Theoretical Computer Science 407 (2008) 496–510

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Time separations of cyclic event rule systems with min–max timing constraints

Qianchuan Zhao *, Jianfeng Mao, Ye Tao

Center for Intelligent and Networked Systems, Department of Automation, TNLIST, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 22 June 2005
Received in revised form 22 July 2008
Accepted 4 August 2008
Communicated by Z. Esik

Keywords:
Discrete event systems
Time separations
Event rule systems
Min–max systems
Cyclic timing constraint graph

ABSTRACT

The analysis of the time separations of events is a fundamental problem in the design and evaluation of discrete event systems. Important progresses have been made based on the event rule system model in the last decade. The existing results for event rule systems with min and max constraints can be summarized briefly as: the exact evaluation of time separations for acyclic systems is NP-complete; for cyclic systems, the structural condition of being tightly coupled is sufficient for long-term time separations of events to be bounded. In this paper, we establish a necessary and sufficient structural boundedness condition—uniformity for cyclic event rule systems with both min and max constraints. Tightly coupled systems are shown to be a special class of uniform systems. The well-known CAS algorithm for finding bounds on long-term time separations is adapted to find finite bounds for uniform systems. Our results are obtained by exploring the algebraic structures guiding the evolution of the systems.

1. Introduction

The evolution of discrete event systems is driven by events. When designing a system, the timing of events is crucial for the system to be both safe and efficient. The analysis of the time separations of events is a central problem in the design and evaluation of timing aspects of discrete event systems. Most of the important progress on this topic has been made in the event rule system framework in the last decade [17,16,26,15,27,4,14]. The framework of event rule systems was introduced for performance analysis of asynchronous circuits by Burns in [2].

For acyclic event rule systems, the problems of interest include:

A1. Develop algorithms to determine the exact time separation of a given pair of events.
A2. If the exact determination of separation is hard, develop fast approximation algorithms.

For **A1**, in [17], a polynomial algorithm to calculate exact time separations was developed for systems with max-only constraints. It was also shown in [17] that the problem of deciding the exact time separations for general acyclic event rule systems with both min and max constraints is known to be NP-complete. In [26,27], algorithms are developed for analyzing acyclic systems with max and linear constraints. The algorithms are conjectured to run in polynomial-time. The exact complexity of the max and linear problem is unfortunately unknown. For **A2**, in [4], McMillan and Dill's time separation algorithm was extended to acyclic event rule systems with both min and max constraints.

For cyclic event rule systems, the problems of interest include:

C1. Establish conditions under which the long-term time separations of events are bounded.

* Corresponding author. Tel.: +86 10 62783612; fax: +86 10 62786911.
E-mail address: zhaoqc@tsinghua.edu.cn (Q. Zhao).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.08.014

C2. Develop algorithms to determine time separations of events or their upper bounds.

For **C1**, based on a modification of McMillan and Dill's time separation algorithm and deep algebraic observations, [16] established that for a class of cyclic max-only event rule systems known as well-formed[1] strongly connected systems, the long-term time separations are bounded. It was extended to the exact timing analysis of a class of CSP programs allowing restricted choices [15]. [4] extended the notion of strong connectivity for well-formed max-only event rule systems to the condition of tightly coupled for well-formed event rule systems with both min and max constraints, and proved that the long-term time separations are bounded for well-formed tightly coupled systems. For **C2**, in [19], a polynomial-time algorithm was described for approximate timing analysis of max-only systems with repeated events. [16] proposed an exact algorithm to determine the exact time separations for cyclic well-formed strongly connected max-only event rule systems. As an extension, a symbolic algorithm based on presburger arithmetic was used in [14] to obtain symbolic expressions of long-term time separations in terms of variables representing component delays. In [4], a pseudopolynomial algorithm known as CyclicApproxSep(CAS) algorithm was proposed to determine an upper bound for the long-term time separations of events for cyclic event rule systems with both min and max constraints (see also [5]).

In this paper, we study **C1** and establish a new structural condition for bounding long-term separations in cyclic event rule systems with both min and max constraints. The new condition is captured by the notion of uniformity which was introduced in [28] in the context of stochastic min–max systems. Our main finding is that in terms of structure, uniform systems are the broadest class of systems with both min and max constraints having finite long-term time separations. We also touch on **C2** by first giving an example of uniform systems where the CAS algorithm in [4] fails to give finite bounds, and by then introducing a modification to CAS so that finite bounds can be obtained. Our method is largely based on the formal model of min–max systems which was first introduced by Gunawardena in [12] to model event rule systems. The existing work under the min–max framework is mostly focused on cycle time analysis (see e.g., [20,13,10,8,6,24] and for well-known special class of max-plus systems, see e.g., [9,1]). Related to our work, [21,25,28] studied the structural properties of min–max systems with focus on existence of global cycle time.

The rest of this paper is organized as follows. In Section 2, we present the definition of cyclic timing constraint graphs (introduced in [4]) as the formal model of cyclic event rule system and present the definition of uniform systems and show that the tightly coupled systems proposed in [4] are a special class of uniform systems. Then in Section 3, we prove that for uniform systems with bounded component delays, the long-term time separations of events are bounded. So we extend the sufficient condition of problem **C1** from tightly coupled systems to a broader type of cyclic event rule systems—uniform systems. Furthermore, we prove that for non-uniform systems, there must be specific edge delay assignments such that the long-term time separations are unbounded. We discuss the test of uniformity in Section 4. In Section 5, we study an existing algorithm—the CAS algorithm given in [4] for problem **C2**. We first present one example of uniform systems where the CAS algorithm fails to give finite bounds. A simple modification to CAS is then given to find finite bounds for all uniform systems. Possible applications of our main results are discussed in Section 6 including a class of systems with choice. At last, Section 7 concludes the paper.

2. Models, definitions and notations

2.1. Cyclic timing constraint graphs

Graphically, cyclic event rule systems can be specified by cyclic timing constraint graphs. In the following, we present the cyclic timing constraint graph model introduced in [4] for describing cyclic event rule systems. The readers can find more details there.

To motivate the event rule system models and their graphical description in cyclic timing constraint graphs, let us first introduce an example before presenting the formal definitions.

Example 1. Fig. 1 shows a circuit similar to the example in [18] (on page 306 Fig. 8.5(b)). It contains three gates: one AND and two NOTs. Fig. 2 is its cyclic timing constraint graph G, in which we use vertices a ,b, c, d, e, f to represent the events x^+ (raising event of signal x), y^+, z^+, x^- (falling event of signal x), y^-, z^- respectively. Fig. 3 is a portion of its unfolded graph G^*. G_0, G_i and C_i are also visualized in it.

Formally, a *cyclic timing constraint graph* of a cyclic event rule system is a directed graph $G = (V, E)$ equipped with edge delays defined as follows.

(1) $G = (V, E)$: G is a directed graph. V is the vertex set of G; E is the edge set of G.
(2) $v \in V$: v is a vertex in V, which represents the corresponding event. There is a unique root vertex RESET in V. It has no input edges. The set of vertices V is divided into two disjoint subsets: the set of min vertices (shown as circles '◯' in figures) and the set of max vertices (shown as squares '□' in figures). The events corresponding to min-type vertices occur when *one* of their input constraints is satisfied; the events corresponding to max-type vertices occur when *all* of their input constraints are satisfied.

[1] The definition of well-formed systems will be given in Section 2.1 as can be found in [16].

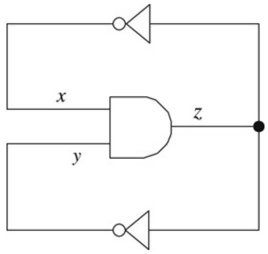

Fig. 1. A circuit.

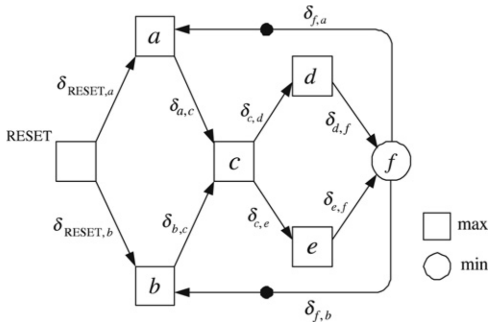

Fig. 2. Cyclic timing constraint graph G of the system in Fig. 1.

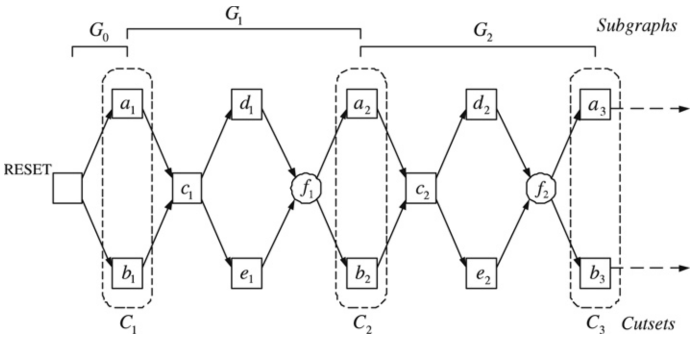

Fig. 3. A portion of unfolded graph G^* of the cyclic timing constraint graph in Fig. 2.

