

Division of Research
School of Business Administration

March 1988
Revised, October 1989

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

Working Paper #542-b

Didier Dubois
Université Paul Sabatier

Kathryn E. Stecké
The University of Michigan

The University of Michigan
School of Business Administration
Ann Arbor Michigan 48109

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 here. Also, an algorithm to properly characterize the periodic behavior of repetitive production processes is described here. 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 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 [1985] and Buzacott and Yao [1986].

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 Cunninghame-Green [1962, 1979] and also studied by Cohen et al. [1985a, 1985b, 1986]. An algebraic framework for this model exists in the theory of dioids (Gondran [1979] and Gondran and Minoux [1979]). 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 [1981], Chretienne [1983], and Ajmone Marsan [1986]), called *event graphs*. Applications to automated manufacturing have been described in Dubois and Stecke [1983, 1989] and Cohen et al. [1985a].

This paper is a survey of the modeling, algorithmic, and theoretical issues associated with timed marked graphs. A companion paper (Dubois and Stecke [1989]) 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: §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 §2.2. §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 §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 [1971]) in the graph-theoretic literature. Various representation issues are also addressed in §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. [1985a]).

The steady state is characterized by solving the recurrence equations in §4. Some material concerning the periodicity of the system from Cohen et al. [1985a] is reviewed. This allows the Primal Mode to be characterized. Other previous results (Cunningham-Greene [1976]) 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, §5 addresses algorithmic issues in solving $\{\max, +\}$ equations. We show that the computation of eigenvalues and critical circuits consists of shortest path calculations. 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. §5.2 provides tools to quantify the influence of the number of resources (tokens in the marked graph) on the length of the period. §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. [1985a, 1985b, 1986]) in the modeling and analysis of discrete-event systems in the field of manufacturing. It improves and extends earlier attempts by Ramchandani [1974] and Ramamoorthy and Ho [1980] 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 [1981]), 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_α , $\alpha = 1, \dots, p$, is the number of tokens on arc number α) is called a marking. Firing transitions when enabled produces sequences of markings. Marked graphs are useful to describe the behavior of concurrent, asynchronous, deterministic processes, such as computer configurations (Diaz [1982], Ajmone Marsan et

al. [1984], and Smigelski et al. [1985]), parallel computations (Reiter [1968]), and distributed algorithms (Peterson [1981]). 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 \mathring{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 $\forall \alpha \in \mathring{A}$, then $r_{\alpha} \leq 1$; if this condition is relaxed to allow $r_{\alpha} < b_{\alpha}$, where $\{b_{\alpha} | \alpha \in \mathring{A}\}$ is a set of finite upper bounds on the number of tokens on each arc a , 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 $(i_j, i_{j+1}) \in \mathring{A}, \forall j = 1, \dots, n-1$. 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 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 [1981]), 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, θ_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 θ_i after the firing starts; after a duration of θ_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 fixed, 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 θ_i to zero and every t_{ij} to $t_{ij} + \theta_{ij}$. 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, e.g., Ramchandani [1974] and Ramamoorthy and Ho [1980]). Timed Petri nets have been studied in some detail (e.g., Chretienne [1983]).

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 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 Figure 1, crosses are vertices, arrows are arcs, and dots on arrows are tokens.

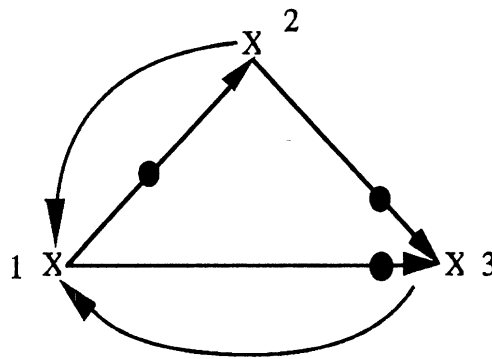


Figure 1. $M_0 = (1, 1, 0, 1, 0)$.

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 Figure 1, the initial marking is: $M_0 = (1,1,0,1,0)$. The set $R(G,M_0)$ is pictured as Figure 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.

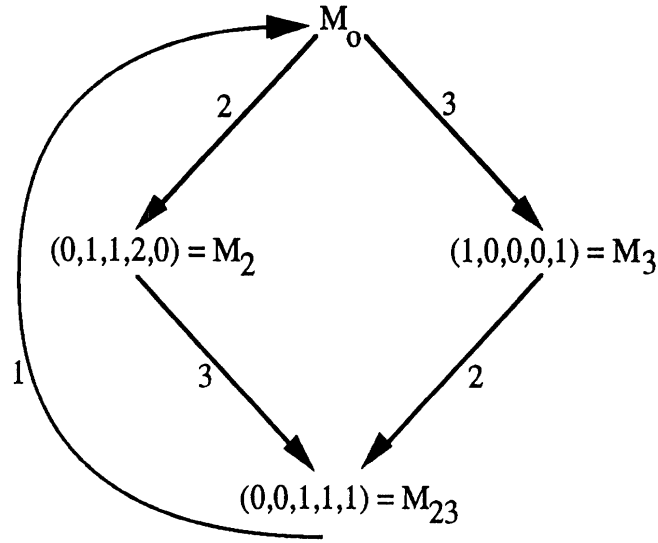


Figure 2. $R(G, M_0)$.

