Discrete Event Dynamic Systems manuscript No.
(will be inserted by the editor)

New representations for (max,+) automata with applications to performance evaluation and control of discrete event systems

Rabah Boukra · Sébastien Lahaye · Jean-Louis Boimond

Received: date / Accepted: date

Keywords Discrete event systems, (max,+) automata, performance evaluation, control

Abstract A large class of timed discrete event systems can be modeled by means of (max,+) automata, that is automata with weights in the so-called (max,+) algebra. In this contribution, specific recursive equations over (max,+) and (min,+) algebras are shown to be suitable for describing extremal behaviors of (max,+) automata. Several pertinent performance indicators can be easily derived or approximated from these representations with a low computation complexity. It is also shown how to define inputs which model exogenous influences on their dynamic evolution, and a new approach for the control of (max,+) automata is proposed.

1 Introduction

Several formalisms have been introduced and experienced for studying Discrete Event Systems (DES). Within the theory initiated by Ramadge and Wonham [26], events correspond to letters and DES are modeled by finite state machines. Most of the results pertain to logical behavior of DES (e.g. to restrict their behavior such that some forbidden states cannot be reached). On the contrary, the approach based on (max,+) algebra focuses on timed behavior of DES (e.g. to find out cycle times, earliest and/or latest dates of sequences-completion) [2]. While automata model naturally nondeterminisms inherent in conflicts or choices (e.g. to capture several possible schedules), (max,+) systems rather fit DES with deterministic behavior (by fixing the schedules in this case).

Stéphane Gaubert has first shown in [10] that automata with multiplicities in (max,+) algebra (also called (max,+) automata) combine these two approaches: concepts on automata can be used with results from (max,+) alge-

LUNAM Université, LISA, Angers, France E-mail: [rabah.boukra, sebastien.lahaye, jean-louis.boimond]@univ-angers.fr

bra to study at the same time logical and timing aspects of DES. In particular, (max,+) automata have been applied to performance evaluation [10, 12, 28], scheduling [29, 14] and control problems [17, 3, 27][1]. The behavior of (max,+) automata is then represented:

- either by recurrent equations over letters which can be seen as a generalization of the state representation with daters for (max,+) linear systems (see Eq. (5) below),
- and/or by formal power series with coefficients in (max,+) algebra which play an equivalent role to languages for logical (boolean) automata (see Eq. (7) below).

Unfortunately, many important problems turn out to be difficult, or even undecidable for general (max,+) automata with these representations. In particular, let us mention that:

- equality and inequality of formal power series representing (max,+) automata, is undecidable [19];
- the pseudo-inverse (i.e. *residuation*) of the product of rational formal power series is rational only for restrictive assumptions [3]. As a by-product, to expect the realizability of controllers, the supervisory control approach in [17] has to be restricted to small subclasses of (max,+) automata.

The present contribution proposes alternative representations for (max,+) automata. These representations describe the behavior of automata less accurately, since only their *extremal* behaviors are depicted. However, it is hoped that these representations make it possible to deal with problems of significant interest with reasonable complexity.

More precisely, we define recursive equations over (max,+) and (min,+) algebras in order to describe the so-called *worst-case* and *optimal-case* behaviors of the automata. It is then shown that these representations have direct applications to evaluate the performances of the systems with low computation complexity. In addition, it is possible to model the influence of exogenous inputs by an additive term in these representations. We then get model equations that are analogous to standard nonautonomous state equations in (max,+) and (min,+) algebras, and we can consider to adapt existing control laws for (max,+) linear systems. As a first attempt in doing so, an open-loop control is investigated for (max,+) automata.

This paper is organized as follows. Next Section contains a brief reminder on idempotent semirings, (max,+) automata and their properties. In Section 3, the representations for extremal behaviors of (max,+) automata are introduced. These lead to some performance evaluation elements described and compared with related results in the literature. In Section 4, inputs are defined for the representations in order to capture some external influences on

[1] Beyond the scope of discrete event systems, there are important applications for image and speech processing, and more generally, weighted automata constitute a theoretical object which is extensively studied (see [9] for an overview).

(max,+) automata. This makes it possible to deal with control, and an open-loop control law is then proposed. The concepts and results are illustrated by means of elementary examples through the paper. A more realistic example is studied in section 5: a jobshop system is considered and some potential contributions for performance evaluation and control of this system are mentioned. A conclusion and some prospects are given in Section 6.

2 PRELIMINARIES

2.1 Dioids

Necessary algebraic concepts on dioids are briefly recalled in this section (see the monographs [2] and [13] for an exhaustive presentation).
A *dioid* is a *semiring* in which the addition $\oplus$ is idempotent. The addition (resp. the multiplication $\otimes$) admits as null element ε (resp. as identity element e)[2]. Due to the idempotency of $\oplus$, a natural order relation is defined by $a \succeq b \iff a \oplus b = a$ ($a \oplus b$ is the least upper bound of $\{a, b\}$). A dioid is *complete* if $\otimes$ distributes over infinite sums and if every subset admits a least upper bound. The greatest lower bound of $\{a, b\}$, noted $\wedge$, then exists as $a \wedge b$ defined by $\oplus_{\{x \preceq a, x \preceq b\}} x$ belongs to the dioid. The greatest element of a complete dioid is noted $\top$ and is equal to the $\oplus$-sum of all its elements.

Example 1 If Σ is a finite set (*alphabet*), the free monoid on Σ is defined as the set Σ^* of finite words with letters in Σ. A *word* $w \in \Sigma^*$ can be written as a sequence $w = a_1 a_2 \ldots a_n$ with $a_1, a_2, \ldots, a_n \in \Sigma$ and n a natural number. *Formal languages* are subsets of the free monoid Σ^*. The set of formal languages, with the union of languages playing the role of addition and concatenation of languages playing the role of multiplication, is a dioid. The zero language is $\varepsilon = \{\}$, the unit language is denoted $e = \{\epsilon\}$ where ϵ is the empty (zero length) string. We say that $u = u_1 \ldots u_k \in \Sigma^*$ is a *subword* of $w \in \Sigma^*$ if there exists a factorization $w = w_1 u_1 w_2 \ldots w_k u_k w_{k+1}$ with $w_i \in \Sigma^*, i = 1, \ldots k+1$. The corresponding subword order on Σ^* is $u \preceq w$ iff u is a subword of $w \in \Sigma^*$.

Example 2 The set $\mathbb{R} \cup \{\pm\infty\}$ with the maximum (resp. the minimum) playing the role of addition and conventional addition playing the role of multiplication is a complete dioid, denoted $\overline{\mathbb{R}}_{\max}$ (resp. $\overline{\mathbb{R}}_{\min}$) , with $e = 0$ and $\varepsilon = -\infty$ (resp. $\varepsilon = +\infty$) and is usually called (max,+) algebra (resp. (min,+) algebra).

The set of $n \times n$ matrices with coefficients in dioid $\overline{\mathbb{R}}_{\max}$ (resp. $\overline{\mathbb{R}}_{\min}$), endowed with the matrix addition and multiplication conventionally defined from $\oplus$ and $\otimes$, is also a dioid, denoted $\overline{\mathbb{R}}_{\max}^{n \times n}$ (resp. $\overline{\mathbb{R}}_{\min}^{n \times n}$). The zero element is the matrix exclusively composed of ε. We denote I_n the identity element, which

[2] As usual, we will often omit the multiplication sign $\otimes$, that is for example we write AB instead of $A \otimes B$.

is the matrix with e on the diagonal and ε elsewhere. Note that, to be able to multiply a $n \times n$ matrix with a $n \times 1$ vector, this vector should be embedded in $\overline{\mathbb{R}}_{\max}^{n\times n}$ by adding $n-1$ columns full of ε. To lighten the presentation, this construction is often omitted in the following (without affecting the results).