(3) $\langle u, v\rangle \in E : \langle u, v\rangle$ is the edge in E, which represents the event rule from the event represented by u to the one represented by v. The set of edges E is also divided into two disjoint subsets: the set of *marked* edges (indicated by '•' on edges) and the set of *unmarked* edges. The timing constraints are different for marked and unmarked edges. A marked edge from u to v specifies a constraint imposed by the k-th occurrence of u on the $k+1$-th occurrence of v; An unmarked edge from u to v specifies a constraint of the k-th occurrence of u on the k-th occurrence of v.
(4) $\delta_{u,v} \in [d_{u,v}, D_{u,v}] : \delta_{u,v}$ is the delay associated with the edge $\langle u, v\rangle$; $d_{u,v}$ and $D_{u,v}$ are the lower and upper bounds of the delay $\delta_{u,v}$ respectively, where $d_{u,v}$ and $D_{u,v}$ are finite.

A cyclic timing constraint graph G can be unfolded to an infinite graph (known as unfolded graph of G). Formally, the *unfolded graph* G^* is defined as follows.

(1) $G^* = (V^*, E^*) : G^*$ is the unfolded graph of G; V^* is the vertex set of G^*; E^* is the edge set of G^*.
(2) $u_k \in V^* : u_k$ is the vertex in V^*, which represents the k-th occurrence of the event represented by u.
(3) $\langle u_k, v_{k'}\rangle \in E^* : \langle u_k, v_{k'}\rangle$ is the edge in E^*, which represents the rule from the event represented by u to the one represented by v.
(4) $\tau_{v_{k'}}$ is the time of k'-th occurrence of the event represented by v as below, where $\text{preds}(v_{k'})$ is the set of vertices with an edge to $v_{k'}$.
   - If v is a max-type vertex, $\tau_{v_{k'}} = \max_{u_k \in \text{preds}(v_{k'})}(\tau_{u_k} + \delta_{u_k,v_{k'}})$;
   - If v is a min-type vertex,[2] $\tau_{v_{k'}} = \min_{u_k \in \text{preds}(v_{k'})}(\tau_{u_k} + \delta_{u_k,v_{k'}})$.

[2] The event occurring earliest will trigger the occurrence of a min-type event (with a delay). Here we assume that for the k'-th occurrence of event v, only the first occurring event among all the events in $\text{preds}(v)$ driving a min-type event v will take effect, the k'-th occurrences of other events in $\text{preds}(v)$ will be ignored.

(5) C_i is the *i*-th cutset of G^*, which should satisfy the following three properties which are the properties from **P1** to **P3** in the definition of tightly coupled mentioned in [4]:
- The size of $\mathcal{R}'(C_1)$ is finite[3];
- $C_i = \{v_i : v_1 \in C_1\}$ after the relabeling process;
- $C_i \bigcap C_j = \emptyset, \forall i \neq j$.

(6) $G_i = (V_i, E_i) : G_i$ is the subgraphs of G^* containing the portion between (including) cutsets C_i and C_{i+1}; V_i is the set of vertices in G_i and E_i is the set of edges in G_i, i.e.,
- $V_i = \mathcal{R}(C_i) \bigcap (\mathcal{R}'(C_{i+1}) \bigcup C_{i+1})$[4];
- $E_i = \{\langle u, v\rangle : u, v \in V_i \text{ and } \langle u, v\rangle \in E^*\}$.

(7) $G_0 = (V_0, E_0) : G_0$ is the subgraph of G^* containing the portion from RESET to the cutset C_1; V_0 is the set of vertices in G_0 and E_0 is the set of edges in G_0, i.e.,
- $V_0 = \mathcal{R}'(C_1) \bigcup C_1$;
- $E_0 = \{\langle u, v\rangle : u, v \in V_0 \text{ and } \langle u, v\rangle \in E^*\}$.

A cyclic timing constraint graph is said to be *well-formed*, if every cycle has at least one marked edge and for every event v in the cyclic component, there exists at least one cycle with exactly one marked edge. All cyclic timing constraint graphs mentioned in the rest of this paper are well-formed by default.

To study the long-term time separations of events in our systems, we also need to define their state function and state vectors with the help of two definitions from [13]:

(1) A min–max expression is a term f in the grammar: $f := x_1, x_2, \ldots | f + a | f \vee f | f \wedge f$, where $x_1, x_2, \ldots$ are real variables, $a \in \mathbf{R}$ is referred to as a parameters and the infix operators $\vee$ and $\wedge$ are used to stand for the maximum and minimum operators respectively, i.e., $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$.
(2) A min–max function of dimension n is a function $F : \mathbf{R}^n \to \mathbf{R}^n$, each of whose components, $F_i : \mathbf{R}^n \to \mathbf{R}$, is a min–max expression of n variables $x_1, \ldots, x_n$.

Now we define the state vectors and state function for cyclic timing constraint graphs:

(1) State vector : denoted as T_i, and $T_i = (\tau_{v_i} | v_i \in C_i)_{n\times 1} \in \mathbf{R}^n$.
(2) State function : denoted as F, and $F : \mathbf{R}^n \to \mathbf{R}^n$ is a min–max function of dimension n, where $T_{i+1} = F(T_i)$ and $n = |C_i|$.

To avoid ambiguity, we define two types of equalities for min–max functions in the rest of paper. For two min–max functions F and H,

(1) $F(X) = H(X)$: means $F(X)$ and $H(X)$ have same values for a specific $X \in \mathbf{R}^n$.
(2) $F = H$: means $F(X) = H(X)$ for all $X \in \mathbf{R}^n$.

It is clear that T_i can be expressed as a min–max function of T_1. Denote this function by F^{i-1}.We assume that F^0 is the identity function. In terms of the composition functions, it is easy to verify that $F^i = F(F^{i-1})$ for all $i \geq 1$.

Example 1 (*Continued*). The state function of the cyclic event rule system in Fig. 2 is:

$$T_{i+1} = F(T_i) = \begin{pmatrix} ((\tau_{a_i} + \delta_{a_i,c_i}) \vee (\tau_{b_i} + \delta_{b_i,c_i}))+ \\ \qquad ((\delta_{c_i,d_i} + \delta_{d_i,f_i}) \wedge (\delta_{c_i,e_i} + \delta_{e_i,f_i})) + \delta_{f_i,a_{i+1}}, \\ ((\tau_{a_i} + \delta_{a_i,c_i}) \vee (\tau_{b_i} + \delta_{b_i,c_i}))+ \\ \qquad ((\delta_{c_i,d_i} + \delta_{d_i,f_i}) \wedge (\delta_{c_i,e_i} + \delta_{e_i,f_i})) + \delta_{f_i,b_{i+1}} \end{pmatrix}, \tag{1}$$

where $T_i = (\tau_{a_i}, \tau_{b_i})^{\mathrm{T}}, i = 1, 2, \ldots$ are the state vectors.

2.2. Uniform systems

Uniform systems are defined based on the structure of cyclic event rule systems. The property of uniformity is inspired by an obvious but extremely useful fact that for each cyclic timing constraint graph G, we can associate a unique pure min–max network $\hat{G}$. By pure, we mean that all delays labeled on the edges of G are set to zeros, i.e., the state functions of $\hat{G}$ are composed of only the two operators $\vee$ and $\wedge$. We refer to $\hat{G}$ as the *skeleton* of G. It is obvious that $\hat{G}^*$, the unfolded graph of $\hat{G}$, has the same cutsets as G. The unfolded graph $\hat{G}^*$ can also be obtained by setting all edge delays in G^* to zeros.

In $\hat{G}$, we define its state function as a pure min–max function $\hat{F} : \mathbf{R}^n \to \mathbf{R}^n$ by introducing the two definitions below:

(1) A pure min–max expression is a term f in the grammar: $f := x_1, x_2, \ldots | f \vee f | f \wedge f$, where $x_1, x_2, \ldots \in \mathbf{R}$ are variables.
(2) A pure min–max function of dimension n is a function $F : \mathbf{R}^n \to \mathbf{R}^n$, each of whose components, $F_i : \mathbf{R}^n \to \mathbf{R}$, is a pure min–max expression of n variables $x_1, \ldots, x_n$.

[3] $\mathcal{R}'(C_i)$ is the set of vertices unreachable from all vertices in C_i, see [4] for its formal definition.
[4] $\mathcal{R}(C_i)$ is the set of vertices reachable from at least one vertex in C_i, see [4] for its formal definition.

Example 1 (*Continued*). By setting all delays to zeros in (1), we obtain the state function of the skeleton $\hat{G}$ of G in Fig. 2. It is given by the pure min–max function as below:

$$\hat{F}(T_i) = \begin{pmatrix} \tau_{a_i} \vee \tau_{b_i} \\ \tau_{a_i} \vee \tau_{b_i} \end{pmatrix}, \tag{2}$$

where $T_i = (\tau_{a_i}, \tau_{b_i})^{\mathrm{T}}$.

Now we define uniform systems stated in terms of the skeletons of cyclic timing constraint graphs. The notion of uniformity was first defined in the min–max systems context in [28].

Definition 1. A cyclic timing constraint graph G is said to be uniform, if there exists a non-negative integer $r \geq 1$ such that,

$$\hat{F}_i^r = \hat{F}_j^r, \quad \forall i, j = 1, \ldots, n. \tag{3}$$

The minimal such integer r is called the height of G. Here F_i^r is the i-th component of F^r.

Example 1 (*Continued*). The system in Example 1 is uniform because $\hat{F}_1(X) = \hat{F}_2(X) = x_1 \vee x_2$ for all $X \in \mathbf{R}^2$, where $X = (x_1, x_2)^{\mathrm{T}}$. Its height is 1.

A natural question is raised, i.e., what is the relationship between uniform systems and tightly coupled systems that is also stated in terms of the skeleton of a cyclic timing constraint graph. To answer this question, let us recall first the definition of tightly coupled systems introduced in [4,3].

