_{.i} \\ & \varepsilon & & \\ & \varepsilon & & \\ & \varepsilon & & \\ \hline \varepsilon\varepsilon\varepsilon & 0 & \varepsilon\varepsilon\varepsilon & \varepsilon \end{array} \right]$$

Hence,

$$M' = \left[ \begin{array}{c|c|c|c} BF_0^*CK & \varepsilon & BF_0^*CK & t_{ij}(BF_0^*)_{.i} \\ & \varepsilon & & \\ & \varepsilon & & \\ \hline \varepsilon\varepsilon\varepsilon & 0 & \varepsilon\varepsilon & \varepsilon \end{array} \right]$$

It is easy to check that the last column of  $M'$ , i.e.  $t_{ij}(BF_0^*)_{.i}$ , is the same as column  $s$  of  $M$ . Indeed

$$M_{.s} = (B(F_0)^*CK)_{.s} = B(F_0)^*C(K)_{.s} = B(F_0)^* \begin{pmatrix} \varepsilon \\ \varepsilon \\ t_{ij} \\ \varepsilon \end{pmatrix},$$

where  $t_{ij}$  is on row  $i$ , since  $C_{is} = 0$ .

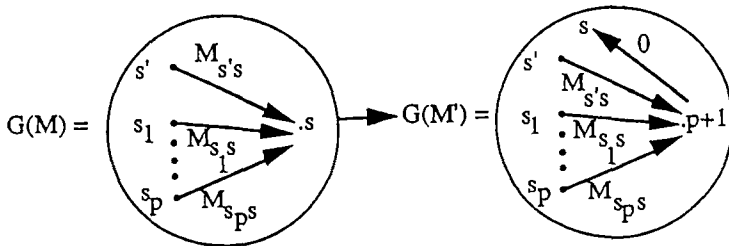
As a result, it is not necessary to recompute everything to get  $M'$ . One only needs to add a token to an arc which already contains one. The column corresponding to the old token is shifted to the new column pertaining to the new token and a zero is put in the bottom of the old token column to make the new token row. That is, letting  $M = \{M(1, s - 1); M_{.s}; M(s + 1, p)\}$ , then

$$M' = \left[ \begin{array}{c|c|c|c} M(1, s - 1) & \varepsilon & M(s + 1, p) & M_{.s} \\ & \varepsilon & & \\ & \varepsilon & & \\ \hline \varepsilon & 0 & \varepsilon & \varepsilon \end{array} \right]$$



$G(M')$  is easily obtained from  $G(M)$  as follows: let  $(s', s)$  be an arc on the critical graph of  $G(M)$ . Then, add a token  $p + 1$  to the arc of  $G$  that contains  $s$ . The results are:

- The  $(s', s)$  arc is deleted (and there is no arc left between  $s'$  and  $s$ ) (since  $M'_{s',s} = \epsilon$ ).
- All arcs in  $G(M)$  that end at  $s$  are deleted. They are input at the new vertex  $p + 1$ .
- A zero-weight arc is added from the new vertex  $p + 1$  to  $s$ .
- The new vertex has no loop on it.



In other words, if arc  $(s', s)$  belongs to a critical circuit, then this circuit remains in  $M'$  with the same weight. However, one more arc  $(p + 1, s)$  of weight zero is added, thus decreasing the average weight. It may then happen that the circuit is no longer critical.

Notice that when a token is added to a previously token-free arc, the computation of  $M'$  is more elaborate. However, we have seen that the same result is produced.

## 6. Summary and future research needs

Previous research on the algebraic representation of timed marked graphs in terms of linear recurrence equations in a  $\{\max, +\}$  algebra has been reviewed and extended. The results extend PERT/CPM concepts to asynchronous and repetitive production processes. Applications of this model to certain classes of flexible manufacturing systems that exhibit periodic behavior are provided in Dubois and Stecke [20].

There are many open research problems involving this approach. Theoretically, determining the minimal representation of a timed marked graph is important. The token space, for instance, does not usually produce a minimal representation. One approach to address this problem might use the concept of minimal disconnecting sets of arcs in directed graphs.