2.2 (Max,+) automata

Automata with multiplicities in the $\overline{\mathbb{R}}_{\max}$ semiring are called (max,+) automata (see [10] for a deeper introduction).
A (max,+) *automaton* G is a quadruple (Q, Σ, α, μ) where

- Q and Σ are finite sets of states and of events;
- $\alpha \in \overline{\mathbb{R}}_{\max}^{1\times|Q|}$ is such that $\alpha_q = \varepsilon$ or e, and a state is said to be initial when $\alpha_q = e$ ($Q_i \subset Q$ denotes the set of initial states);
- $\mu : \Sigma^* \to \overline{\mathbb{R}}_{\max}^{|Q|\times|Q|}$ is a morphism specified by the matrix family $\mu(a) \in \overline{\mathbb{R}}_{\max}^{|Q|\times|Q|}$, $a \in \Sigma$. For a string $w = a_1 \dots a_n$, we have

$$\mu(w) = \mu(a_1 \dots a_n) = \mu(a_1) \dots \mu(a_n),$$

  where the matrix multiplication involved here, is the one of $\overline{\mathbb{R}}_{\max}^{|Q|\times|Q|}$. A coefficient $[\mu(a)]_{pq} \neq \varepsilon$ means that the occurrence of event a causes a state transition from p to q, and the value $[\mu(a)]_{pq}$ is interpreted as the duration associated to a (namely, the time activation of event a before it can occur).

Remark 1 This definition is slightly different from that in [10] where initial and final delays are considered. In the present paper, we restrict our attention to (max,+) automata in which the initial delays (that is the coefficients in α different from ε) are all equal to $e = 0$ (this assumption is without loss of generality since an adequate transformation is always possible). In addition, the vector of final delays is not considered, hence all states can be thought of as final (marked) states (as it is the case for heap automata [12]).

Example 3 Figure 1 is an example of graphic representation which can be associated with every (max,+) automaton:

- the nodes correspond to states $q \in Q$;
- an arrow exists from state $p \in Q$ to state q if there exists an event $a \in \Sigma$ such that $[\mu(a)]_{pq} \neq \varepsilon$: it represents the state transition when event a occurs. The arrow is labelled by the event and the multiplicity associated with the state-transition, namely '$a/[\mu(a)]_{pq}$'.
- an input arrow symbolizes an initial state.

For this example, we have $Q = \{I, II\}$, $\Sigma = \{a, b\}$, and

$$\alpha = \begin{pmatrix} e & e \end{pmatrix}, \ \mu(a) = \begin{pmatrix} \varepsilon & 3 \\ \varepsilon & 2 \end{pmatrix}, \mu(b) = \begin{pmatrix} \varepsilon & \varepsilon \\ 2 & \varepsilon \end{pmatrix}.$$

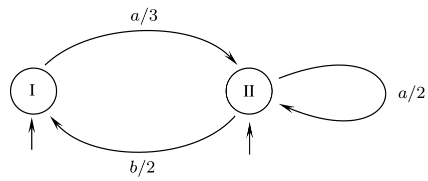

Fig. 1 A (max,+) automaton.

If $[\mu(a)]_{pq} \neq \varepsilon$, (p, a, q) denotes the *transition* in G, and H refers to the set of transitions

$$H \triangleq \{(p, a, q) \in Q \times \Sigma \times Q \mid [\mu(a)]_{pq} \neq \varepsilon\}. \tag{1}$$

In order to lighten the presentation, an element of H can also be denoted by an integer corresponding to its position in the set (it is then assumed that H constitutes a fixed sequence). For a given state $p \in Q$, we define the set $H_p \subset H$ by:

$$H_p = \{(r, a', s) \in H \mid r = p\}.$$

Let $m \geq 1$ and $\pi = (q_0, a_1, q_1)(q_1, a_2, q_2) \ldots (q_{m-1}, a_m, q_m)$ be a sequence of transitions. We call π a *path* from q_0 to q_m. We denote $\sigma(\pi)$ the product $\otimes$ of the weights on π, that is

$$\sigma(\pi) = \bigotimes_{i=1,\ldots,m} [\mu(a_i)]_{q_{i-1},q_i} = \sum_{i=1,\ldots,m} [\mu(a_i)]_{q_{i-1},q_i}. \tag{2}$$

Let $p, q \in Q$ and $w \in \Sigma^*$. We denote by $p \overset{w}{\rightsquigarrow} q$ the set of paths from p to q which are labeled by w (for $P, R \subset Q$, $P \overset{w}{\rightsquigarrow} R$ denotes the union of $p \overset{w}{\rightsquigarrow} q$ for every $p \in P$, $q \in R$). It can be shown that

$$[\mu(a_1 a_2 \ldots a_m)]_{q_0 q_m} = \bigoplus_{\pi \in q_0 \overset{a_1 \ldots a_m}{\rightsquigarrow} q_m} \sigma(\pi)\ . \tag{3}$$

The usual representation for a (max,+) automaton describes its dynamic evolution by means of a vector $x(w) \in \overline{\mathbb{R}}_{\max}^{1\times|Q|}$ defined by

$$x(w) = \alpha\mu(w). \tag{4}$$

The element $[x(w)]_q$ is interpreted as the greatest date over all initial states at which state q is reached when w is completed (with the convention that $[x(w)]_q = \varepsilon$ if the state q is not reached when w is completed). The elements of x are *generalized daters*, and we have

$$\begin{cases} x(\epsilon) = \alpha, \\ x(wa) = x(w)\mu(a). \end{cases} \tag{5}$$

It is easy to check that

$$x(w)_q = \bigoplus_{\pi \in Q_i \overset{w}{\rightsquigarrow} q} \sigma(\pi) \ . \tag{6}$$

The generalized daters can equivalently be written as formal series

$$x = \bigoplus_{w \in \Sigma^*} x(w)w. \tag{7}$$

These formal power series are often referred to as the behavior of the (max,+) automaton.

A $(\max, +)$ automaton is said to be *deterministic* if

- it has a unique initial state, namely, there is a unique $q \in Q$ such that $\alpha_q \neq \varepsilon$;
- and from each state, no two state transitions share the same event, namely, if for all $a \in \Sigma$ each line of $\mu(a)$ contains at most one element not equal to ε.

Different definitions can be found for the notion of *ambiguity* in the literature. Commonly, an automaton is said to be unambiguous if there is at most one path with a given origin, end, and label (see for example [4]), that is $\forall p, q \in Q, \forall w \in \Sigma^*; |p \overset{w}{\rightsquigarrow} q| \leq 1$. Whereas in [16, 15] "unambiguous" rather characterizes automata in which there is at most one successful path for any label, that is $\forall w \in \Sigma^*; |Q_i \overset{w}{\rightsquigarrow} Q_f| \leq 1$ (Q_f stands for the subset of final states). The latter definition is more restrictive than the former. In the present paper, we need to distinguish automata with a related property that we call "strongly unambiguous" (to signify a restriction compared to the common definition for unambiguous, but it is a more general notion than the second definition above).