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. ■

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. ■

Note that strong connectedness is required to insure that all vertices in V are eventually enabled. In §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 §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 $\Gamma^-(i) = \{j \mid (j,i) \in \dot{A}\}$ be the set of predecessors of vertex i . Then the following inequality holds:

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

There are r_{ji} tokens on arc (j,i) . But 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 equation (1) for all transitions and noticing that no other constraint impacts the value of $x_i(n)$, we obtain the following result:

Proposition 2:
$$\forall i \in V, x_i(n) = \max_{j \in \Gamma^{-1}(i)} x_j(n-r_{ji}) + t_{ji}. \quad (2)$$

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

$(\mathbb{R}, \oplus, *)$ is an algebraic structure known as a dioid (Gondran [1979]). 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 θ , 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:

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

With these conventions, equation (2) can be written as (Cohen et al. [1986]):

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

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) Δ 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 equation (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 equation (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 ;

$$\bullet \dot{A}(A) = \{(i,j) | A_{ij} > \varepsilon\};$$

$$\bullet A_{ij} = \text{the weight on arc } (i,j).$$

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 [1979]):

Proposition 3: Equation $X = XA \oplus B$ has a solution if and only if no circuit of $G(A)$ has a weight greater than $e = 0$. 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 [1971]). A^* may also be generated by substituting $XA \oplus B$ for X infinitely many times into the right hand side of equation (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 [1962]. More precisely, it is enough to compute $\{A^n | n \leq |V(A)|\}$, where $|V(A)|$ is the number of rows in A .

Now equation (3) can be written as:

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

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 equation (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 $V = \{i, j\} \iff$ 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 equation (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 γ 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 γ 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:

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

Then equation (3) now becomes:

$$X(n) = \sum_{j=0}^{\Delta} X(n-1)F_j \oplus U(n)B, \quad (5)$$

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 $i \in U$), equation (5) becomes (Cohen et al. [1986]):

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

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,

$$X(n) = \sum_{\ell=0}^L X(n-\ell)C_{\ell},$$

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

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 only come 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 equation (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 equation (4) becomes:

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

where

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

The problem with equation (6) is that the vector χ 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 $\hat{A} \supseteq \hat{A}^*$ be the set of marked arcs, $I = \{i \in V \mid (i,j) \in \hat{A}\}$, and $F = \{j \in V \mid (j,i) \in \hat{A}^*\}$. $I(F)$ is called the set of initial vertices (set of final vertices). Note that the matrix associated with $G_0 = (V, \hat{A} - \hat{A}^*)$ is F_0 .

Two cases are now considered:

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

In this case, equation (4) reduces to:

$$X(n) = X(n-1)F_1(F_0)^*. \quad (7)$$

Using G and F as input and output vertices, respectively, associated with sub-vectors $\hat{X}(n)$ and $\tilde{X}(n)$ of $X(n)$, we have:

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

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

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

Now the feedback equation is:

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

where K is a $|F| \times |G|$ 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 equation (7):

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

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

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 that will bear one token each, as shown in Figure 3.

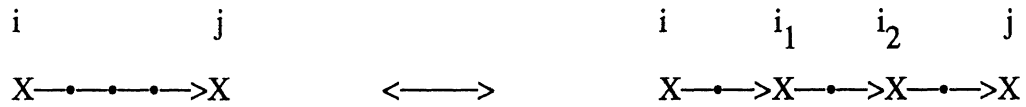


Figure 3. Arc (i,j) of graph G containing 3 tokens.

In 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 Figure 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:

$$x_j(n) = \max \{x_{i_1}(n-r_{ij}) + t_{ij}, a\},$$

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

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

$$\begin{aligned} x_{i_k}^{(n)} &= x_{i_{k-1}i_k}^{(n-1)} + t_{i_{k-1}i_k} \\ &\vdots \\ &\vdots \\ &\vdots \end{aligned}$$

$$x_{i_1}^{(n)} = x_{ii_1}^{(n-1)} + t_{ii_1},$$

where $t_{i_kj} = t_{i_{k-1}i_k} = \dots = 0$ and $t_{ii_1} = t_{ij}$.

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 equation (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 θ 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 equation (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 equation (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 Figure 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 equation (2).