Another important problem is to understand how the initial marking affects the steady-state periodic behavior. It should be possible to define equivalence classes of initial markings, where all markings in a class lead to the same periodic behavior. This research problem is related to the reachability analysis of timed Petri nets. The problem is similar to that of determining ergodicity of Markov matrices, where various initial states in a Markov state transition graph may lead to various asymptotic probability vectors. Also, cyclic behavior can be observed (in the probabilistic sense). Since vanishing states and disjoint sets of strongly connected states can be found in Markovian systems, it may be that the set of live markings in a timed marked graph can be clustered into a subset of vanishing markings and several disjoint cycles of markings, where each cycle corresponds to a specific periodic behavior. A vanishing marking is one that is reachable from no marking in a cycle but *not* conversely. These are conjectures to suggest further research.

Timed marked graphs are a special type of Petri nets. This paper emphasizes the following analogy: marked graphs are to Petri nets what linear systems are to general continuous systems (differential equations) in the field of automatic control. Timed marked graphs are linear in a path algebra and can be analyzed by means of eigenvalues, eigenvectors, and projectors in a way similar to conventional linear systems. The analogy between linear systems theory and timed marked graphs has been examined by Cohen et al. [10,12].

### Acknowledgements

We gratefully acknowledge the expert evaluations and suggestions of the three anonymous referees of this paper and the three anonymous referees of the associated applications paper, as well as the Associate Editors and the Editors, Joe Mazzola and Maurice Queyranne. One of us (K.S.) would also like to acknowledge a summer research grant from the Graduate School of Business Administration of The University of Michigan.

### References

- [1] M. Ajmone Marsan, G. Balbo and G. Conte, A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems, *ACM Trans. Comp. Systems* 2 (2) (1984)93–122.
- [2] M. Ajmone Marsan, G. Balbo and G. Conte, *Performance Models of Multiprocessor Systems* (MIT Press, Cambridge, MA, 1986).
- [3] G. Bel and D. Dubois, A unified setting for various models of automated material flow and production systems, *Comp. Industry* 6(1985)227–235.
- [4] J. Browne, D. Dubois, K. Rathmill, S.P. Sethi and K.E. Stecke, Classification of flexible manufacturing systems, *The FMS Magazine* 2 (2) (1984)114–117.
- [5] J.A. Buzacott and D.D.W. Yao, Flexible manufacturing systems: A review of analytical models, *Manag. Sci.* 32 (7) (1986).
- [6] B.A. Carré, An algebra for network routing problems, *J. Inst. Math. Appl.* 7(1971)273–294.