Definition 2. [4,3] A cyclic timing constraint graph G is tightly coupled, if the unfolded graph G^* has a sequence of cutsets, C_i for all $i \geq 1$, such that the following condition is satisfied: for every vertex u in C_i and every vertex v in C_{i+1}, there exists at least one path from u to v, such that all vertices along the path are associated with max constraints.

Proposition 1. *Tightly coupled systems are uniform systems with height $r = 1$ with*

$$\hat{F}_u = \hat{F}_v = (\vee_{v_1 \in C_1} \tau_{v_1}), \quad \forall u, v.$$

Proof. Observe that for a path $(v_1, \ldots, x, y, \ldots, u_2)$ from $v_1 \in C_1$ to $u_2 \in C_2$, if all vertices along it are max constraints, then in $\hat{G}^*$ (the unfolded graph of $\hat{G}$), all vertices along the path $(v_1, \ldots, x, y, \ldots, u_2)$ from v_1 to u_2 will be max vertices, thus for the state functions in $\hat{G}$, we have

$$\tau_{u_2} \geq \cdots \geq \tau_y \geq \tau_x \geq \cdots \tau_{v_1}.$$

So, from the property of tightly coupled systems, we can see that for every $u_2 \in C_2$,

$$\tau_{u_2} \geq \vee_{v_1 \in C_1} \tau_{v_1}.$$

At the same time, $\vee_{v_1 \in C_1} \tau_{v_1}$ is the maximal pure min–max function from $\mathbf{R}^n$ to $\mathbf{R}$. We can conclude that in $\hat{G}^*$, for every $u_2 \in C_2$,

$$\tau_{u_2} = \vee_{v_1 \in C_1} \tau_{v_1}.$$

Thus, the pure min–max function $\hat{F}$ from T_1 to T_2 in $\hat{G}^*$ must satisfy

$$\hat{F}_u = \hat{F}_v = (\vee_{v_1 \in C_1} \tau_{v_1}), \quad \forall u, v.$$

This implies that tightly coupled systems are uniform with $r = 1$. □

Proposition 1 implies that tightly coupled systems are special cases of uniform systems. Not all uniform systems (even with height $r = 1$) are tightly coupled. The uniform system in Example 1 turns out to be a non-tightly coupled system. In fact, every path from a_i to a_{i+1} contains a min vertex f_i.

3. Uniformity as a structural condition for boundedness

In this section, we present the boundedness results based on the structural property of uniformity. Let us define some convenient notations for the unfolded graph G^*. Let $|C_i| = n$ and $|G_i| = m$ for all $i \geq 1$ in the following notations:

(1) $\Lambda_i(u_i, v_i)$ is the exact time separation from the vertex u_i to the vertex v_i in C_i.
(2) $\Delta_i(u_i, v_i)$ is the exact time separation from the vertex u_i to the vertex v_i in G_i.
(3) $\Lambda_i : \Lambda_i = (\Lambda_i(u_i, v_i); u_i, v_i \in C_i)_{n \times n}$ is the matrix of pairwise exact time separations of vertices in cutset C_i.
(4) $\Delta_i : \Delta_i = (\Delta_i(u_i, v_i); u_i, v_i \in G_i)_{m \times m}$ is the matrix of pairwise exact time separations of vertices in subgraph G_i.
(5) $\Pi(u)$ is the set of paths from the vertices in cutset C_1 to the vertex u.
(6) $L(u)$ is the largest path delay to vertex u, i.e., $L(u) = \max_{P \in \Pi(u)}(\sum_{\langle x,y \rangle \text{along } P} D_{x,y})$.
(7) $l(u)$ is the least path delay to vertex u, i.e., $l(u) = \min_{P \in \Pi(u)}(\sum_{\langle x,y \rangle \text{along } P} d_{x,y})$.

(8) $\bar{\Lambda}_i(u_i, v_i) : \bar{\Lambda}_i(u_i, v_i) = L(v_i) - l(u_i)$.
(9) $\bar{\Lambda}_i : \bar{\Lambda}_i = (\bar{\Lambda}_i(u_i, v_i); u_i, v_i \in C_i)_{n\times n}$.
(10) Define some predicates of matrices as follows. Let A, B be the two matrices with same size and $\star$ represent $>, \geq, <, \leq, =$:

$$A \star B \Leftrightarrow \forall i, j : A(i, j) \star B(i, j).$$

(11) Define two matrix operations as follows. Let A, B be the two matrices with same size and $\star$ represent $\vee$ or $\wedge$. $A \star B = C$, where $\forall i, j, C(i, j) = A(i, j) \star B(i, j)$
(12) $\Lambda_k^* : \Lambda_k^* = \vee_{i\geq k}(\Lambda_i)$ is the matrix of least upper bounds of pairwise time separations of vertices in the cutsets C_i for $i \geq k$.
(13) $\Delta_k^* : \Delta_k^* = \vee_{i\geq k}(\Delta_i)$ is the matrix of least upper bounds Δ of pairwise time separations of vertices in the subgraphs G_i for $i \geq k$.

The problem **C1** is equivalent to determining whether Δ_k^* is bounded for some k. Since C_i is contained in G_i, it holds that $\Lambda_i(u_i, v_i) = \Delta(u_i, v_i)$ for all $u_i, v_i \in C_i$. Furthermore, if Λ_i is finite (i.e., all elements of Λ_i as a $n \times n$ matrix are finite), then Δ_i must also be finite since all edge delays in G_i are bounded. Thus, **C1** is reduced to determining whether Λ_k^* is bounded for some k. In the following, we study **C1** by investigating the boundedness of Λ_k^*.

Theorem 1. *If G is a uniform system with height r, then $\Lambda_i \leq \bar{\Lambda}_{r+1}$ for all $i \geq r + 1$ regardless of the value of Λ_1.*

Remark. From Theorem 1, we establish $\Lambda_{r+1}^* \leq \bar{\Lambda}_{r+1}$. This implies that the long-term time separations of events must be bounded for all uniform systems no matter what the initial state and edge delay intervals are.[5] So, uniformity is sufficient for long-term time separations to be bounded. This generalizes the boundedness result of tightly coupled systems established by [4].

Proof. Let $U_i = R(C_i) \bigcap (R'(C_{r+i}) \bigcup C_{r+i})$, $i \geq 1$. Since G_i is isomorphic to G_1 for all $i \geq 2$, U_i must also be isomorphic to U_1 for all $i \geq 2$. Thus, the largest path length and the shortest path length from C_i to $u_{r+i} \in C_{r+i}$ must be equal to $L(u_{r+1})$ and $l(u_{r+1})$ for all $i \geq 2$.

From the monotone property of min and max operations, it is straightforward to see that for all vertices v in G, if v is of max type, it must hold for the state τ_v in G^* that

$$\max_{k\in \text{preds}(v)} (\tau_k + d_{k,v}) \leq \tau_v \leq \max_{k\in \text{preds}(v)} (\tau_k + D_{k,v}).$$

If v is of min type,

$$\min_{k\in \text{preds}(v)} (\tau_k + d_{k,v}) \leq \tau_v \leq \min_{k\in \text{preds}(v)} (\tau_k + D_{k,v}).$$

Let all edges $\langle u, v\rangle$ in U_i have fixed delays $D_{u,v}$ and denote the occurrence time of vertex $v \in U_i$ as τ_v^+, we have $\tau_v \leq \tau_v^+$. Due to the facts that for four real numbers x, y and a, b,

$$\max(x + a, y + b) \leq \max(x, y) + \max(a, b)$$
$$\min(x + a, y + b) \leq \min(x, y) + \max(a, b)$$

we can show recursively that

$$\tau_{v_{r+i}} \leq \tau_{v_{r+i}}^+ \leq \hat{F}_v^r(T_i) + L(v_{r+1}), \quad \forall v_{r+i} \in C_{r+i}.$$

Recall that T_i is the state vector of C_i for G. Dually, we can obtain

$$\tau_{u_{r+i}} \geq \hat{F}_u^r(T_i) + l(u_{r+1}), \quad \forall u_{r+i} \in C_{r+i}.$$

Thus, we can establish

$$\tau_{v_{r+i}} - \tau_{u_{r+i}} \leq \hat{F}_v^r(T_i) + L(v_{r+1}) - \hat{F}_u^r(T_i) - l(u_{r+1}), \quad \forall u_{r+i}, v_{r+i} \in C_{r+i}.$$

From the uniformity of the system, i.e., $\hat{F}_v^r = \hat{F}_u^r, \forall u, v \in C_1$, we have $\hat{F}_v^r(T_i) = \hat{F}_u^r(T_i)$ for all $i \geq 1$. Thus, for all $T_i \in \mathbf{R}^n$,

$$\tau_{v_{r+i}} - \tau_{u_{r+i}} \leq L(v_{r+1}) - l(u_{r+1}), \quad \forall u_{r+i}, v_{r+i} \in C_{r+i}.$$

Note that the bound is independent of the values of T_i provided, we can establish the inequality $\Lambda_{r+i}(u_{r+i}, v_{r+i}) \leq L(v_{r+1}) - l(u_{r+1}) = \bar{\Lambda}_{r+1}(u_{r+1}, v_{r+1})$ for all vertices $u_{r+i}, v_{r+i} \in C_{r+i}$ and $i \geq 1$, or in matrix term $\Lambda_i \leq \bar{\Lambda}_{r+1}$ for all $i \geq r + 1$. $\square$

[5] As long as the delay intervals are finite.