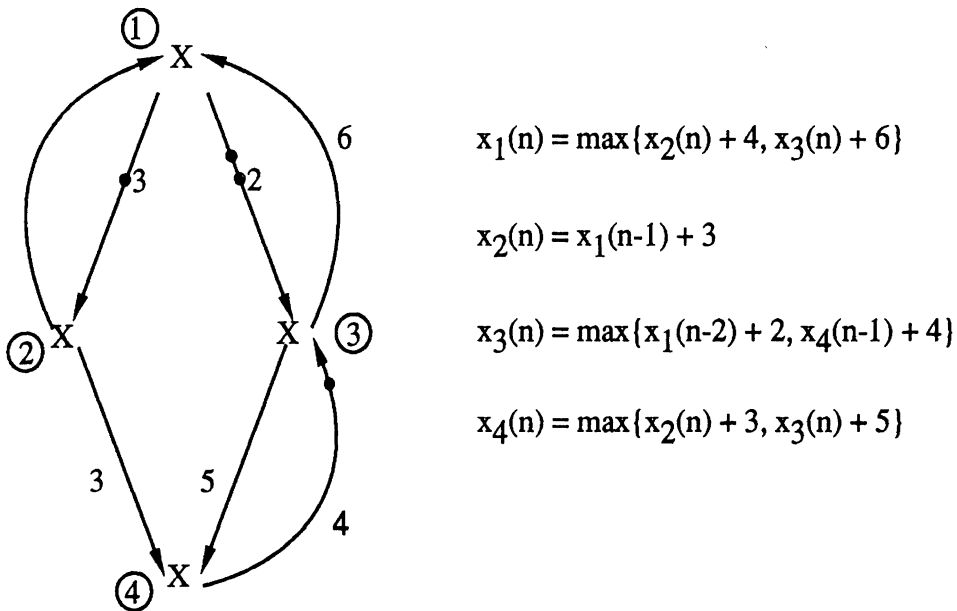


Figure 4. Petri net example with algebraic representation.

The equations of Figure 4 directly define the matrices F_0 , F_1 , and F_2 as defined by equation (3):

$$F_0 = \begin{array}{|c|c|c|c|} \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline 4 & \epsilon & \epsilon & 3 \\ \hline 6 & \epsilon & \epsilon & 5 \\ \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \end{array}$$

$$F_1 = \begin{array}{|c|c|c|c|} \hline \epsilon & 3 & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & 4 & \epsilon \\ \hline \end{array}$$

$$F_2 = \begin{array}{|c|c|c|c|} \hline \epsilon & \epsilon & 2 & \epsilon \\ \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \end{array}$$

$(F_0)^*$ is computed as:

$$(F_0)^* = \begin{array}{|c|c|c|c|} \hline e & \epsilon & \epsilon & \epsilon \\ \hline 4 & e & \epsilon & 3 \\ \hline 6 & \epsilon & e & 5 \\ \hline \epsilon & \epsilon & \epsilon & e \\ \hline \end{array}$$

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 equation (9):

$$x_2(n) = \max \{x_2(n-1) + 7, x_3(n-1) + 9\}$$

$$x_3(n) = \max \{x_2(n-1) + 7, x_3(n-1) + 9, x_5(n-1)\}$$

$$x_5(n) = \max \{x_2(n-1) + 6, x_3(n-1) + 8\}.$$

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 matrix B:

$$\begin{matrix}
 B = & \begin{matrix} \begin{matrix} \epsilon & e & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & e \\ \epsilon & \epsilon & e & \epsilon & \epsilon \\ \epsilon & \epsilon & e & \epsilon & \epsilon \end{matrix} \end{matrix} & C = & \begin{matrix} \begin{matrix} e & e & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & e & \epsilon \\ \epsilon & \epsilon & \epsilon & e \end{matrix} \end{matrix} & K = & \begin{matrix} \begin{matrix} 3 & \epsilon & \epsilon & \epsilon \\ \epsilon & 2 & \epsilon & \epsilon \\ \epsilon & \epsilon & 4 & \epsilon \\ \epsilon & \epsilon & \epsilon & e \end{matrix} \end{matrix}
 \end{matrix}$$

Multiplying these matrices provides:

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

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. 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. 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.

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_∞ (due dates, for example) to be reached at time t_∞ , with $v_r \leq t_\infty$, $r \in \theta$. From this 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 (Cunninghame-Green [1979] and Cohen et al. [1985a]):

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

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

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

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

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

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

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

and then the knowledge of $Q(n)$, $Q(n-1)$, ..., $Q(n-\Delta+1)$, 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), then it is possible to transform equation (15), when $\Delta=1$, into the dual of (12):

$$\zeta(n) = B(F_0) * CK \zeta(n+1), \quad (16)$$

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 equation (16) by adding a sufficient number of dummy vertices as in §3.4, Case b).

4. SOLVING {max, +} RECURRENCE EQUATIONS

§3 provides a linear representation of timed marked graphs in a nonstandard 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 Cunninghame-Green [1962, 1976], Gondran and Minoux [1979], and Cohen et al. [1985a]. Results on more general algebraic structures can be found in Zimmermann [1981] and Gondran and Minoux [1984]. 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 §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 λ such that $Z(n-1)M = \lambda Z(n-1)$, where $\lambda Z(n-1)$ is obtained here by adding λ 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 λ 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 Cohen et al. [1985a]) that $\lambda > \varepsilon$ and $x_i > \varepsilon, \forall i$. Moreover, given M , λ is unique and has a nice graph-theoretical interpretation. Namely λ is the maximal average weight of circuits in $G(M)$. The average weight $\bar{W}(\gamma)$ of a circuit γ 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 equation (18) as $\bar{W}(\gamma) = {}^{l(\gamma)}\sqrt{W(\gamma)}$ (see Ramchandani [1974]).