Definition 1 A (max,+) automaton is said to be *strongly unambiguous* if there is at most one path with given end and label starting from any initial state, that is $\forall q \in Q, \forall w \in \Sigma^*; |Q_i \overset{w}{\rightsquigarrow} q| \leq 1$.

Deterministic automata are (strongly) unambiguous. In strongly unambiguous $(\max, +)$ automata, for given $q_m \in Q$, $w \in \Sigma^*$, $Q_i \overset{w}{\rightsquigarrow} q_m$ is the empty set or a singleton. Let us denote $\pi = (q_0, a_1, q_1)(q_1, a_2, q_2) \ldots (q_{m-1}, a_m, q_m)$ the unique path recognizing $a_1 a_2 \ldots a_m$ ($q_0 \in Q_i$), Eqs. (3)-(6) are then reduced to

$$x(w)_{q_m} = [\alpha\mu(a_1 a_2 \ldots a_m)]_{q_m} = \mu(a_1 a_2 \ldots a_m)_{q_0, q_m} = \bigotimes_{i=1\ldots m} [\mu(a_i)]_{q_{i-1}, q_i} \tag{8}$$

3 Representations for extremal behaviors of (max,+) automata

In this section, several representations are introduced to model the extremal behaviors of (max,+) automata. Instead of exactly describing their evolution like the usual models (5)-(7) (by differentiating each event-occurrence), these representations bring only bounds for the behavior of the corresponding DES (by considering all the possible event-occurrences). These representations are proposed in Subsections 3.1 and 3.2. We can argue that the loss of accuracy is compensated by the low complexity of the manipulations with these new models. In particular, it is shown in Subsection 3.3 that several performance indicators can then be evaluated in polynomial time complexity. Another advantage of these representations is underlined in next section: it is possible to define the influence of exogenous inputs to get standard nonautonomous state equations in (max,+) and (min,+) algebras, and we can consider to transpose existing control laws for (max,+) linear systems.

3.1 Representation for extremal behaviors in terms of sequence durations

Let us define matrix $A \in \mathcal{D}^{|H|\times|H|}$ as follows ($\mathcal{D}$ can stand for $\overline{\mathbb{R}}_{\max}$ or $\overline{\mathbb{R}}_{\min}$). For j and k indices of transitions (p, a, q) and (r, a', s) in H,

$$A_{jk} = \begin{cases} [\mu(a')]_{rs} & \text{if } s = p, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{9}$$

Example 4 The nondeterministic, but strongly unambiguous, (max,+) automaton G_1 represented in figure 2 is such that

$$H = \{(I, a, I), (I, b, II), (II, c, II)\},$$

and

$$A = \begin{pmatrix} 3 & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon \\ \varepsilon & 2 & 1 \end{pmatrix}.$$

For example $A_{2,1} = 3$ brings the information that transition (I, b, II) can occur consecutively to the occurrence of transition (I, a, I) which has a duration equal to 3 time units.

Definition 2 A vector $x \in \mathcal{D}^{|H|\times 1}$ is said to be homogeneous if $\forall p \in Q$, $\forall i, j \in H_p$, $x_i = x_j$, i.e., the entry of an homogeneous vector corresponding to a transition $(p, a, q) \in H$ depends only on the origin p of the transition. Since the sum of $\mathcal{D}$ is idempotent, this value is also equal to $\bigoplus_{i\in H_p} x_i$. We shall let x_p denote this value.

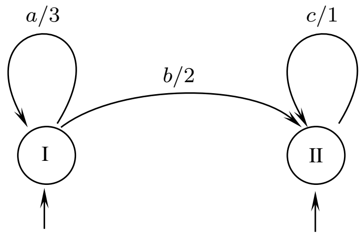

Fig. 2 Automaton G_1.

Since the rows of A corresponding to transitions with the same origin are identical, we deduce that for every homogeneous vector $y \in \mathcal{D}^{|H|\times 1}$, the vector $x = A \otimes y$ is homogeneous. We define inductively the following sequence of homogeneous vectors $x(n) \in \mathcal{D}^{|H|\times 1}$ for $n \in \mathbb{N}$ by:

$$\begin{cases} x(1)_{(p,a,q)} = \alpha_p & \text{for every } (p,a,q) \in H \\ x(n+1) = A \otimes x(n) & \text{for } n \geq 1. \end{cases} \tag{10}$$

Proposition 1 *Let $\overline{A}$ and $\overline{x}(n)$, $n \in \mathbb{N}$ be defined respectively by* (9) *and* (10) *with $\mathcal{D}$ corresponding to $\overline{\mathbb{R}}_{\max}$. Then $\overline{x}(n+1)_p$, $n \in \mathbb{N}$, $p \in Q$ is the maximum element of $\sigma_{n,p} = \{x(w)_p \mid |w| = n\}$ for each $p \in Q$, that is the maximum completion date among sequences of length n leading to state p.*

Proof 1 *The proof proceeds by induction.*
Base case. *For every $p \in Q$, we have by definition*

$$\overline{x}(1)_p = \alpha_p = x(\epsilon)_p = \{x(w)_p \mid |w| = 0\}$$

Inductive case.

$$\begin{aligned}
&\max \sigma_{n+1,p} \\
&\quad = \bigoplus_{\{w \mid |w|=n+1\}} x(w)_p \\
&\quad = \bigoplus_{q\in Q} \bigoplus_{\{a \mid (q,a,p)\in H\}} \bigoplus_{\{w' \mid |w'|=n\}} x(w')_q \otimes \mu(a)_{qp} \quad \text{(by Eq. (5))} \\
&\quad = \bigoplus_{q\in Q} (\bigoplus_{\{w' \mid |w'|=n\}} x(w')_q) \otimes (\bigoplus_{\{a \mid (q,a,p)\in H\}} \mu(a)_{qp}) \\
&\quad = \bigoplus_{q\in Q} \overline{x}(n+1)_q \otimes (\bigoplus_{\{a \mid (q,a,p)\in H\}} \mu(a)_{qp}) \quad \text{(by inductive hyp.)} \\
&\quad = \bigoplus_{q\in Q} \bigoplus_{(q,a,p)\in H_q} \mu(a)_{qp} \otimes \overline{x}(n+1)_{(q,a,p)} \\
&\quad = \bigoplus_{(q,a,p)\in H_q} \bigoplus_{j\in H_p} \overline{A}_{j,(q,a,p)} \otimes \overline{x}(n+1)_{(q,a,p)} \quad \text{(by Eq. (9))} \\
&\quad = [\overline{A} \otimes \overline{x}(n+1)]_p \\
&\quad = \overline{x}(n+2)_p
\end{aligned}$$

Example 5 Let us consider again automaton G_1 studied in Example 4 and depicted in figure 2. We have

$$\overline{x}(n) = \begin{bmatrix} \overline{x}(n)_{(I,a,I)} \\ \overline{x}(n)_{(I,b,II)} \\ \overline{x}(n)_{(II,c,II)} \end{bmatrix}.$$

Vector $\overline{x}$ satisfies Eq. (10), that is:

$$\overline{x}(1) = \begin{pmatrix} e \\ e \\ e \end{pmatrix}, \quad \overline{x}(n+1) = \begin{pmatrix} 3 & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon \\ \varepsilon & 2 & 1 \end{pmatrix} \otimes \overline{x}(n).$$

Table 1 contains the first values obtained thanks to this recurrence in $\overline{\mathbb{R}}_{\max}$.