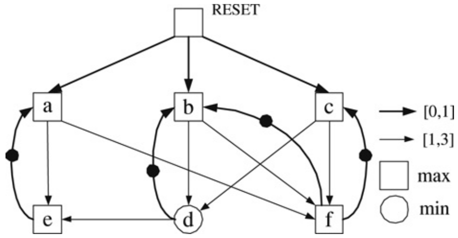

Fig. 4. A uniform system with height 2.

Table 1
Time separations for the systems in Fig. 4

i	1	2	3	4
Λ_i	$\begin{pmatrix}0 & \varepsilon & \varepsilon\\ \varepsilon & 0 & \varepsilon\\ \varepsilon & \varepsilon & 0\end{pmatrix}$	$\begin{pmatrix}0 & \varepsilon & \varepsilon\\ 3 & 0 & 1\\ 3 & 3 & 0\end{pmatrix}$	$\begin{pmatrix}0 & 6 & 6\\ 3 & 0 & 1\\ 3 & 3 & 0\end{pmatrix}$	$\begin{pmatrix}0 & 6 & 6\\ 3 & 0 & 1\\ 3 & 3 & 0\end{pmatrix}$

Below is an example illustrating Theorem 1.

Example 2. This example is inspired by an example in [11]. Its cyclic timing constraint graph is shown in Fig. 4. For this example, we have

$$\hat{F}(T_1) = \begin{pmatrix}\tau_{a_1} \vee (\tau_{b_1} \wedge \tau_{c_1})\\ \tau_{a_1} \vee \tau_{b_1} \vee \tau_{c_1}\\ \tau_{a_1} \vee \tau_{b_1} \vee \tau_{c_1}\end{pmatrix}, \quad \hat{F}^2(T_1) = \begin{pmatrix}\tau_{a_1} \vee \tau_{b_1} \vee \tau_{c_1}\\ \tau_{a_1} \vee \tau_{b_1} \vee \tau_{c_1}\\ \tau_{a_1} \vee \tau_{b_1} \vee \tau_{c_1}\end{pmatrix}.$$

It is clear from the expression [6] of $\hat{F}^2(T_1)$ that this is a uniform system with height $r = 2$ (and apparently not tightly coupled). In Table 1, Λ_i, $i = 1, 2, 3, 4$, are the exact time separations computed by the branch and bound method in [17] with $\Lambda_1 = \Theta$, where $\varepsilon \overset{\Delta}{=} +\infty$ and $\Theta = \begin{pmatrix}0 & \varepsilon & \varepsilon\\ \varepsilon & 0 & \varepsilon\\ \varepsilon & \varepsilon & 0\end{pmatrix}$. It can be verified that $\Lambda_i \leq \bar{\Lambda}_3 = \begin{pmatrix}0 & 6 & 6\\ 6 & 0 & 6\\ 6 & 6 & 0\end{pmatrix}$ for all $i \geq 3$, even if $\Lambda_1 = \Theta$ (the elements of Λ_i may not be finite for $i = 1, 2$).

In the following, we show that uniformity, as a structural condition, is also necessary for long-term time separation to be bounded.

Theorem 2. *If G is a non-uniform system, for every bounded Λ_1, there exists a set of finite edge delays such that Λ_k^* is not bounded for any finite k.*

Remark. With Theorems 1 and 2, we have established a necessary and sufficient structural boundedness condition for cyclic event rule systems with both min and max constraints. It should be noted that the long-term time separations of non-uniform systems can still be bounded with some other specific edge delays. All structural properties based on the skeleton of a cyclic timing constraint graph cannot be used to judge whether a specific non-uniform system is bounded or not because they cannot capture the information of edge delays. However, the structural boundedness condition can still be very valuable in real applications because edge delays cannot be easily determined *a priori*. The structural property about uniformity can be helpful in designing event rule systems with guarantee of finite time separations.

We shall prove Theorem 2 constructively. Before giving the formal proof, let us show the intuition behind it by considering the non-uniform system in Example 3.

Example 3. Consider the cyclic timing constraint graph as shown in Fig. 5. Fig. 6 is a portion of its unfolded graph.

State functions $\hat{F}^i(T_1)$, $i = 1, 2, 3, 4$, are listed in Table 2 with $T_1 = (\tau_{a_1}, \tau_{b_1}, \tau_{c_1})^{\mathrm{T}}$. It can be seen that

$$\hat{F}^2(\hat{F}^2) = \hat{F}^2. \tag{4}$$

This implies that

$$\hat{F}^{2k}(\hat{F}^2) = \hat{F}^2, \quad \text{for } k \geq 1. \tag{5}$$

[6] In general, rules such as $f = f \wedge f = f \vee f$, $f_1 \vee f_2 = f_2 \vee f_1$, $f_1 \wedge f_2 = f_2 \wedge f_1$, $(f_1 \vee f_2) \wedge f_3 = (f_1 \wedge f_3) \vee (f_2 \wedge f_3)$ and $(f_1 \wedge f_2) \vee f_3 = (f_1 \vee f_3) \wedge (f_2 \vee f_3)$ can be used to obtain concise expressions for the pure min–max function $F^k(T_i)$.

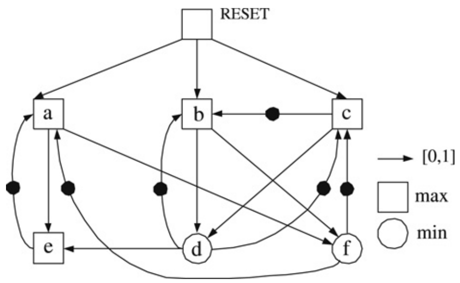

Fig. 5. A non-uniform system.

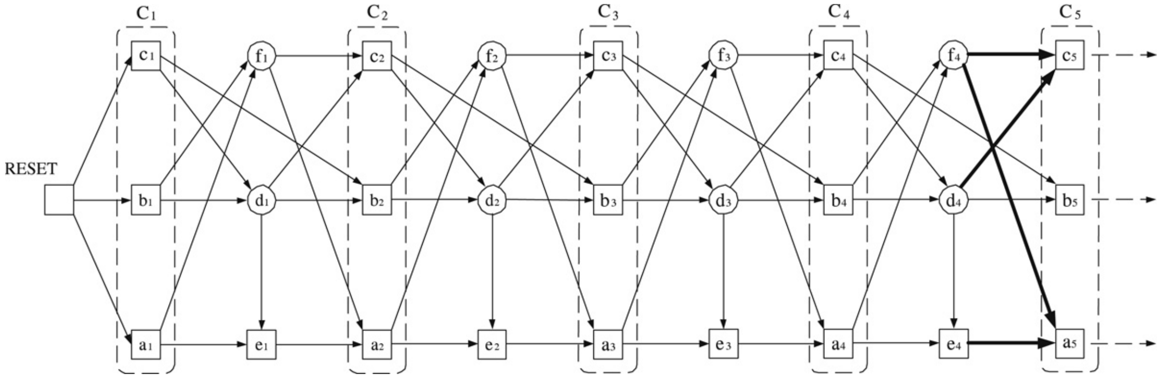

Fig. 6. Unfolded graph of the system in Fig. 5.

Table 2
State functions for the system in Fig. 5

i	1	2	3	4
$\hat{F}^i(T_1)$	$\begin{pmatrix}\tau_{a_1} \vee (\tau_{b_1} \wedge \tau_{c_1}) \\ \tau_{c_1} \\ (\tau_{a_1} \vee \tau_{c_1}) \wedge \tau_{b_1}\end{pmatrix}$	$\begin{pmatrix}\tau_{a_1} \vee (\tau_{b_1} \wedge \tau_{c_1}) \\ (\tau_{a_1} \vee \tau_{c_1}) \wedge \tau_{b_1} \\ (\tau_{a_1} \vee \tau_{b_1}) \wedge \tau_{c_1}\end{pmatrix}$	$\begin{pmatrix}\tau_{a_1} \vee (\tau_{b_1} \wedge \tau_{c_1}) \\ (\tau_{a_1} \vee \tau_{b_1}) \wedge \tau_{c_1} \\ (\tau_{a_1} \vee \tau_{c_1}) \wedge \tau_{b_1}\end{pmatrix}$	$\begin{pmatrix}\tau_{a_1} \vee (\tau_{b_1} \wedge \tau_{c_1}) \\ (\tau_{a_1} \vee \tau_{c_1}) \wedge \tau_{b_1} \\ (\tau_{a_1} \vee \tau_{b_1}) \wedge \tau_{c_1}\end{pmatrix}$

The system must be non-uniform because we can choose a vector T_1 such that $\hat{F}^{2k}(T_1)$ is always a non-constant vector. For instance, we can choose $T_1 = (2 \quad 1 \quad 2)^{\mathrm{T}}$. We have $\hat{F}^2(T_1) = (2 \quad 1 \quad 2)^{\mathrm{T}}$. Let $Y = \hat{F}^2(T_1) = (2 \quad 1 \quad 2)^{\mathrm{T}}$. Next is our construction leading to unbounded long-term time separations for some of the events in G. From Eq. (4), we have

$$\hat{F}^2(Y) = Y. \tag{6}$$

From Eq. (6) and direct calculation, we can verify that

$$\hat{F}^2(Y + i \cdot \xi) = Y + i \cdot \xi, \quad i = 1, 2, 3, \ldots \tag{7}$$

holds for the vector $\xi = (1 \quad 0 \quad 1)^{\mathrm{T}}$. In Fig. 6, choose 1 as the delay of the thick edges (edges pointing to vertices a_{2i+3} and c_{2i+3}, $i = 1, 2, 3, \ldots$ in G^*) and choose 0 as the delay of other edges. The above selection of edge delays in G^* leads to in G^*:

$$T_3 = \hat{F}^2(T_1), \quad T_{2i+3} = F^2(T_{2i+1}) = \hat{F}^2(T_{2i+1}) + \xi, \quad i \geq 1. \tag{8}$$