The maximum of $\bar{W}(\gamma)$ 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 $W^* = \max_{\gamma} \{\bar{W}(\gamma) | l(\gamma) \leq m\}$ be the maximal average weight of circuits. We have $W^* = \lambda$, an eigenvalue of M . Any circuit γ_0 such that $\bar{W}(\gamma_0) = \lambda$ 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 Gordan and Minoux [1984] and Cohen et al. [1985a]): Let

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

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 λ , define M_λ such that $(M_\lambda)_{ij} = M_{ij} - \lambda, \forall i,j$, which can be written: $M_\lambda = \lambda^{-1}M$. Since λ 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 Cunningham-Green [1979] (and also shown in Gondran and

Minoux [1984] and Cohen et al. [1985a]) 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:

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

where the Σ 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 (Cunninghame-Green [1979] and Cohen et al. [1985a]):

•If i and j are vertices in V_0 belonging to the same connected component, then

$$(M_\lambda)_i^+ \text{ and } (M_\lambda)_j^+ \text{ are collinear. That is, there exists a scalar } \mu \text{ such that } (M_\lambda)_i^+ = \mu (M_\lambda)_j^+.$$

Hence choosing one vertex i_ℓ in each connected component C_ℓ , we get 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 $\alpha_1, \alpha_2, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_k, \alpha$, such that:

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

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 λ '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. (See Cunninghame-Green [1979], Gondran [1979, 1984], and Cohen et al. [1985b], however.)

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:

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

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:

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

thus, generalizing equation (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. [1985a].

- 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 $C_\ell, \ell=1, \dots, K$, 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:

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

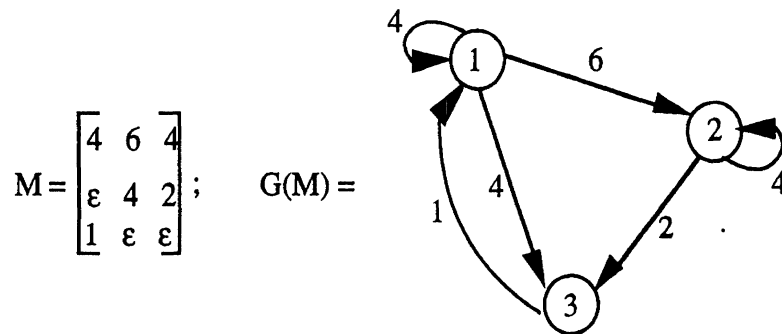
Clearly, $Q_i = (M_\lambda)^i Q_0 = Q_0 (M_\lambda)^i$. $Q_0 = PM^d$ is the projector of M^d . Q_0 can be obtained from the knowledge of a basis of generators of the eigenspace on the left, (extracted from the rows of $(M_\lambda^d)^+$, as explained in §4.1) and a basis of generators of the eigenspace on the right, extracted 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 Cunningham-Green [1979] and shown in Cohen et al. [1985a] 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_λ :

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

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:

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

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 §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. [1985a].

Proposition 4: Let M be an irreducible order-d-periodical matrix with eigenvalue λ . 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

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

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

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

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

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

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

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

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

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

$$\forall n \geq n_0, X(n) = \lambda^n X Q_i, \text{ if } n-d(n:d) = i, i=0, \dots, d-1 \text{ with } d=l.c.m. \\ (d_1, \dots, d_k) \text{ and the rank } Q_0 = \sum_{i=1}^k d_i.$$

These results are very useful to understand 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 λ^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 $\{\underline{X}^i = X Q_i \mid i=0, \dots, d-1\}$, 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 Cunningham-Green [1976] on solving $\{\max, +\}$ systems of equations of the form

$$X M = V, \tag{24}$$

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

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

$$XM \lesssim V, \quad (25)$$

where \lesssim is the usual vector-ordering: $V \lesssim V', \iff V_i \leq V'_i, \forall i$. 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 categorized 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 equation (26) has a solution, then \hat{X} is the greatest solution. ■

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:

$$\hat{Y}(N-n), n \geq 0, \text{ such that } \hat{Y}(N) = XQ_0 \text{ and } \forall n \leq N,$$

$$\hat{Y}(N-n) = \sup\{Y(N-n) \mid Y(N-n)(M_\lambda)^n \lesssim XQ_0\} \quad (26)$$

$$= {}^t((M_\lambda)^n \cdot (XQ_0)^{-1})^{-1}, \quad \text{using Theorem 4}$$

$$= {}^t(Q_i {}^t(XQ_0)^{-1})^{-1}, \quad \text{for } n \geq n_0 \text{ and } n-d(n:d) = i.$$

Hence the latest steady state is defined by the column vectors

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

(In particular, $Y^d = Q_0$.)

The following property can be established:

Proposition 6: For every $i=1, \dots, d$, $\bar{Y}^i > \underline{X}^{d-i}$.

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

$$\underline{X}^{d-i} \in \{Y(N-n) \mid Y(N-n)(M_\lambda)^n \lesssim \underline{X}^0 \triangle XQ_0\}.$$

Indeed, $\underline{X}^i (M_\lambda)^{dp+i} = XQ_0 (M_\lambda)^{d-i} (M_\lambda)^i Q_0 = XQ_0$, since $Q_0^2 = Q_0 = Q_0 (M_\lambda)^d$. But \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 events, the slack times (s_j) associated with the occurrence of events are defined by:

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

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

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

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 : $G_i \geq G_j \iff$ 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, λ_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:

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

Figure 5. Matrix M.