Table 1

n	1	2	3	4	5	...
$\overline{x}_{(I,a,I)}(n)$	e	3	6	9	12	...
$\overline{x}_{(I,b,II)}(n)$	e	3	6	9	12	...
$\overline{x}_{(II,c,II)}(n)$	e	2	5	8	11	...

For example, the possible sequences of length 4 leading to state II are $\{cccc, bccc, abcc, aabc, aaab\}$. We obtain by means of usual representation (5)

$$x(cccc)_{II} = 4,\ x(bccc)_{II} = 5,\ x(abcc)_{II} = 7, x(aabc)_{II} = 9,\ x(aaab)_{II} = 11,$$

which leads to $\sigma_{4,II} = \{4, 5, 7, 9, 11\}$.
On the other hand, we have

$$H_{II} = \{(II, c, II)\},$$

hence

$$\overline{x}(5)_{II} = \overline{x}(5)_{(II,c,II)} = 11,$$

which corresponds to the maximum element of $\sigma_{4,II}$, that is the maximum completion time for sequences of length 4 leading to state II.

The next proposition contributes to characterize the optimal behavior of automata. This representation is formulated in (min,+) algebra but we must keep in mind that it describes (max,+) automata.

Proposition 2 *Let $\underline{A}$ and $\underline{x}(n)$, $n \in \mathbb{N}$ be defined respectively by* (9) *and* (10) *with $\mathcal{D}$ corresponding to $\overline{\mathbb{R}}_{\min}$. Then $\underline{x}(n+1)_p$, $n \in \mathbb{N}$, $p \in Q$ is a minorant of $\sigma_{n,p} = \{x(w)_p \mid |w| = n\}$ for each $p \in Q$, that is a minorant of the possible completion dates of sequences of length n leading to state p.*

Proof 2 $\forall q_{n+1} \in Q$,

$$\begin{aligned}
\underline{x}(n+1)_{q_{n+1}} &= [\underline{A} \otimes \underline{x}(n)]_{q_{n+1}} && \textit{(using (10))},\\
&= \bigoplus_{l \in H} \underline{A}_{q_{n+1},l} \otimes \underline{x}(n)_l, \\
&= \bigoplus_{q_n \in Q} \bigoplus_{k \in H_{q_n}} \underline{A}_{q_{n+1},k} \otimes \underline{x}(n)_k, \\
&= \bigoplus_{q_n \in Q} \bigoplus_{k \in H_{q_n}} \underline{A}_{q_{n+1},k} \otimes \underline{x}(n)_{q_n}, && (\underline{x} \textit{ is homogeneous}).
\end{aligned}$$

By Eq. (9) defining $\underline{A}$, we have

$$\forall q_{n+1} \in Q, \bigoplus_{k \in H_{q_n}} \underline{A}_{q_{n+1},k} = \bigoplus_{a_n \in \Sigma} \mu(a_n)_{q_n,q_{n+1}}.$$

This leads to, $\forall q_{n+1} \in Q$,

$$\begin{aligned}
&\underline{x}(n+1)_{q_{n+1}} \\
&= \bigoplus_{q_n \in Q} \bigoplus_{a_n \in \Sigma} \mu(a_n)_{q_n,q_{n+1}} \otimes \underline{x}(n)_{q_n}, \\
&= \bigoplus_{q_n \in Q} \bigoplus_{a_n \in \Sigma} \underline{x}(n)_{q_n} \otimes \mu(a_n)_{q_n,q_{n+1}}, \\
&= \bigoplus_{q_n \in Q} \dots \bigoplus_{q_1 \in Q} \bigoplus_{a_n \in \Sigma} \dots \bigoplus_{a_n \in \Sigma} \underline{x}(1)_{q_1} \otimes \mu(a_1)_{q_1,q_2} \otimes \dots \otimes \mu(a_n)_{q_n,q_{n+1}}, \\
&= \bigoplus_{q_n \in Q} \dots \bigoplus_{q_1 \in Q} \bigoplus_{a_n \in \Sigma} \dots \bigoplus_{a_1 \in \Sigma} \alpha_{q_1} \otimes \mu(a_1)_{q_1,q_2} \otimes \dots \otimes \mu(a_n)_{q_n,q_{n+1}}, \qquad (*) \\
&= \min_{q_n \in Q} \dots \min_{q_1 \in Q} \min_{a_n \in \Sigma} \dots \min_{a_1 \in \Sigma} \underline{x}(1)_{q_1} + \mu(a_1)_{q_1,q_2} + \dots + \mu(a_n)_{q_n,q_{n+1}}, \ (**)
\end{aligned}$$

Clearly, the term $\alpha_{q_1} \otimes \mu(a_1)_{q_1,q_2} \otimes \dots \otimes \mu(a_n)_{q_n,q_{n+1}}$ is not equal to $\varepsilon(=+\infty)$ if q_1 is an initial state and if the path $(q_1, a_1, q_2) \dots (q_n, a_n, q_{n+1})$ exists in the automaton. We then have $[\underline{x}(n+1)]_{q_{n+1}} = \varepsilon$ if there is no sequence of length n leading to q_{n+1}. Let us assume that $[\underline{x}(n+1)]_{q_{n+1}} \neq \varepsilon$ and denote $\underline{a}_1 \dots \underline{a}_n \in \Sigma^*$ a label successful for the minimization in $(**)$. We consider two cases:

(i) If set $Q_i \overset{\underline{a}_1 \dots \underline{a}_n}{\rightsquigarrow} q_{n+1}$ is a singleton, no $\min_{q_i \in Q}$, $i = 1, \dots n$, operates in $(**)$ and $\underline{x}(n+1)_{q_{n+1}}$ is equal to $[\alpha\mu(\underline{a}_1) \otimes \dots \otimes \mu(\underline{a}_n)]_{q_{n+1}} = x(\underline{a}_1 \dots \underline{a}_n)_{q_{n+1}}$ (by identification with Eq. (8)).

(ii) If there exist several paths with label $\underline{a}_1 \dots \underline{a}_n$ from a state in Q_i to q_{n+1}, then $\mu(\underline{a}_1)_{q_1,q_2} \otimes \dots \otimes \mu(\underline{a}_n)_{q_n,q_{n+1}}$ with $q_1 \in Q_i$ may take different values. In this case, Eq. $(**)$ computes the minimum of these values whereas the weight in the (max,+) automaton $x(\underline{a}_1 \dots \underline{a}_n)_{q_{n+1}} = [\alpha\mu(\underline{a}_1) \otimes \dots \otimes \mu(\underline{a}_n)]_{q_{n+1}}$ corresponds to the maximum of these values. In other words, the value computed by Eq. $(**)$ may be strictly less than $x(\underline{a}_1 \dots \underline{a}_n)_{q_{n+1}}$.

Arguments stated in (i) and (ii) lead to conclude that is $\underline{x}(n+1)_{q_{n+1}}$ is a minorant of $\sigma_{n,q_{n+1}} = \{x(w)_{q_{n+1}} \mid |w| = n\}$.

Proposition 3 *If G is strongly unambiguous, then $\underline{x}(n+1)_p$ is the minimum element of $\sigma_{n,p}$.*

Proof 3 *If G is strongly unambiguous, then $Q_i \overset{\underline{a}_1 \dots \underline{a}_n}{\rightsquigarrow} q_{n+1} \leq 1$ for every $q_{n+1} \in Q$ and every $\underline{a}_1 \dots \underline{a}_n \in \Sigma^*$. Only the case (i) in the proof of Prop. 2 is possible, and $\underline{x}(n+1)_{q_{n+1}}$ then belongs to set $\sigma_{n,q_{n+1}} = \{x(w)_{n+1} \mid |w| = n\}$.*

Example 6 Let us consider again automaton G_1 represented in figure 2. It is strongly unambiguous, and as claimed in Proposition 3, then $\underline{x}(n+1)_p$ gives the minimum completion time for sequences of length n leading to state p.
Table 2 contains the first values obtained for $\underline{x}$ by means of recurrence (10) in $\overline{\mathbb{R}}_{\min}$.