From Eqs. (7) and (8) and $T_1 = (2 \quad 1 \quad 2)^{\mathrm{T}}$, we have for all $i \geq 1$

$$T_{2i+3} = (2 + i \quad 1 \quad 2 + i)^{\mathrm{T}}. \tag{9}$$

From Eq. (9), $\Delta_{2i+3}(b_{2i+3}, a_{2i+3}) \geq \tau_{a_{2i+3}} - \tau_{b_{2i+3}} = 2 + i - 1 = 1 + i$ and $\Delta_{2i+3}(b_{2i+3}, c_{2i+3}) \geq \tau_{c_{2i+3}} - \tau_{b_{2i+3}} = 2 + i - 1 = 1 + i$. Thus, $\Delta_{2i+3}(b_{2i+3}, a_{2i+3})$ and $\Delta_{2i+3}(b_{2i+3}, c_{2i+3})$ must be infinite when i goes to $+\infty$, i.e., Λ_k^* is not bounded for all finite k.

3.1. Proof of Theorem 2

We need some lemmas. Denote S as the set of pure min–max functions $\mathbf{R}^n \to \mathbf{R}^n$.

Lemma 1. *S is a finite set.*

Proof. Due to the basic properties

$$f = f \vee f$$

$$f = f \wedge f$$

$$f_1 \vee f_2 = f_2 \vee f_1$$

$$f_1 \wedge f_2 = f_2 \wedge f_1$$

$$(f_1 \vee f_2) \wedge f_3 = (f_1 \wedge f_3) \vee (f_2 \wedge f_3)$$

$$(f_1 \wedge f_2) \vee f_3 = (f_1 \vee f_3) \wedge (f_2 \vee f_3)$$

one can show that all pure min–max functions can be expressed in the so-called CNF (conjunctive normal form)

$$f = \bigwedge_{i \in I} (f_i)$$

where $f_i, i \in I$ are all pure max functions of form $f_i = \bigvee_{j \in J_i} x_j$ and $J_i \subseteq \{1, \dots, n\}$. Since the total number of different subsets of $\{1, \dots, n\}$ is finite, there are only a finite number of pure max functions. As a result, the total number of different CNFs of pure min–max functions is also finite. Thus, the size of S is finite. □

Lemma 2. *For a pure min–max function $\hat{F}$: $\mathbf{R}^n \to \mathbf{R}^n$, there must always exist $k_0 \geq 1$ and $d \geq 1$ such that $\hat{F}^{k_0} = \hat{F}^d(\hat{F}^{k_0})$.*

Proof. Obviously, $\hat{F}^i \in S$ for all $i \geq 1$. Since S is finite, there must exist j such that $\hat{F}^i = \hat{F}^j$ where $j > i$. Let $k_0 = i$ and $d = j - i$. Then $\hat{F}^{k_0} = \hat{F}^{k_0+d}$, i.e., $\hat{F}^{k_0} = \hat{F}^d(\hat{F}^{k_0})$. □

Obviously, a uniform system G with state function F is a system with $k_0 = r$ and $d = 1$, where r is its height. A tightly coupled system is a system with $k_0 = 1$ and $d = 1$.

Lemma 3. *For pure min–max function $\hat{H}$: $\mathbf{R}^n \to \mathbf{R}^n$, assume that there exists a non-constant real vector $Y = (y_1, y_2, \dots, y_n)^{\mathrm{T}}$ (i.e., $y_1, y_2, \dots, y_n$ are not all the same) such that $Y = \hat{H}(Y)$. Without loss of generality, assume $y_1 \geq y_2 \geq \cdots y_p > y_{p+1} \geq \cdots \geq y_n$ for some $1 \leq p < n$. If $Y' = (y'_1, y'_2, \dots, y'_n)^{\mathrm{T}}$ is a vector satisfying $y'_1 \geq y'_2 \geq \cdots y'_p > y'_{p+1} \geq \cdots \geq y'_n$ and all inequality signs are exactly the same[7] as those which appeared in the elements of Y, then $Y' = \hat{H}(Y')$.*

Proof. The conclusion follows from the order preserving property of both max and min operations. That is, the $\max(y_1, y_2)$ ($\min(y_1, y_2)$) always picks the larger (smaller) one of the two numbers y_1 and y_2. The choice does not depend on what the values of y_1 and y_2 are. In general, every pure min–max function $f(y_1, y_2, \dots, y_n)$ can be viewed as a selection process which will pick from $y_1, y_2, \dots, y_n$ a number of specific ranks. For two sets of numbers $\{y_1, y_2, \dots, y_n\}$ and $\{y'_1, y'_2, \dots, y'_n\}$, if for every i, y_i and y'_i have the same rank in the set $\{y_1, y_2, \dots, y_n\}$ and the set $\{y'_1, y'_2, \dots, y'_n\}$ respectively, then a pure min–max function $f(y_1, y_2, \dots, y_n)$ will always return the value of the same location in both sets, that is, $f(y_1, y_2, \dots, y_n) = y_i$ if and only if $f(y'_1, y'_2, \dots, y'_n) = y'_i$. By applying this fact to all elements of the function $\hat{H}$, we can establish the desired conclusion. □

Lemma 4. *Let $\hat{F}$ be a pure min–max function. For an integer $k \geq 1$, if there is an $X \in \mathbf{R}^n$ such that $\hat{F}^k(X)$ is a non-constant vector (i.e., $\hat{F}^k_1(X), \dots, \hat{F}^k_n(X)$ are not all the same.), then there must be a Boolean vector $Z \in \mathbf{B}^n$ such that $\hat{F}^k(Z)$ is a non-constant vector.*

Proof. Let $Y = \hat{F}^k(X)$. Without loss of generality, assume that $y_1 \leq \cdots \leq y_n$. Since Y is a non-constant vector, there must be an index $i_0 \geq 2$ such that $y_1 = \cdots = y_{i_0-1} < y_{i_0} \leq \cdots \leq y_n$. Since $\hat{F}^k$ is also a pure min–max function, it is clear that $y_1 \in \{x_1, \dots, x_n\}$. Let p, q be two indices such that $x_p = y_1$ and $x_q = y_{i_0}$. So, $x_p < x_q$. Define a Boolean vector Z such that $z_i = 0$ if $x_i \leq x_p = y_1$ and $z_i = 1$ if $x_i > x_p$ for $i = 1, \dots, n$. Based on Z, we define a real vector X^Z whose elements only take values from two distinct values $\{x_p, x_q\}$ as follows: $x^Z_i = x_p$ if $z_i = 0$ and $x^Z_i = x_q$ if $z_i = 1$. Note $\hat{F}^k(X^Z)$ is also a vector whose elements only take values from the set $\{x_p, x_q\}$. Furthermore, Z can be obtained from X^Z by subtracting the value x_p from all elements of X^Z and then dividing the resulting vector by the value $x_q - x_p$, i.e.,

$$Z = \left(\frac{x^Z_1 - x_p}{x_q - x_p}, \dots, \frac{x^Z_n - x_p}{x_q - x_p} \right). \tag{10}$$

It is easy to show that for a pure min–max function f of n variables,

(1) $f(x_1 - h, \dots, x_n - h) = f(x_1, \dots, x_n) - h$ holds for $h \in \mathbf{R}$.

---

[7] That is, when $y_i \geq y_{i+1}$ is true, we ask $y'_i > y'_{i+1}$ if $y_i > y_{i+1}$; or $y'_i = y'_{i+1}$ if $y_i = y_{i+1}$.

(2) $f(\frac{x_1}{d},\ldots,\frac{x_n}{d}) = \frac{f(x_1,\ldots,x_n)}{d}$ holds for $d > 0$.
(3) $f(x_1,\ldots,x_i\ldots,x_n) > a$ and $x_i \le a$ imply that
$f(x_1,\ldots,x_i\ldots,x_n) = f(x_1,\ldots,a,\ldots,x_n)$.
(4) $f(x_1,\ldots,x_i\ldots,x_n) > a$ and $x_i > a$ imply that
$f(x_1,\ldots,b,\ldots,x_n) > a, \forall b \in (a,+\infty)$.

By applying (1) and (2) to $\hat{F}_i^k(X^Z)$ sequentially with $h = x_p$ and $d = x_q - x_p$ for $i = 1,\ldots,n$, we obtain from Eq. (10) that $\frac{\hat{F}_i^k(X^Z)-x_p}{x_q-x_p} = \hat{F}_i^k(Z)$ holds for the Boolean vector Z. For $\hat{F}_{i_0}^k(X)$ and $a = x_p < y_{i_0}$, by applying (3) for all i s.t. $x_i \le x_p$ and (4) for all i s.t. $x_i > x_p$, we have $\hat{F}_{i_0}^k(X^Z) > x_p$. This implies that $\hat{F}_{i_0}^k(X^Z) = x_q = y_{i_0}$. Dually, we can show that $\hat{F}_1^k(X^Z) = x_p = y_1$. Based on these facts, we can establish for X^Z that $\hat{F}_{i_0}^k(X^Z) > \hat{F}_1^k(X^Z)$. This implies that $\frac{\hat{F}_{i_0}^k(X^Z)-x_p}{x_q-x_p} > \frac{\hat{F}_1^k(X^Z)-x_p}{x_q-x_p}$. As a result, $\hat{F}_{i_0}^k(Z) = 1$ and $\hat{F}_1^k(Z) = 0$. □

Proof of Theorem 2. We prove the theorem constructively. We pick [0, 1] as the delay of all edges in the non-uniform system G. From Lemma 2, there must exist some k_0 and d such that for all $T_1 \in \mathbf{R}^n$

$$\hat{F}^d(\hat{F}^{k_0}(T_1)) = \hat{F}^{k_0}(T_1). \tag{11}$$

For any given $T_1 \in \mathbf{R}^n$, based on whether $\hat{F}^{k_0}(T_1)$ is a non-constant vector, we divide the proof into these two cases.