It is assumed that $G_1 \geq G_2$.

Proposition 7. M has a left eigenvalue equal to the $\max\{\lambda_1, \lambda_2\}$. Moreover, if $\lambda_1 > \lambda_2$, then λ_2 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

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

Hence, λ_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

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

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

Note that when $\lambda_1 > \lambda_2$, the eigenvectors of M for λ_2 are the eigenvectors of $M(2)$ completed with ε '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 λ_1 are defined by:

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

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 $G(M(1,2)) \triangleq G_{1,2}$) 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 $G_{1,2}$. 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 $X^1(n) = X^1 M(1)^n$. In the long run, if $Q_0(1)$ denotes the projector built on $M(1)$, we have $X^1(n) = X^1 Q_0(1) \lambda_1^n$, for $n - (n:d_1) = i$, if d_1 is the order of periodicity of $M(1)$.

Now

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

For a large enough n , we can compute a periodical behavior of $X^2(n)$ by noticing that λ_1 is the only interesting eigenvalue of M when G_1 has been excited as well as G_2 . This means, that there exists n_0 and $n > n_0$ such that

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

and we get $X^2(n) = X^1 Q_i(1) \lambda_1^n (M(2) \lambda_1^{-1})^*$, 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 $X^2 M(2)^n = X^2 Q_j(2) + n \lambda_2$, 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 equation (28).

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

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

Now, since $\lambda_1 > \lambda_2$, there exists n''_0 such that $n > n''_0 \implies b_n > a_n$, i.e., $(X^2(n))_k > (X^2 M(2)^n)_k$. Hence, the term $X^2 M(2)^n$ 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 §§3 and 4 on practical problems, computational tools are needed. Such tools already exist in the literature, in graph theory (Karp [1978]). These are applied to the computation of eigenvalues of matrices in the $\{\max, +\}$ algebra. 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 ε . Each arc (i,j) is labeled by M_{ij} . Let λ 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 λ 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 solved by Karp [1978], who provides a very efficient algorithm to calculate λ 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:

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

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 equation (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 [1978] 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 $k^* = m-d$,

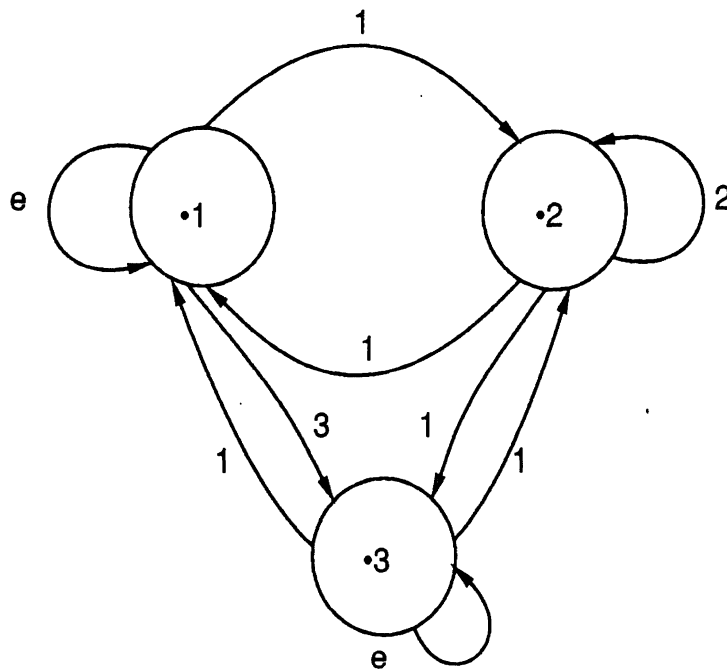
$$\lambda = \frac{(M^m)_{ij^*} - (M^{k^*})_{ij^*}}{m - k^*}.$$

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.

A =

| | | | | | |
|------------|---|---|---|------------|----------|
| | k | 0 | 1 | 2 | 3 |
| vertices j | | | | | |
| 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 this array is obtained as

$$\forall j, \quad A_{jk} = \max_{i=1, \dots, 3} \{A_{ik-1} + M_{ij}\}$$

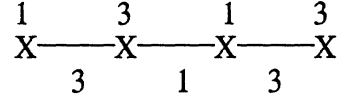
which says that $(M^k)_{1j} = (M^{k-1})_{1j}$. Now array B, containing as entry (j,k) the term $\frac{(M^3)_{ij}}{3-k}$, $k = 0, 1, \text{ and } 2$, is obtained as:

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 in the right 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 λ : $j = 2$ and 3. For $j = 3$ and $k = 1$, we

can expect to capture a critical circuit γ 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 λ

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:

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

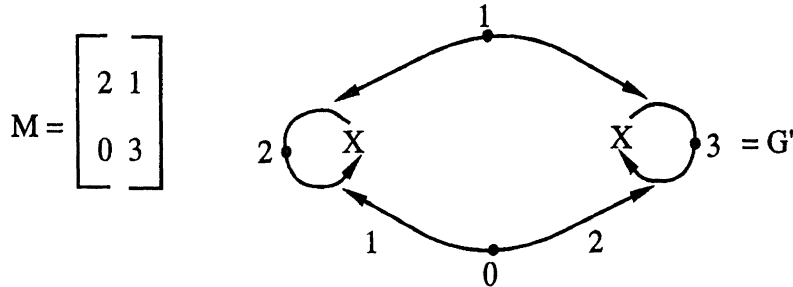
This result has been proved by Ramchandani [1974] (see also Ramamoorthy and Ho [1980]). However, Ramchandani provided no computationally efficient method for computing λ or for finding critical circuits. Moreover, Ramamoorthy and Ho only provide a polynomial algorithm to determine, given a guessed cycle 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 ϕ , since every arc has one token on it. We have $\tilde{X}(n) = \hat{X}(n) = X(n) = X_{n-1} F_1 = X_{n-1} M$. 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. [1985a]). But in practice, starting from a timed marked graph always leads to $M_{ij} \geq 0, \forall i, j$.

Now, given a critical circuit γ_0 on $G(M)$, γ_0 corresponds to a circuit γ'_0 of critical events in the original marked graph, G . Since $\lambda = \frac{w(\gamma_0)}{\tau(\gamma_0)}$ it is clear that by adding a token on the critical circuit γ'_0 , the average weight of this circuit becomes $w(\gamma_0)/(\tau(\gamma'_0) + 1)$. This may make γ_0 no longer critical. On the other hand, there is no other way of decreasing λ (i.e., to improve the performance of the marked graph), since adding tokens on other circuits would leave λ 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 §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 §3.4), B' , C' , and K' are obtained as follows:

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

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

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

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,

$$M' = B'(F'_0)^*C'K'$$

$$\begin{aligned}
 &= \begin{bmatrix} B \\ b \end{bmatrix} (F_0')^* \begin{bmatrix} C & c \end{bmatrix} K' = \begin{bmatrix} K & \varepsilon \\ \varepsilon & \underline{t_{ij}} \end{bmatrix} \\
 &= \begin{bmatrix} B(F_0')^* \\ (F_0')^*_j \end{bmatrix} \cdot \left[CK' \begin{bmatrix} \varepsilon \\ \underline{t_{ij}} \\ \varepsilon \end{bmatrix} \right] \dots i^{\text{th}} \text{ row} \\
 &= \left[\begin{array}{c|c} B(F_0')^* CK & \underline{t_{ij}} B(F_0')^*_i \\ \hline (F_0')^*_i CK & \underline{t_{ij}} (F_0')^*_j \end{array} \right] = M'.
 \end{aligned}$$

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 $\underline{i,j}$ 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 $\underline{i,j}$ is on a maximal weight path from i_l to i_k .
- There is no token on this path, except k .

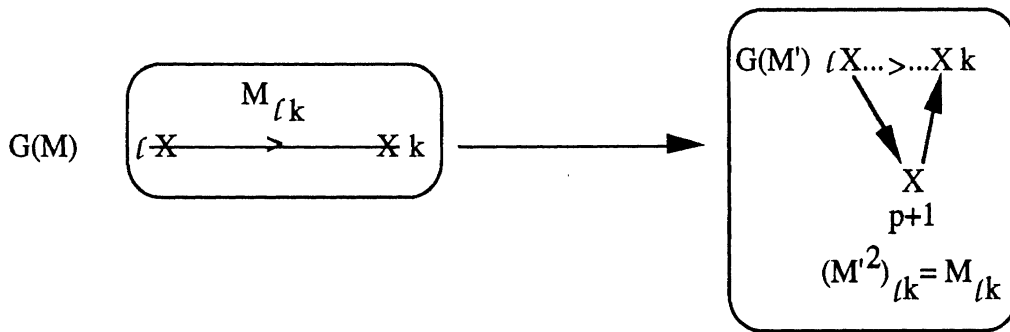
$$\begin{aligned}
 \text{Now } ((F_0')^*_j CK)_k &= \text{maximal weight of a path from } \underline{j} \text{ to } k \text{ in } G \\
 &= ((F_0')^*_j CK)_k M'_{p+1,k},
 \end{aligned}$$

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

Moreover, arc $(\underline{i}, \underline{j})$ is on a maximal-weighted path from i_ℓ to i_k in $G(F_0)$ and so $(F_0^*)_{i_\ell i_k} + t_{ij} + (F_0^*)_{j i_k} = (F_0^*)_{i_\ell i_k}$. We can conclude (multiplying by K on the right), that $M_{\ell k} = M'_{\ell, p+1} + M_{p+1, k}$, 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 $G(M)$ (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:



Notice that there may still be an arc (ℓ, k) in $G'(M')$, but we have $M'_{\ell k} \leq M_{\ell k}$.

b) **Case 2:** Arc $(\underline{i}, \underline{j})$ already has a token.

$$G = \bar{X} \xrightarrow[\substack{t_{ij} \\ p+1}]{i} \bar{X} \xrightarrow[\substack{0 \\ s}]{q} \bar{X} \xrightarrow{j} \bar{X}$$

In this case, we transform G into G' , which contains one more node q , with $t_{iq} = t_{ij}$ and $t_{qj} = 0$ (as done in §3.4). Hence F'_0 and F'_1 are defined as follows ($|V| = q-1$):

$$F'_0 = \begin{array}{|c|c|} \hline F_0 & e \\ \hline e & e \\ \hline \end{array} \quad \begin{array}{l} q \\ q \end{array}$$

since arcs (\underline{i}, q) and (q, \underline{j}) bear tokens.