Table 2

n	1	2	3	4	5	...
$\underline{x}_{I,a,I}(n)$	e	3	6	9	12	...
$\underline{x}_{I,b,II}(n)$	e	3	6	9	12	...
$\underline{x}_{II,c,II}(n)$	e	1	2	3	4	...

We have mentioned in example 5 that the set of possible completion dates for the sequences of length 4 and leading to state II, is given by

$$\sigma_{4,II} = \{4, 5, 7, 9, 11\},$$

and $H_{II} = \{(II, c, II)\}$. The minimum element of set $\sigma_{4,II}$ is given by:

$$\underline{x}(5)_{(II,c,II)} = 4.$$

3.2 Representations for extremal behaviors in terms of sequence lengths

Propositions 1-3 make it possible to evaluate the maximum and minimum execution time for a given length of sequences. Dually, the next propositions estimate the minimum and maximum lengths for sequences completed before a given date. Note that we use variables which depend on time t and which can wrongly remind counter variables exclusively manipulated in (min,+) algebra (see for ex. [2, §5.2]). The present variables are implied in representations over (max,+) and (min,+) algebras to model extremal behaviors of (max,+) automata.

Notation 1 *Let T denote the set of possible durations associated with events, that is,*

$$T \triangleq \{\tau \mid \exists a \in \Sigma, \exists p \in Q, \exists q \in Q \text{ with } [\mu(a)]_{pq} = \tau\}.$$

We assume that $T \subset \mathbb{N} \setminus \{0\}$.

We define the matrices denoted $E_\tau \in \mathcal{D}^{|H|\times|H|}$, $\tau \in T$, as follows. For j and k indices of transitions (p, a, q) and (r, a', s) in H.

$$[E_\tau]_{jk} = \begin{cases} 1 & \text{if } s = p \text{ and } [\mu(a')]_{rs} = \tau, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{11}$$

Note that for any homogeneous vector $x \in \mathcal{D}^{|H|\times 1}$ the vectors $E_\tau \otimes x$, $\tau \in T$, are also homogeneous. We define inductively the following sequence of homogeneous vectors $z(t) \in \mathcal{D}^{|H|\times 1}$ for $t \in \mathbb{N}$:

$$\begin{cases} z(0)_{(p,a,q)} = \alpha_p & \text{for every } (p, a, q) \in H \\ z(t) = \bigoplus_{\tau\in T, \tau\leq t} E_\tau \otimes z(t-\tau) \oplus z(t-1) & \text{for } t > 0. \end{cases} \tag{12}$$

Proposition 4 *Let $\overline{E}_\tau$, $\tau =\in T$, and $\overline{z}(t)$, $t \in \mathbb{N}$ be defined respectively by (11) and (12) with $\mathcal{D}$ corresponding to $\overline{\mathbb{R}}_{\max}$. Then $\overline{z}(t)_p$, $t \in \mathbb{N}$, $p \in Q$, is a majorant of set $\gamma_{t,p} = \{|w| \mid x(w)_q \leq t\}$, that is a majorant of the possible lengths among sequences leading to state p before or at date t.*
If G is a strongly unambiguous automaton, then $\overline{z}(t)_p$ is the maximum element of $\gamma_{t,p}$.

Proof 4 *Let $n \geq 0$ and $w \in \Sigma^*$ with $|w| = n$, we denote π_n a path from an initial state with label w, that is $\pi_n \in Q_i \overset{w}{\rightsquigarrow} q_n$, $q_n \in Q$. We use mathematical induction to prove that $\overline{z}(\sigma(\pi_n))_{q_n} \geq n$.*
Base case. *For $n = 0$, we have q_0 an initial state, $\sigma(\pi_0) = 0$ and $\overline{z}(0)_{q_0} = \alpha_{q_0} = 0$.*
Inductive step. *We denote $(q_0, a_1, q_1)\ldots(q_{n_1}, a_n, q_n)$ the successive transitions in π_n and $[\overline{E}_\tau]_{q_n,k}$, $k \in H$, the entries $[\overline{E}_\tau]_{(q_n,a_{n+1},q_{n+1}),k}$ identical for all $a_{n+1} \in \Sigma$ and $q_{n+1} \in Q$. We have*

$$\begin{aligned}
&[\overline{z}(\sigma(\pi_n))]_{q_n}\\
&= \bigoplus_{\tau\in T, t\geq\tau} \overline{z}(\sigma(\pi_n) - \tau)_{q_n} \oplus \overline{z}(\sigma(\pi_n) - 1)_{q_n}\\
&\geq \bigoplus_{\tau\in T, t\geq\tau} \overline{z}(\sigma(\pi_n) - \tau)_{q_n}\\
&= \bigoplus_{\tau\in T, t\geq\tau} \bigoplus_{k\in H} [\overline{E}_\tau]_{q_n,k} \otimes \overline{z}(\sigma(\pi_n) - \tau)_k\\
&\geq \bigoplus_{\tau\in T, t\geq\tau} [\overline{E}_\tau]_{q_n,(q_{n-1},a_n,q_n)} \otimes [\overline{z}(\sigma(\pi_n) - \tau)]_{(q_{n-1},a_n,q_n)},\\
&\qquad\qquad \text{(by specifying for } k = (q_{n-1}, a_n, q_n))\\
&\geq \bigoplus_{\tau\in T, t\geq\tau} [\overline{E}_\tau]_{q_n,(q_{n-1},a_n,q_n)} \otimes [\overline{z}(\sigma(\pi_{n-1}) + \mu(a_n)_{q_{n-1}q_n} - \tau)]_{(q_{n-1},a_n,q_n)},\\
&\qquad\qquad \text{(by definition of } \sigma \text{ and } \pi_n)\\
&= 1 \otimes \overline{z}(\sigma(\pi_{n-1}))_{q_{n-1}},\\
&\qquad\qquad \text{(since } \exists\tau \in T \text{ such that } [\mu(a_n)]_{q_{n-1},q_n} = \tau)\\
&\geq 1 + (n - 1)\ (= n) \qquad\qquad \text{(by inductive assumption)}.
\end{aligned}$$

Now let $t \in \mathbb{R}$ such that $x(w)_{q_n} \leq t$. Since $\overline{z}$ is monotone (i.e. $\overline{z}(t) \geq \overline{z}(t-1)$) we have (based on Eq. (6) for the last inequality):

$$\overline{z}(t)_{q_n} \geq \overline{z}(x(w)_{q_n})_{q_n} \geq \overline{z}(\sigma(\pi_n))_{q_n}.$$

We first have shown that $\overline{z}(\sigma(\pi_n))]_{q_n} \geq |\pi_n|(= n)$ and we deduce that

$$\overline{z}(t)_{q_n} \geq |\pi_n|.$$

The same reasoning leads to the same conclusion for any path π' with label w' such that $t \geq x(w')_{q_n}$, and we can conclude that $\overline{z}(t)_{q_n}$ is a majorant of $\gamma_{t,q_n} = \{|w||x(w)_{q_n} \leq t\}$.