Case 1. $\hat{F}^{k_0}(T_1)$ is a non-constant vector.

Let $Y = \hat{F}^{k_0}(T_1)$ and $\mathcal{A} = \{u|Y_u \ge Y_v, \forall v \in C_1\}$. We can rewrite Eq. (11) as

$$\hat{F}^d(Y) = Y. \tag{12}$$

We choose $\delta_{k,u_{i\cdot d+k_0+1}} = 1$ for all $u \in \mathcal{A}$, $k \in \text{preds}(u_{i\cdot d+k_0+1})$ and $i \ge 1$. Then pick 0 as the delays for all other edges in G^*. We have

$$T_{k_0+1} = \hat{F}^{k_0}(T_1); \quad T_{i\cdot d+k_0+1} = \hat{F}^d(T_{(i-1)d+k_0+1}) + \xi, \quad i \ge 1. \tag{13}$$

where $\xi \in \mathbf{R}^n$ and $\xi_v = \begin{cases} 1 & v \in \mathcal{A} \\ 0 & v \notin \mathcal{A} \end{cases}$. By applying Lemma 3 to the case in which $\hat{H} = \hat{F}^d$ and $Y' = Y + i\xi$, we have from Eq. (12) that

$$Y + i\xi = \hat{F}^d(Y + i\xi). \tag{14}$$

That is,

$$\hat{F}^{k_0}(T_1) + i\xi = \hat{F}^d(\hat{F}^{k_0}(T_1) + i\xi), \quad i \ge 1. \tag{15}$$

From Eqs. (13) and (15), we have

$$T_{i\cdot d+k_0+1} = \hat{F}^{k_0}(T_1) + i\xi, \quad i \ge 0. \tag{16}$$

From Eq. (16), we obtain

$$\Lambda_{i\cdot d+k_0+1}(v_{i\cdot d+k_0+1}, u_{i\cdot d+k_0+1}) \ge \tau_{u_{i\cdot d+k_0+1}} - \tau_{v_{i\cdot d+k_0+1}} > i, \quad \forall u \in \mathcal{A}, v \notin \mathcal{A}.$$

Thus, $\Lambda_{i\cdot d+k_0+1}(v_{i\cdot d+k_0+1}, u_{i\cdot d+k_0+1})$ must tend to $+\infty$ $\forall u \in \mathcal{A}, v \notin \mathcal{A}$ when i approaches $+\infty$.

Case 2. $\hat{F}^{k_0}(T_1)$ is a constant real vector.

Since G is non-uniform, there must exist some $X \in \mathbf{R}^n$ such that $\hat{F}^{k_0}(X)$ is a non-constant vector (otherwise G will be a uniform system with height k_0). Without loss of generality, we assume that X is a Boolean vector (recall Lemma 4). Let $Y = \hat{F}^{k_0}(X)$ and $\mathcal{A} = \{u|Y_u \ge Y_v, \forall v \in C_1\}$. Notice that Y is also a Boolean vector. By applying Lemma 2 for X, we have

$$Y = \hat{F}^d(Y). \tag{17}$$

We choose $\delta_{k,u_{i\cdot d+k_0+1}} = 1$ for all $u \in \mathcal{A}$, $k \in \text{preds}(u_{i\cdot d+k_0+1})$ and $i \ge 0$. Then pick 0 as the delays for all other edges in G^*. We have

$$T_{k_0+1} = \hat{F}^{k_0}(T_1) + \xi; \quad T_{i\cdot d+k_0+1} = \hat{F}^d(T_{(i-1)d+k_0+1}) + \xi, \quad i \ge 1. \tag{18}$$

where $\xi \in \mathbf{R}^n$ and $\xi_v = \begin{cases} 1 & v \in \mathcal{A} \\ 0 & v \notin \mathcal{A} \end{cases}$. By applying Lemma 3 to the case in which $Y' = \hat{F}^{k_0}(T_1) + i\xi$ and $\hat{H} = \hat{F}^d$, we obtain from Eq. (17) that

$$\hat{F}^{k_0}(T_1) + i\xi = \hat{F}^d(\hat{F}^{k_0}(T_1) + i\xi), \quad i \ge 1. \tag{19}$$

Note the condition of Lemma 3 is satisfied since $\hat{F}^{k_0}(T_1)$ is a constant vector and $\xi = Y$. From Eqs. (18) and (19), we have

$$T_{i\cdot d+k_0+1} = \hat{F}^{k_0}(T_1) + (i+1)\xi, \quad i \ge 0. \tag{20}$$

From Eq. (20), we obtain

$$\Lambda_{i\cdot d+k_0+1}(v_{i\cdot d+k_0+1}, u_{i\cdot d+k_0+1}) \ge \tau_{u_{i\cdot d+k_0+1}} - \tau_{v_{i\cdot d+k_0+1}} = i+1, \quad \forall u \in \mathcal{A}, v \notin \mathcal{A}.$$

Thus, $\Lambda_{i\cdot d+k_0+1}(v_{i\cdot d+k_0+1}, u_{i\cdot d+k_0+1})$ must tend to $+\infty$ $\forall u \in \mathcal{A}, v \notin \mathcal{A}$ when i approaches $+\infty$.

From both cases above, we can conclude that Λ_k^* is not always bounded for all finite k and for all possible bounded Λ_1. □

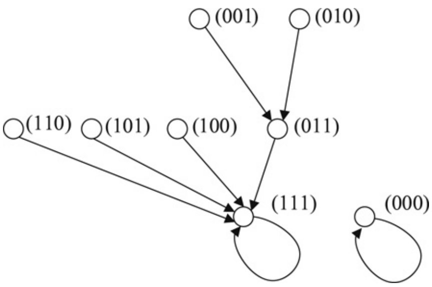

Fig. 7. Illustration of state transition graph of the Boolean function for Example 2.

4. Test of uniformity

Since the structural property of uniformity can help design circuits with guaranteed finite time separations, its test problem becomes valuable. As we did for the examples, direct test of uniformity is to manually derive and compare the elements of the pure min–max function sequence $F^k(T_1)$. This is somewhat *ad hoc* in nature. Some systematic way is needed.

In this section, we will reduce the problem to a test on monotone Boolean functions. For a pure min–max function $\hat{F} : \mathbf{R}^n \to \mathbf{R}^n$, we can always obtain a monotone Boolean function $\hat{F}_B : \mathbf{B}^n \to \mathbf{B}^n$ by replacing the operation 'min' with 'AND' and replacing the operation 'max' with 'OR'. For example, the monotone Boolean function $\hat{F}_B$ corresponding to the pure min–max function $\hat{F}$ in Example 2 is

$$\hat{F}_B(Z) = \begin{pmatrix} z_a + (z_b \cdot z_c) \\ z_a + z_b + z_c \\ z_a + z_b + z_c \end{pmatrix}. \tag{21}$$

We define uniformity for a Boolean function $\hat{F}_B : \mathbf{B}^n \to \mathbf{B}^n$ in the similar way as for a pure min–max function: $\hat{F}_B$ is uniform if there is a non-negative integer $r \geq 1$ such that for all Boolean vectors Z, $(\hat{F}_B)^r_i(Z) = (\hat{F}_B)^r_j(Z)$, $\forall i, j = 1, \ldots, n$.

Proposition 2. *A cyclic event rule system given by a pure min–max function $\hat{F}$ is uniform if and only if $\hat{F}_B$ is a uniform Boolean function.*

Proof. The ONLY IF part is obvious since $\mathbf{B}^n \subset \mathbf{R}^n$ and $(\hat{F}_B)^r(Z) = (\hat{F})^r(Z)$ on $\mathbf{B}^n$, as a result for all $Z \in \mathbf{B}^n$, it holds that $(\hat{F}_B)^r_i(Z) = \hat{F}^r_i(Z) = \hat{F}^r_j(Z) = (\hat{F}_B)^r_j(Z)$. Let us turn to the IF part. We prove by contradiction. Suppose $\hat{F}_B$ is uniform with height r, but there is an $X \in \mathbf{R}^n$ such that $\hat{F}^r(X)$ is a non-constant vector. According to Lemma 4, there must be a Boolean vector Z such that $\hat{F}^r(Z)$ is also a non-constant vector. Note $\hat{F}^r(Z) = (\hat{F}_B)^r(Z)$. This is in contradiction with the assumption that $\hat{F}_B$ is uniform with height r. □

For small scale system, Proposition 2 allows us to use the iteration graph [22] of the Boolean function $\hat{F}_B$ to determine the uniformity of $\hat{F}$. The iteration graph is a graph with 2^n nodes. The nodes are labeled by n-dimensional Boolean vectors. There is an edge from node Z to Z' if and only if $Z' = \hat{F}_B(Z)$. See [22] for discussions on the properties of iteration graphs of general discrete iterations on finite set. Cycles are called attractors. The special cycles with length 1 are called fixed points. For monotone Boolean functions $\hat{F}_B$, the two points $(1, \ldots, 1)^T$ and $(0, \ldots, 0)^T$ are always fixed points. The state transition graph of a uniform Boolean function has the feature that $(1, \ldots, 1)^T$ and $(0, \ldots, 0)^T$ are the only two *attractors*.[8] For example, the state transition graph of the Boolean function for Example 2 in Eq. (21) is shown in Fig. 7. From the state transition graphs, one can also decide the height of a uniform function, that is, the length of the longest path from any state to one of the two fixed points. From Fig. 7 we can see that $r = 2$. One longest path is from (010) to (011) to (111).