That is, F'_0 remains the same, except that one row and one column of ϵ 's are added to it. Moreover, F'_1 has also one more row and column. Rows 1 to $q-1$ and columns 1 to $q-1$ of F'_1 are the same as those of F_1 , except that $(F'_1)_{ij} = \epsilon$ (because of the deletion of arc

(i, j). Moreover, row (resp: column) q of F'_1 is filled with ϵ 's, except that row $(F'_1)_{qj} = 0$ (resp: $(F'_1)_{iq} = t_{ij}$). That is,

| | | Columns | |
|----------|-------|------------|------------|
| | | j | q |
| $F'_1 =$ | Row i | | ϵ |
| | | ϵ | t_{ij} |
| | | | ϵ |
| | Row q | ϵ | 0 |

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

| | | j | q |
|--------|-----|------------|------------|
| $B' =$ | s | B | 0 |
| | | | ϵ |
| | p+1 | ϵ | 0 |

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

| | | s | p+1 |
|--------|---|------------|------------|
| $C' =$ | i | C | ϵ |
| | | ϵ | 0 |
| | | C | ϵ |
| | q | ϵ | ϵ |

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

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

That is, $(K')_{ii} = K_{ii}$, except that $K'_{ss} = 0$.

$$K'_{p+1,p+1} = t_{ij}$$

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

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

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

Hence $M' =$

$$\left[\begin{array}{c|cc|c} & s & & p+1 \\ \hline BF_0^*CK & \epsilon & BF_0^*CK & t_{ij}(BF_0^*)_{\underline{i}} \\ \hline & \epsilon & & \\ \hline & \epsilon & & \\ \hline \epsilon \epsilon \epsilon & 0 & \epsilon \epsilon & \epsilon \end{array} \right]$$

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

Indeed,

since $C_{is} = 0$.

$$M_{.s} = (B(F_0) * CK)_{.s} = B(F_0) * C(K)_{.s} = B(F_0) * \begin{pmatrix} \epsilon \\ \epsilon \\ \underline{t_{ij}} \\ \epsilon \end{pmatrix}, \text{ where } \underline{t_{ij}} \text{ is on row } i,$$

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 $M = \{M(1,s-1), M_{.s}, M(s+1,p)\}$,

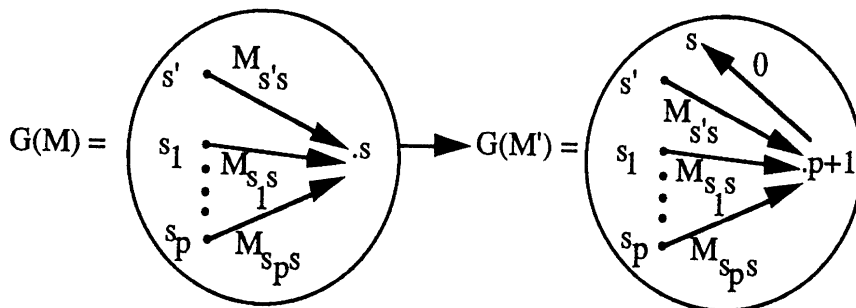
Then $M' =$

| | | | |
|------------|------------|------------|------------|
| $M(1,s-1)$ | ϵ | $M(s+1,p)$ | $M_{.s}$ |
| | ϵ | | |
| | ϵ | | |
| ϵ | 0 | ϵ | ϵ |

s

$G(M')$ 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 [1989].

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 open 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). As 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. [1985b, 1987].

ACKNOWLEDGEMENTS

We gratefully acknowledge the expert evaluations of three anonymous Referees each for both this and the associated applications paper as well as the Associate Editors and the Editors, Joe Mazzola and Maurice Queyranne.