Let us now consider the case of G being strongly unambiguous. We can check that

$$\overline{z}(t)_{q_n} \neq \varepsilon$$

implies that there exists $(q_0, a_1, q_1)\ldots(q_{n-1}, a_n, q_n)$ *with* $q_0 \in Q_i$ *such that*

$$\mu(a_1)_{q_0 q_1}\ldots\mu(a_n)_{q_{n-1}q_n} \leq t\ .$$

By the definition of strong unambiguity, set $Q_i \overset{a_1\ldots a_0}{\rightsquigarrow} q_n$ *is a singleton and by Eq. (8)*

$$x(a_1\ldots a_n)_{q_n} = \mu(a_1)_{q_0 q_1}\ldots\mu(a_n)_{q_{n-1}q_n}.$$

We then have $x(a_1\ldots a_n)_{q_n} \leq t$ *which means that* $\overline{z}(t)_{q_n} = |a_1\ldots a_n|$ *belongs to* $\gamma_{t,q_n} = \{|w||x(w)_{q_n} \leq t\}$*. As it has been shown to be a majorant, we can conclude that* $\overline{z}(t)_{q_n}$ *is the maximum element of this set.*

Example 7 We consider (max,+) automaton G_2 represented in figure 3. Note that G_2 is nondeterministic (from state II two transitions are possible according to b) but is strongly unambiguous.

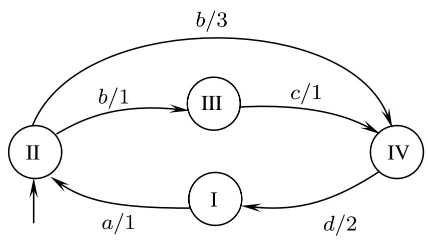

Fig. 3 Automaton G_2.

We have

$$H = \{(I, a, II), (II, b, III), (III, c, IV), (II, b, IV), (IV, d, I)\}, T = \{1, 2, 3\}.$$

Using definition given by Eq. (11)), we get

$$\overline{E}_1 = \begin{pmatrix} \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ 1\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ 1\ \varepsilon\ \varepsilon\ \varepsilon \\ 1\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ 1\ \varepsilon\ \varepsilon \end{pmatrix}, \overline{E}_2 = \begin{pmatrix} \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ 1 \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \end{pmatrix}, \overline{E}_3 = \begin{pmatrix} \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon \\ \varepsilon\ \varepsilon\ \varepsilon\ 1\ \varepsilon \end{pmatrix}.$$

Based on Eqs. (12), we have :

$$\overline{z}(t) = \begin{bmatrix} \overline{z}(t)_{(I,a,II)} \\ \overline{z}(t)_{(II,b,III)} \\ \overline{z}(t)_{(III,c,IV)} \\ \overline{z}(t)_{(II,b,IV)} \\ \overline{z}(t)_{(IV,d,I)} \end{bmatrix},$$

$$\overline{z}(0) = \begin{bmatrix} \varepsilon \\ e \\ \varepsilon \\ e \\ \varepsilon \end{bmatrix}, \quad \overline{z}(t) = \bigoplus_{\tau \in T, \tau \geq t} \overline{E}_\tau \otimes \overline{z}(t-\tau) \oplus \overline{z}(t-1) \ for \ t > 0.$$

Table 3 contains the first values obtained by means of this recurrence in $\overline{\mathbb{R}}_{\max}$.

Table 3

t	0	1	2	3	4	5	6	7	...
$\overline{z}(t)_{(I,a,II)}$	ε	ε	ε	ε	3	3	3	3	...
$\overline{z}(t)_{(II,b,III)}$	e	e	e	e	e	4	4	4	...
$\overline{z}(t)_{(III,c,IV)}$	ε	1	1	1	1	1	5	5	...
$\overline{z}(t)_{(II,b,IV)}$	e	e	e	e	e	4	4	4	...
$\overline{z}(t)_{(IV,d,I)}$	ε	ε	2	2	2	2	2	6	...

For example, the possible sequences leading to state IV and completed at date $t = 7$ are: $\{b, bc, bcdabc\}$. We have $\gamma_{7,IV} = \{1, 2, 6\}$.
On the other hand, we have

$$\overline{z}(7)_{IV} = \overline{z}_{(IV,d,I)}(7) = 6,$$

which corresponds to the maximum element of $\gamma_{7,IV}$, that is the maximum length of sequences leading to state IV and completed before or at $t = 7$.

Proposition 5 *Let $\underline{E}_\tau$, $\tau =\in T$, and $\underline{z}(t)$, $t \in \mathbb{N}$ be defined respectively by (11) and (12) with $\mathcal{D}$ corresponding to $\overline{\mathbb{R}}_{\min}$. Then $\underline{z}(t)_p$, $t \in \mathbb{N}$, $p \in Q$, is a minorant of set $\gamma_{t,p}$, that is a minorant of the possible lengths among sequences leading to state p before or at date t.*
If G is a strongly unambiguous (max,+) automaton, then $\underline{z}(t)_p$ is the minimum element of $\gamma_{t,p}$.

Proof 5 *Straightforward by rewriting proof of Prop. 4 in $\overline{\mathbb{R}}_{\min}$ ($\oplus$ then corresponds to* min *and the order is inverted).*

Example 8 Let us consider again (max,+) automaton G_2 represented in figure 3. Matrices $\underline{z}$ and $\underline{E}_\tau$ are replicas in $\overline{\mathbb{R}}_{\min}$ of $\overline{z}$ and $\overline{E}_\tau$ given in Ex. 7. Table 4 contains the first values obtained for $\underline{z}(t)$.

For example, the possible sequences leading to state I and completed at $t = 5$ are: $\{bd, bcd\}$. We have $\gamma_{5,I} = \{2, 3\}$.
On the other hand,

$$\underline{z}(5)_I = \underline{z}(5)_{(I,a,II)} = 2,$$

which corresponds to the minimum element of $\gamma_{5,I}$.

Remark 2 The results in Propositions 1-5 can be refined as follows.

Table 4

t	0	1	2	3	4	5	6	7	...
$\underline{z}_{I,a,II}(t)$	ε	ε	ε	ε	3	2	2	2	...
$\underline{z}_{II,b,III}(t)$	e	e	e	e	e	e	e	e	...
$\underline{z}_{III,c,IV}(t)$	ε	1	1	1	1	1	1	1	...
$\underline{z}_{II,b,IV}(t)$	e	e	e	e	e	e	e	e	...
$\underline{z}_{IV,d,I}(t)$	ε	ε	2	1	1	1	1	1	...

- In $\overline{\mathbb{R}}_{\max}$, $\overline{x}(n+1)_{(p,a,q)}$ gives the latest date at which the transition (p,a,q) can occur consecutively to n events.
  Then $\overline{x}(n+1)_{(p,a,q)}+[\mu(a)]_{pq}$ is the maximum execution time for sequences of length $(n+1)$ ending with event a and leading to q.
- In $\overline{\mathbb{R}}_{\min}$, $\underline{x}(n+1)_{p,a,q}$ minors the date at which transition (p,a,q) can occur consecutively to n events.
  Then $\underline{x}(n+1)_{(p,a,q)}+[\mu(a)]_{pq}$ is a minorant for the completion dates of sequences of length $(n+1)$ ending with event a and leading to q.
- In $\overline{\mathbb{R}}_{\max}$, $\overline{z}(t)_p$ majors the length of sequences preceding transition (p,a,q) and completed at t.
  Then $\overline{z}(t)_p+1$ majors the length of sequences ending by event a, leading to state q and completed at $t+[\mu(a)]_{pq}$.
- In $\overline{\mathbb{R}}_{\min}$, $\underline{z}(t)_p$ minors the length of sequences preceding transition (p,a,q) and completed at t.
  Then $\overline{z}(t)_p+1$ minors the length of sequences ending by event a, leading to state q and completed at $t+[\mu(a)]_{pq}$.