Although from the above analysis, we know uniformity can always be decided in finite time, it still remains open whether there exists a polynomial algorithm to check uniformity. For larger systems, we suggest using symbolic model checkers (e.g., [7]) on $\hat{F}_B$ to handle the uniformity checking problem. The idea is to test whether for all Boolean vectors Z, the Boolean vector sequence $(\hat{F}_B)^k(Z)$, $k = 1, 2, \ldots$ finally contains either $(1, \ldots, 1)^T$ or $(0, \ldots, 0)^T$. Using a symbolic model checker, once we find $\hat{F}_B$ is uniform, we can further search for the height r of $\hat{F}_B$ by applying the model checkers again to judge whether all components of $(\hat{F}_B)^r$ are the same on $\mathbf{B}^n$.

Next, we present a sufficient condition for uniformity in Proposition 3.

Proposition 3. *If there exists a uniform cutset $G'_i \subseteq G_i$ (see Fig. 8) with height 1 in the cyclic component G_i, the cyclic event rule system as a whole is uniform.*

8 Attractors include both limit cycles and fix points.

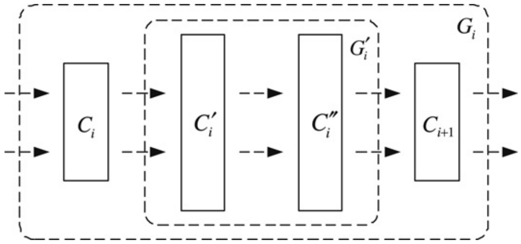

Fig. 8. Illustration of Proposition 3.

Proof. Without loss of generality, assume there exist two cutsets $C_i', C_i'' \subseteq G_i$ such that $C_i'' \subseteq R(C_i')$ and $G_i' = R(C_i') \bigcap (R'(C_i'') \bigcup C_i'')$, see Fig. 8. Define T_i' and T_i'' as the state vectors of C_i' and C_i'' respectively. Denote $\hat{F}^*$ as the function from T_i' to T_i''. The proof can be divided into the three cases below:

- $C_i' = C_i$ and $C_i'' \neq C_{i+1}$,
- $C_i' \neq C_i$ and $C_i'' = C_{i+1}$,
- $C_i' \neq C_i$ and $C_i'' \neq C_{i+1}$.

These three cases can be proved in a similar way. We prove the third case as an example. Denote $\hat{F}'$ as the function from T_i to T_i' and $\hat{F}''$ the function from T_i'' to T_{i+1}. Since G_i' is uniform with height 1, $\hat{F}_u^* = \hat{F}_v^*, \forall u, v \in C_i''$. Let $\hat{F}_u^* = \hat{g}, \forall u \in C_i''$, where $\hat{g}$ is a pure min–max function. Again from $\hat{F} = \hat{F}''(\hat{F}^*(\hat{F}'))$, it must hold that $\hat{F}_v = \hat{g}(\hat{F}'), \forall v \in C_i$. Thus, the system is uniform with height 1. □

Remark. Proposition 3 has an interesting implication for designing uniform systems. If we have an existing piece of design which is known to be uniform with height 1, say, tightly coupled, we can generate a larger uniform system in which the events in this piece form a cutset.

Remark. Proposition 3 is only a sufficient condition. When there is no subset uniform, the whole system may still be uniform.

5. Adapting CAS algorithm to find bounds on time separations

In this section, we study problem **C2** for which the CAS algorithm in [4] has been shown to give finite bounds on time separations on tightly coupled systems. Since tightly coupled systems are special uniform systems, it is natural to ask whether CAS can be used directly on uniform systems with provable bounded estimation. It turns out not to be the case. In this section, we first present one example of uniform systems where CAS fails to give finite bounds. The example is shown as in Fig. 9 (showing the portion up to the cutset C_2 of the unfolded graph G^*). We assume that all delays on all edges in the repeated part G_1 of the system are 0. Since we focus on the long-term behavior, as in [4], we leave initial behavior unspecified, i.e., we assume that the delays in G_0 are arbitrary. It can be directly verified that

$$\hat{F}^i(T_1) = (g(T_1), g(T_1), g(T_1))^{\mathrm{T}}, \quad i = 1, 2, \ldots,$$

where $g(T_1) = (\tau_{a_1} \wedge \tau_{b_1}) \vee (\tau_{a_1} \wedge \tau_{c_1}) \vee (\tau_{b_1} \wedge \tau_{c_1})$. So, the system is a uniform system with height $r = 1$. The long-term time separations are all finite and in fact are all 0. Meanwhile, the result of direct calculation by CAS on this system is

$$\Lambda_i' = \begin{pmatrix} 0 & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 \end{pmatrix}, \quad \text{for } i = 1, 2, \ldots. \tag{22}$$

We can see that all time separation bounds are infinite.

For a uniform system of height r, to make CAS produce finite bounds, we slightly change the way of computing bounds for the cutset C_i. To be specific, we do the following: First compute $\bar{\Lambda}_{r+1}$. Then for all $i \geq r$, the new bounds for the cutset C_i is set as the tighter one of (1) the bounds computed by CAS and (2) $\bar{\Lambda}_{r+1}$. Then we run CAS as usual except that whenever the bounds of C_i, $i \geq r$ are used, we use the new bounds. Here we use the fact that CAS always gives upper bounds for the subgraph G_{i+1} whenever some upper bounds on the cutset C_i are used. In this way, we can make sure finite bounds are obtained since Theorem 1 establishes that for all $i \geq r$, Λ_i is bounded above by Λ_{r+1} which is finite.

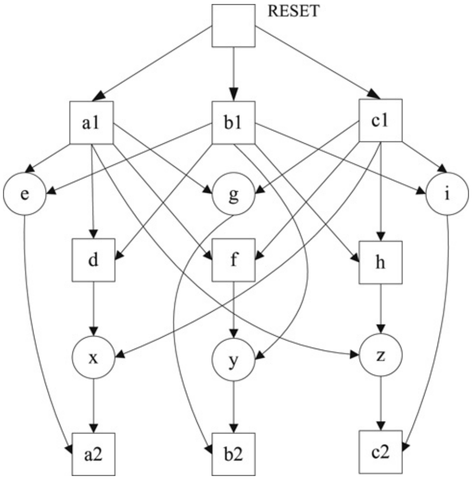

Fig. 9. A uniform system on which CAS algorithm fails to give finite bounds on time separations.

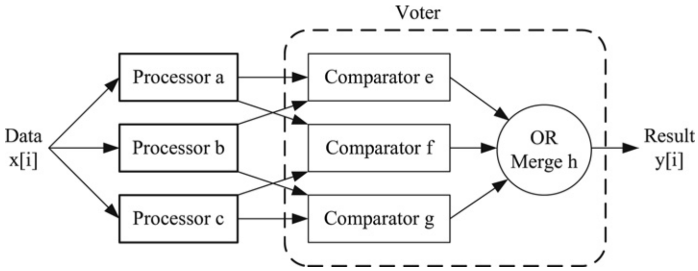

Fig. 10. A redundant computing scheme in TMR structure.

6. Discussions on applications

The structural property of uniformity has the following two implications when used in practice. On the one hand, it can be used to confirm the boundedness of long-term time separations of a general class of cyclic event rule systems. Example 1 is such an example. It is uniform but not tightly coupled. As a more practical example, we would like to analyze an iterative computing system in Triple Modular Redundancy (TMR) structure [23] as shown in Fig. 10. For a given sequence of data $x[i]$, $i = 1, 2, \ldots,$ the three processors a, b and c run identical operation on $x[i]$ and work in parallel. The purpose of using three processors is to combat the random failure in hardware by taking advantage of redundancy. The time spent for handling $x[i]$ on a, b and c are $\delta_{a_i} \in [d_a, D_a]$, $\delta_{b_i} \in [d_b, D_b]$ and $\delta_{c_i} \in [d_c, D_c]$ respectively. The outputs of the three processors are fed into three comparators e, f and g. As long as both inputs of a comparator are ready, it will compare whether the two input values are the same. If they are the same, the value will be sent by the comparator as a valid output to the OR merge h. The cycle of redundant computing for $x[i]$ is finished once h receives one valid input. That input is used to produce the output $y[i]$ of the cycle. A new cycle then starts by loading every processor with $x[i+1]$.[9]

Here, we focus on the timing evolution of an ideal case in which there is no random failure at all processors. The question is whether the system has bounded long-term time separation. To analyze the system, we construct a cyclic timing constraint graph G for the system as in Fig. 11 and write down the timing constraints as follows. Note delays are not marked in the figure.

$$\begin{aligned}
\tau_{a_i} &= \tau_{h_{i-1}} + \delta_{a_i} \\
\tau_{b_i} &= \tau_{h_{i-1}} + \delta_{b_i} \\
\tau_{c_i} &= \tau_{h_{i-1}} + \delta_{c_i} \\
\tau_{e_i} &= (\tau_{a_i} \vee \tau_{b_i}) + \delta_{d_i} \\
\tau_{f_i} &= (\tau_{a_i} \vee \tau_{c_i}) + \delta_{e_i} \\
\tau_{g_i} &= (\tau_{b_i} \vee \tau_{c_i}) + \delta_{f_i} \\
\tau_{h_i} &= (\tau_{e_i} \wedge \tau_{f_i} \wedge \tau_{g_i}) + \delta_{h_i}.
\end{aligned}$$

9 When a new cycle for $x[i+1]$ is started, the processor running unfinished task for $x[i]$ will cancel it and will load the new data $x[i+1]$.