- [7] P. Chretienne, Les réseaux de Petri temporisés, Thèse d'État, Université de Paris VI, France (1983).
- [8] G. Cohen, D. Dubois, J.P. Quadrat and M. Viot, A linear-system theoretic view of discrete-event processes and its use for performance evaluation in manufacturing, *IEEE Trans. Autom. Control* AC-30 (3) (1985)210–220.
- [9] G. Cohen, P. Moller, J.P. Quadrat and M. Viot, Linear system theory for discrete-event systems, *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, Nevada (1984), pp. 539–544.
- [10] G. Cohen, P. Moller, J.P. Quadrat and M. Viot, Une théorie lineaire des systèmes à événements discrets, Rapport No. 362, INRIA, Rocquencourt, France (1985).
- [11] G. Cohen, P. Moller, J.P. Quadrat and M. Viot, Dating and counting events in discrete event systems, *Proc. 25th IEEE Conf. on Decision and Control*, Athens, Greece (1986).
- [12] G. Cohen, P. Moller, J.P. Quadrat and M. Viot, A 2D-discrete event linear system theory, *Algèbres Exotiques et Systèmes à Evénements Discrets* (INRIA, Rocquencourt, France, 1987), pp. 147–165.
- [13] F. Commonner, A.W. Holt, S. Even and A. Pnueli, Marked directed graphs, *J. Comp. Systems Soc.* 5(1971)511–523.
- [14] R.A. Cuninghame-Green, Describing industrial processes and approximating their steady-state behavior, *Oper. Res. Quart.* 13(1962)95–100.
- [15] R.A. Cuninghame-Green, Projections in a minimax algebra, *Math. Progr.* 10(1976)111–123.
- [16] R.A. Cuninghame-Green, *Minimax Algebra*, Lecture Notes in Economics and Mathematical Systems 166 (Springer, New York, 1979).
- [17] G.B. Dantzig, W.O. Blattner and M.R. Rao, Finding a cycle in a graph with minimum cost-to-time ratio with application to a ship routing problem, in: *Théorie des Graphes, Proc. Int. Symp.*, Rome, Italy (1966) (Dunod, Paris, 1967), pp. 209–214.
- [18] M. Diaz, Modeling and analysis of communication and cooperation protocols using Petri net based models, *Comp. Networks*, 6 (6) (1982)419–441.
- [19] D. Dubois and K.E. Stecke, Using Petri nets to represent production processes, *Proc. 22nd IEEE Conf. on Decision and Control*, San Antonio, TX (1983), pp. 1062–1067.
- [20] D. Dubois and K.E. Stecke, Dynamic analysis of repetitive decision-free discrete-event processes: Applications to production systems, *Ann. Oper. Res.* (1990), this volume.
- [21] R.W. Floyd, Shortest paths, *Comm. Assoc. Comput. Machinery* (1962)345.
- [22] M. Gondran, Les éléments  $p$ -réguliers dans les dioïdes, *Discr. Math.* 25(1979)33–39.
- [23] M. Gondran and M. Minoux, Valeurs propres et vecteurs propres dans les dioïdes et leur interprétation en théorie des graphes, *Bull. Dir. Et. Rech. EDF, Serie C* (1977)24–41.
- [24] M. Gondran and M. Minoux, *Graphes et Algorithmes* (Eyrolles Editions, Paris, 1979).
- [25] M. Gondran and M. Minoux, Linear algebra in dioïds: A survey of recent results, in: *Algebraic and Combinatorial Methods in Operations Research*, ed. R.E. Burkard, R.A. Cuninghame-Green and U. Zimmermann, *Ann. Discr. Math.* 9(1984)147–164.
- [26] R.M. Karp, A characterization of the minimum cycle mean in a digraph, *Discr. Math.* 23(1978) 309–311.
- [27] J. Kuntzmann, *Théorie des Réseaux* (Dunod, Paris, 1972).
- [28] E.L. Lawler, Optimal cycles in doubly weighted directed linear graphs, in: *Théorie des Graphes, Proc. Int. Symp.*, Rome, Italy (1966) (Dunod, Paris, 1967).
- [29] P. Moller, Théorème de Cayley–Hamilton dans les dioïdes et application à l'étude des systèmes à événements discrets, *Proc. 7th Int. Conf. on Systems and Optimization*, Antibes, France (Springer, Berlin, 1986), pp. 215–226.
- [30] G.J. Oldser and C. Roos, Cramer and Cayley–Hamilton in the max-algebra, *Linear Algebra Appl.* 101(1988)87–108.
- [31] J.L. Peterson, *Petri Net Theory and the Modeling of Systems* (Prentice–Hall, Englewood Cliffs, NJ, 1981).

- [32] C.V. Ramamoorthy and G.S. Ho, Performance evaluation of asynchronous concurrent systems using Petri nets, *IEEE Trans. Software Engin.* 6 (5) (1980)440–449.
- [33] C. Ramchandani, Analysis of asynchronous concurrent systems by timed Petri nets, Ph.D. Dissertation, Department of Electrical Engineering, MIT, Cambridge, MA (1974).
- [34] R. Reiter, Scheduling parallel computations, *J. Assoc. Comp. Machinery* 15(1968)590–599.
- [35] T. Smigelski, T. Murata and M. Sowa, A timed Petri net model and simulation of a dataflow computer, *Int. Workshop on Timed Petri Nets*, Torino, Italy (1985), pp. 56–63.
- [36] U. Zimmermann, Linear and combinatorial optimization in ordered algebraic structures, *Ann. Discr. Math.* 10(1981).