Note that by iterating this reasoning, it is possible to specify a suffixing subsequence (instead of only an event) to refine the indicators. In addition, it is possible to define symmetric representations to get the indicators refined for prefixing subsequences.

3.3 Applications to performance evaluation

Let us now focus on applications of the new representations for performance evaluation and compare them with closely related paper [10]. The next subsection is devoted to a more general discussion regarding related work.

For some systems, it is important to have knowledge of the maximum execution time for sequences of given length n, that is the maximum element of the set composed of completion times for sequences corresponding to n events. Its calculation is presented in [10] as follows in $\overline{\mathbb{R}}_{\max}$:

$$\begin{aligned} l_n^{worst} &= \bigoplus_{w\in\Sigma^n}\bigoplus_{p\in Q}[x(w)]_p, \\ &= \bigoplus_{w\in\Sigma^n}\bigoplus_{p\in Q}[\alpha\mu(a_1)...\mu(a_n)]_p, \\ &= \bigoplus_{p\in Q}[\alpha M^n]_p, \end{aligned}$$

with

$$M = \bigoplus_{a \in \Sigma} \mu(a). \tag{13}$$

The time complexity of the computation above is $O(n|Q|^3)$ (complexity for n multiplications in $\overline{\mathbb{R}}_{\max}^{|Q|\times|Q|}$).
The representation introduced in Proposition 1 also enables us to evaluate this indicator since

$$l_n^{worst} = \bigoplus_{j \in H} [\overline{x}(n+1)]_j = \bigoplus_{j \in H} [\overline{A}^n \overline{x}(1)]_j. \tag{14}$$

A direct application of this formula would lead to a complexity in $O(n|H|^3)$ time. Nevertheless, it is possible to reduce the complexity by exploiting the particular form of matrix $\overline{A}$ derived from A defined by Eq. (9). In fact, one can observe that, by definition, all the non-ε entries of a column of A are identical. Let us then denote

- $A_{\bullet,k}$ the column of A with index k,
- H_k for $k = (r, a', s)$, the subset of H defined by $H_k = \{(p, a, q) \in H | p = s\}$,
- $a_{\bullet,k}$ the identical value for the non-ε entries in column k of A.

It is easy to check that

$$[A^n]_{\bullet,k} = \left(\bigoplus_{j \in H_k} A_{\bullet,j} \right) \otimes a_{\bullet,k}^n \ .$$

This property makes it possible to compute l_n^{worst} expressed by Eq. (14) in $O(n+|H|^2)$ time. For automata in which the number of arcs $|H|$ is close to the number of states $|Q|$, we can claim that our approach has a lower complexity than the method in [10][3]. In addition, time complexity of our approach is less affected when length n of the sequence grows.

Remark 3 In [10], authors show how to compute the maximum execution time when constrained in a sublanguage. This increases noticeably time complexity because matrix M in (13) then corresponds to a tensor product of matrices μ. As mentioned in Remark 2, our approach also makes it possible to refine the indicator by specifying chosen transitions (or even subsequences) which prefix or suffix the sequences. Note that for these refinements, time complexity of our approach is not affected.

For other systems, it is important to be able to compute the minimum execution time for sequences of given length n. It is referred to as the *optimal case* in [10] and formulated as follows in $\overline{\mathbb{R}}_{\min}$:

$$l_n^{opt} = \bigoplus_{w \in \Sigma^n} \bigoplus_{p \in Q} x(w)_p.$$

[3] It must be noted that there exist automata in which $|Q| < |H|$, as well as others in which $|Q| > |H|$, and the comparison between methods complexity may then lead to contradictory conclusions.

The algorithm which is given in [10] only applies to a reduced class of (max,+) automata (deterministic automata) and is announced to "suffer of a greater complexity" than the worst case. In [28], it is shown that the computation of the "optimal-case" is NP-complete[4].
In proposition 2, based on the new representation (10), a minorant of this element (or even the minimum element if the automaton is strongly unambiguous) is obtained by a simple recurrence in $\overline{\mathbb{R}}_{\min}$, as:

$$l_n^{opt} \geq \bigoplus_{j \in H} [\underline{x}(n+1)]_j = \bigoplus_{j \in H} [\underline{A}^n \underline{x}(1)]_j.$$

The time complexity of this computation is the same as that of the "maximum execution time", that is $O(n + |H|^2)$. Let us stress that, compared to [10], Proposition 1 extends the class of (max,+) automata (from deterministic to strongly unambiguous)[5] for which the minimum execution time can be computed, and has a better time complexity. To the best of our knowledge, there does not exist algorithm for computing the minimum execution time for ambiguous automata. Our approach makes it possible to compute a minorant with linear complexity.

Remark 4 As in [10], our approach can take profit of the spectral properties of matrices defined over $\overline{\mathbb{R}}_{\max}$ or $\overline{\mathbb{R}}_{\min}$. In few words (see Section V in [10] for details) a matrix $A \in \mathcal{D}^{k \times k}$ ($\mathcal{D}$ stands for $\overline{\mathbb{R}}_{\max}$ or $\overline{\mathbb{R}}_{\min}$) can admit a cyclicity property, that is, we have $A^{n+c} = \lambda^c A^n$ for n large enough and $c \in \mathbb{N}$, $\lambda \in \mathbb{R}$. The scalar λ corresponds to the spectral radius of A and several algorithms exist to compute it. Let us remind that if A is irreducible, then Karp's algorithm computes λ in $O(|H| \times E)$ time where E is the number of non-ε entries of A and Howard's algorithm shows experimentally an almost linear average execution time [7]. For $\overline{A}$ (resp. $\underline{A}$), λ characterizes the growth of the maximum (resp. minimum) execution time for n large enough, and these indicators can be easily deduced for larger lengths of sequences. In future works, we plan to exploit the particular forms of the matrices we manipulate in order to make more explicit the contributions of spectral theory for our performance indicators, and in particular:

- to identify conditions on (max,+) automata for the irreducibility of matrices A, and more generally to study their *robustness* [6], that is conditions for which A admits a cyclicity property;

[4] The result is actually stated for a subclass of systems (see discussion in §3.4) with direct extension to any (max,+) automaton.

[5] Note that, unlike (boolean) finite automata, nondeterministic (max,+) automata cannot always be determinized, that is transformed into equivalent deterministic (max,+) automata (see e.g. [10, 23]). Despite the fact that it was studied by numerous researchers, this problem is still rather open, and to the best of our knowledge, strongly unambiguous (max,+) automata don't satisfy necessarily conditions which are known in the literature to be sufficient for determinization.

– to study eigenvectors of A which would give initial vectors for recurrence (10) so that execution times grow according to λ immediately (i.e. from sequence length $n = 0$).