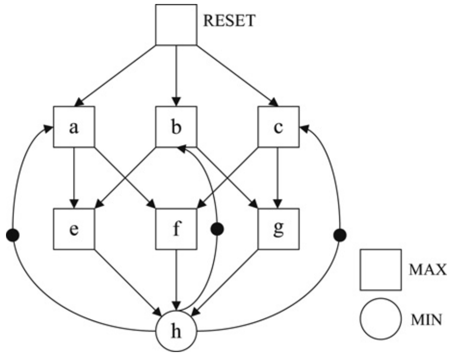

Fig. 11. Cyclic timing constraint graph for the system in Fig. 10.

Taking $T_i = (\tau_{a_i}, \tau_{b_i}, \tau_{c_i})^{\mathrm{T}}$, $i = 1, 2, \ldots$, as the state vectors, we can obtain the state function of the skeleton $\hat{G}$

$$\hat{F}(T_i) = \begin{pmatrix} (\tau_{a_i} \vee \tau_{b_i}) \wedge (\tau_{a_i} \vee \tau_{c_i}) \wedge (\tau_{b_i} \vee \tau_{c_i}) \\ (\tau_{a_i} \vee \tau_{b_i}) \wedge (\tau_{a_i} \vee \tau_{c_i}) \wedge (\tau_{b_i} \vee \tau_{c_i}) \\ (\tau_{a_i} \vee \tau_{b_i}) \wedge (\tau_{a_i} \vee \tau_{c_i}) \wedge (\tau_{b_i} \vee \tau_{c_i}) \end{pmatrix}. \quad (23)$$

It is straightforward to see that the system is uniform with height 1. At the same time, it is also evident that the system is not tightly coupled since every path from a_i to a_{i+1} contains both minimum and maximum nodes in $\hat{G}$ (also true for nodes b and c). As an application of Theorem 1, we can conclude that the system has finite long-term time separation.

On the other hand, it could also play some role in the design of asynchronous systems. For example, Proposition 3 has an interesting implication for designing uniform systems. If we have an existing piece of design which is known to be uniform with height 1, say, tightly coupled, we can generate a larger uniform system in which the events in this piece form a cutset.

At last, we would like to point out, although there is no choice in the cyclic timing constraint graphs we considered in this paper, the structural condition of uniformity for bounding long-term time separation for events has the potential to cover more general cases. An easy extension is the case that choice is made among several alternations sharing a common skeleton. Take the two alternation case as example. Let G and G' be two cyclic timing constraint graphs whose skeletons $\hat{G}$ and $\hat{G}'$ are the same. We combine them into one system with choice such that the new state equation is given in the form of

$$T_{i+1} = F(T_i) \text{ or } F'(T_i),$$

where the decision on whether F or F' is chosen is made by some outside control mechanism. For this case, we can still define uniformity on $\hat{F}$ $(= \hat{F}')$ only. Since the proofs of Theorems 1 and 2 are based on skeletons, these two theorems still hold for the new class of systems with choice.

7. Conclusions

In this paper, our main contribution is the establishment of the sufficient and necessary structural condition for time separations for cyclic event rule systems to be bounded, namely uniformity. This result is obtained by exploring the algebraic structures—the skeletons of the cyclic timing constraint graphs of the systems. There are many interesting open problems, for example, the general structural boundedness condition for systems with choice, how to test uniformity more efficiently and sufficient conditions for uniformity with height 2 or more.

Acknowledgements

This work was supported in part by NSFC (Grant Nos 60274011, 60574067 and 60721003), National Key Project of China, Fundamental Research Funds from Tsinghua University and Chinese Scholarship Council, Ministry of Eduction of China, and the NCET program (NCET2004-04-0094) of China. The authors would like to thank the anonymous reviewers for their many constructive comments and Prof. Da-Zhong Zheng for his encouragement in the study of min–max systems. The first author would like to thank Prof. Bruce Krogh for his hospitality during the first author's visit to the ECE department of Carnegie Mellon University as a visiting scholar.

References

[1] F. Baccelli, G. Cohen, G.J. Olsder, J.-P. Quadrat, Synchronization and Linearity, John Wiley & Sons, Inc., 1992.
[2] S.M. Burns, Performance analysis and optimization of asynchronous circuits, Ph.D. Thesis, California Inst. Technol., Pasadena, 1991.
[3] Supratik Chakraborty, Polynomial-time techniques for approximate timing analysis of asynchronous systems, Ph.D. Thesis, Stanford Univ., Stanford, CA, 1998.

[4] Supratik Chakraborty, Kenneth Y. Yun, David L. Dill, Timing analysis of asynchronous systems using time separation of events, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 18 (8) (1999) 1061–1076.
[5] Supratik Chakraborty, David. L. Dill, Kenneth. Y. Yun, Efficient algorithms for approximate time separation of events, in Sadhana: Academy Proceedings in Engineering Sciences, The Indian Academy of Sciences 27 (2) (2002) 129–162.
[6] Yiping Cheng, Da-Zhong Zheng, Ultimate periodicity of orbits for min–max systems, IEEE Transactions on Automatic Control 47 (2002) 1937–1940.
[7] E.M. Clarke, O. Grumberg Jr., D.A. Peled, Model Checking, MIT Press, Cambridge, MA, 1999.
[8] J. Cochet-Terrasson, S. Gaubert, J. Gunawardena, A constructive fixed point theorem for min–max functions, Dynamics and Stability of Systems 14 (1999) 407–433.
[9] G. Cohen, P. Moller, J.-P. Quadrat, M. Viot, Algebraic tools for the performance evaluation of discrete event systems, Proceedings of IEEE 77 (1) (1989) 39–58.
[10] S. Gaubert, J. Gunawardena, The duality theorem for min–max functions, Comptes Rendus Académie des Sciences 326 (1998) 43–48.
[11] S. Gaubert, J. Gunawardena, Existence of eigenvectors for monotone homogeneous functions, Technical Report HPL-BRIMS-99-08, HP Lab, 1999.
[12] J. Gunawardena, Timing analysis of digital circuits and the theory of min–max functions, in: Digest of Technical Papers of the ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems, ACM, New York, 1993.
[13] J. Gunawardena, Min–max functions, J. DEDS 4 (1994) 377–406.
[14] H. Hulgaard, T. Amon, Symbolic timing analysis of asynchronous systems, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 19 (10) (2000) 1093–1104.
[15] H. Hulgaard, S.M. Burns, Bounded delay timing analysis of a class of csp programs, Formal Methods in System Design 11 (1997) 265–294.
[16] H. Hulgaard, S.M. Burns, T. Amon, G. Borriello, An algorithm for exact bounds on the time separation of events in concurrent systems, IEEE Transactions on Computers 44 (11) (1995) 1306–1317.
[17] K.L. McMillan, D.L. Dill, Algorithm for interface timing verification, in: Proc. IEEE Int. Conf. Computer Design: VLSI in Computers and Processors, 1992, pp. 48–51.
[18] C.J. Myers, Asynchronous Circuit Design, John Wiley & Sons, Inc., 2001.
[19] C.J. Myers, T.H.-Y. Meng, Synthesis of timed asynchronous circuits, IEEE Transactions VLSI Systems 1 (2) (1993) 106–119.
[20] G.J. Olsder, Eigenvalues of dynamic max-min systems, J. DEDS 1 (1991) 177–207.
[21] G.J. Olsder, On structural properties of min–max systems on Technical Report TWI Report 93-95, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1993.
[22] F. Robert, Discrete Iterations-A Metric Study, Springer-Verlag, New York, 1986.
[23] Kang G. Shin, Hagbae Kim, A time redundancy approach to TMR failures using fault-state likelihoods, IEEE Transactions on Computers 43 (10) (1994) 1151–1162.
[24] Yuegang Tao, Wende Chen, Cycle time assignment of min–max systems, International Journal of Control 76 (2003) 1790–1799.
[25] J.W. van der Woude, Subiono, Conditions for the structural existence of an eigenvalue of a bipartite (min, max, $+$)-system, Theoretical Computer Science 293 (1) (2003) 13–24.
[26] E.A. Walkup, Optimization of linear max-plus systems with application to timing analysis, Ph.D. Thesis, Univ. Washington, Seattle, 1995.
[27] T.-Y. Yen, A. Ishii, A. Casavant, W. Wolf, Efficient algorithms for interface timing verification, Formal Methods in System Design 12 (3) (1998) 241–265.
[28] Qianchuan Zhao, Da-Zhong Zheng, Xinping Zhu, Structure properties of min–max systems and existence of global cycle time, IEEE Transactions on Automatic Control 46 (1) (2001) 148–151.