BIBLIOGRAPHY

- AJMONE MARSAN, MARCO, BALBO, GIANFRANCO, and CONTE, GIOVANNI, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems", ACM Transactions on Computer Systems, Vol. 2, No. 2, pp. 93-122 (May 1984).
- AJMONE MARSAN, MARCO, BALBO, GIANFRANCO, and CONTE, GIOVANNI, Performance Models of Multiprocessor Systems, The MIT Press, Cambridge, MA (1986).
- BEL, GERARD and DUBOIS, DIDIER, "A Unified Setting for Various Models of Automated Material Flow and Production Systems", Computers in Industry, Vol. 6, pp. 227-235 (1985).
- BROWNE, JIM, DUBOIS, DIDIER, RATHMILL, KEITH, SETHI, SURESH P., and STECKE, KATHRYN E., "Classification of Flexible Manufacturing Systems", The FMS Magazine, Vol. 2, No. 2, pp. 114-117 (April 1984).
- BUZACOTT, JOHN A. and YAO, DAVID D. W., "Flexible Manufacturing Systems: A Review of Analytical Models", Management Science (1986).
- CARRÉ, B. A., "An Algebra for Network Routing Problems," Journal of the Institute of Mathematical Applications, Vol. 7, pp. 273-294 (1971).
- CHRETIENNE, PHILIPPE, "Les Réseaux de Petri Temporisés," Thèse d'État, Université de Paris VI, France (June 1983).
- COHEN, GUY, DUBOIS, DIDIER, QUADRAT, JEAN PIERRE, and VIOT, MICHEL, "A Linear-System Theoretic View of Discrete-event Processes and Its Use for Performance Evaluation in Manufacturing", IEEE Transactions on Automatic Control, Vol. AC-30, No. 3, pp. 210-220 (March 1985a).
- COHEN, GUY, MOLLER, P., QUADRAT, JEAN PIERRE, and VIOT, MICHEL, "Linear System Theory for Discrete-event Systems", Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, Nevada, pp. 539-544 (December 1984).
- COHEN, GUY, MOLLER, P., QUADRAT, JEAN PIERRE, and VIOT, MICHEL, "Une Théorie Lineaire des Systemes à Evénements Discrets," Rapport No. 362, INRIA, Rocquencourt, France (1985b).
- COHEN, GUY, MOLLER, P., QUADRAT, JEAN PIERRE, and VIOT, MICHEL, "Dating and Counting Events in Discrete Event Systems", Proceedings of the 25th IEEE Conference on Decision and Control, Athens, Greece (December 1986).
- COHEN, GUY, MOLLER, P., QUADRAT, JEAN PIERRE, and VIOT, MICHEL, "A 2-Discrete Event Linear System Theory", Algebres Exotiques et Systemes à Evénements Discrets, INRIA, Rocquencourt, France, pp. 147-165 (June 1987).
- COMMONNER, F., HOLT, A. W., EVEN, S., and PNUELI, A., "Marked Directed Graphs," Journal of the Computer Systems Society, Vol. 5, pp. 511-523 (1971).

- CUNNINGHAME-GREEN, RAYMOND A., "Describing Industrial Processes and Approximating Their Steady-state Behavior", Operational Research Quarterly, Vol. 13, pp. 95-100 (1962).
- CUNNINGHAME-GREEN, RAYMOND A., "Projections in a Minimax Algebra", Mathematical Programming, Vol. 10, pp. 111-123 (1976).
- CUNNINGHAME-GREEN, RAYMOND A., Minimax Algebra (Lecture Notes in Economics and Mathematical Systems, Vol. 166), Springer Verlag, New York (1979).
- DIAZ, MICHEL, "Modeling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Models", Computer Networks, Vol. 6, No. 6, pp. 419-441 (December 1982).
- DUBOIS, DIDIER and STECKE, KATHRYN E., "Using Petri Nets to Represent Production Processes", Proceedings of the 22nd IEEE Conference on Decision and Control, San Antonio, Texas, pp. 1062-1067 (December 1983).
- DUBOIS, DIDIER and STECKE, KATHRYN E., "Dynamic Analysis of Repetitive Decision-free Discrete Event Processes: Applications to Production Systems", Working Paper No. 543-b, DOR, The University of Michigan, GSBA, Ann Arbor MI (October 1989).
- FLOYD, R. W., "Shortest Paths," Communications of the Association of Computing Machinery, p. 345 (1962).
- GONDRAN, M., "Les Eléments p-réguliers dans les Dioïdes", Discrete Mathematics, Vol. 25, pp. 33-39 (1979).
- GONDRAN, M. and MINOUX, M., Graphes et Algorithmes, Eyrolles Editions, Paris, France (1979).
- GONDRAN, M. and MINOUX, M., "Algebraic and Combinational Methods in Operations Research," (R. E. Burkard, R. A. Cunningham-Green and U. Zimmermann, Editors), Annals of Discrete Mathematics, Vol. 19, North Holland, Amsterdam (1984).
- KARP, RICHARD M. "A Characterization of the Minimum Cycle Mean in a Digraph," Discrete Mathematics, Vol. 23, pp. 309-311 (1978).
- PETERSON, JAMES L., Petri Net Theory and the Modeling of Systems, Prentice-Hall, Inc., Englewood Cliffs NJ (1981).
- RAMAMOORTHY, C. V. and HO, GARY S., "Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets", IEEE Transactions on Software Engineering, Vol. 6, No. 5, pp. 440-449 (September 1980).
- RAMCHANDANI, CHANDER, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets", Ph.D. dissertation, Department of Electrical Engineering, MIT, Cambridge MA (1974).
- REITER, R., "Scheduling Parallel Computations", Journal of the Association of Computing Machinery, Vol. 15, pp. 590-599 (1968).

SMIGELSKI, T., MURATA, T. and SOWA, M., "A Timed Petri Net Model and Simulation of a Dataflow Computer", International Workshop on Timed Petri Nets, Torino, Italy, pp. 56-63 (July 1-3, 1985).

ZIMMERMANN, U., "Linear and Combinational Optimization in Ordered Algebraic Structures", Annals of Discrete Mathematics, Vol. 10, North Holland, Amsterdam (1981).