The representations from Prop. 4 and 5 proposed in section 3.3 enable us to compute two other performance indicators which, to the best of our knowledge, have not been studied in the literature:

(i) A majorant of the greatest number of events occurring until a given time instant is obtained using:

$$\bigoplus_{j \in H} [\overline{z}(t)]_j .$$

(ii) A minorant of the least number of events executed before, or at, a given time instant is obtained using:

$$\bigoplus_{j \in H} [\underline{z}(t)]_j .$$

For strongly unambiguous automata greatest and least numbers of events are obtained (instead of majorants and minorants). The computation of these indicators has a similar time complexity than the indicators presented previously since it merely implies matrixes multiplications in $\overline{\mathbb{R}}_{\max}$ and $\overline{\mathbb{R}}_{\min}$.

Remark 5 The refinements mentioned in Remark 3 have analogous formulations for the minimum execution time and the two indicators (i)-(ii) (using arguments in Remark 2).

Remark 6 To give more explicit illustrations of the results, let us sketch out some possible realistic applications:

– For verification and validation activities of real-time systems, maximum and minimum execution times give guarantees about the worst and best case completion times, whereas indicators (i)-(ii) provide guarantees on the minimum and maximum amount of tasks completed at a given date.
– For manufacturing systems in which several schedules are possible, maximum and minimum execution times give bounds for the makespan, and indicators (i)-(ii) provide bounds on the number of parts that can be delivered at a given date.

3.4 Related work

In [28, 27] authors consider time-weighted systems by associating to finite-state automata a ”time-weighted function” and a ”mutual exclusion function”. The time-weighted function assigns a nonnegative value to each event, interpreted as the duration of executing the event. From a time-weighted system, they build a heap model (heap models are particular (max,+) automata, see [12]) in which the upper contour of the piece associated to an event is defined from the time-weighted function. In this case, the upper contour has the same value for all the slots occupied by the piece. In the corresponding (max,+) automaton, all the transitions involving a given event have the same weight (one possible duration for an event whatever the current state). As a consequence, paths with a same label have an identical weight. For these systems, it is shown in [28] that the computation of minimum execution time is NP-complete, and an efficient algorithm is presented for the computation of the maximum execution time (time complexity is $O(|H||\mathcal{R}|^2)$, where $\mathcal{R}$ is a clique covering of Σ). The comparison with the results in this paper has to be weighted by the fact that a more restrictive class of systems is considered. At first sight, this class is less general than strongly unambiguous automata. In fact, on the one hand, one can observe that as paths with a same label have equal weights, then maximization in Eq. (6) doesn't operate and the differentiation upon ambiguity is no more meaningful. More precisely, since for any $w \in \Sigma^*$, $q \in Q$, we have $x(w)_q = \sigma(\pi)$ for all $\pi \in Q_i \overset{w}{\rightsquigarrow} q$, we can claim that all the results specified in this paper for strongly unambiguous automata also apply to (max,+) automata derived from time-weighted systems as in [28, 27]. On the other hand, strongly unambiguous automata in which events have several possible durations cannot be captured by time-weighted systems defined in [28].

Our work is also to be compared with other approaches for discrete event systems, and in particular, quantitative analysis of timed and time Petri nets. Several methods have used enumerative methods (that is techniques based on the construction of the state classes graph and the discrete reachability graph) which suffer from the state space explosion problem. A recent approach [5] proposes to compute efficiently (in polynomial time on the size of the net model) bounds through the solution of linear programming problems derived from the structure of the net, the initial marking and the time interpretation. They consider time Petri nets with firing frequency intervals (that is nets in which conflicting transitions are in extended free-choice conflicts and behave according to frequency interval constraints), as well as a stochastic extension of time Petri nets. In addition, a so-called average operational behavior is assumed, that is, basically: during the observation period an equal number of tokens enter and leave each place. A confrontation of our results extent requires to compare the modeling power of (max,+) automata with the class of Petri nets considered in [5]. Unfortunately, we only have partial answers to this question. It is shown in [12] that the behavior of any safe[6] timed Petri net can be expressed by a so-called heap automaton which is a special type of (max,+) automaton. We have to admit that the class of systems considered this way is restricted. Nevertheless, note that there is no assumption on the structure of the net (compared with extended free-choice conflicts), all the logical feasible choices for conflicts are considered (preselection policy) and there is no balance assumption. In addition, we are convinced that a wider class of Petri nets can be equivalently modeled by (max,+) automata. First, safeness

[6] At most one token can be in a place at any time.

is only a sufficient condition in the result of [12]. This has motivated an attempt towards the modeling of bounded[7] timed Petri nets which is presented in [21], together with compositions techniques making it possible to build (max,+) automata models in a modular manner. Secondly, automata with weights in real-interval based semiring (instead of (max,+) semiring) are introduced in [18], and the relations with time Petri nets (more precisely nets obtained by means of synchronous products of T-time state machines) are shown.

4 Representations for non-autonomous (max,+) automata

Discrete event systems which are modeled by (max,+) automata with the usual representation (4) or with the representations introduced in Propositions 1-5, can be considered as *autonomous* in the sense that their dynamic evolution isn't subjected to exogenous influences. To the best of our knowledge, it is still an open problem to define inputs for the usual representation (4) in order to model external influences. This section investigates how to enrich the new representations for (max,+) automata such that exogenous inputs can be taken into account. For this first attempt, we restrict our attention on the representation for the worst-case behavior in terms of sequence durations defined in Proposition 1. It appears in §4.1 that the definition of inputs then leads to a model for (max,+) automata which is similar to a state equation for nonautonomous systems. The next subsection tends to demonstrate that such a representation can be exploited for the control of (max,+) automata. In particular, a well-known open-loop control law for (max,+) linear systems [2, 20] is here adapted to (max,+) automata.

4.1 Definition of exogenous inputs for the worst-case behavior in terms of sequence duration

For a system modeled by a (max,+) automaton, we consider external influences acting on *controllable* state-transitions.

Definition 3 A state-transition is said to be controllable if the validation of the event implied in the transition can be delayed.

It is assumed that events-occurrences can be observed and counted[8]. In other words, at any time it is possible to know the number of transitions having occurred until then.

Let $\overline{A}$ be defined by (9) with $\mathcal{D}$ corresponding to $\overline{\mathbb{R}}_{\max}$, and $\overline{v}(n) \in \overline{\mathbb{R}}_{\max}^{|H|\times 1}$, $n \in \mathbb{N}$, a vector modeling the external influence. We define inductively the

[7] At most a bounded number of tokens can be in a place at any time.

[8] The case of unobservable events should be considered in future work.

following sequence of vectors $\overline{x}_c(n) \in \overline{\mathbb{R}}_{\max}^{|H|\times 1}$ for $n \in \mathbb{N}$ by:

$$\begin{cases} \overline{x}_c(0) = \varepsilon, \\ \overline{x}_c(n+1) = \overline{A}\overline{x}_c(n) \oplus \overline{B}\overline{v}(n) \end{cases} . \tag{15}$$

with $\overline{B} \in \overline{\mathbb{R}}_{\max}^{|H|\times|H|}$ given by

$$\overline{B}_{jk} = \begin{cases} e & \text{if } j = k \text{ and } j \text{ is a controllable transition,} \\ \varepsilon & \text{otherwise.} \end{cases}$$

Let us note that if a transition j is uncontrollable, then $B_{jk} = \varepsilon$ for every $k \in H$ and

$$\overline{x}_c(n+1)_j = [\overline{A}\overline{x}_c(n)]_j \ , \tag{16}$$

which means that input $\overline{v}(n)_j$ has no influence on signal $\overline{x}_c(n+1)$. In the inverse case,

$$\overline{x}_c(n+1)_j = [\overline{A}\overline{x}_c(n)]_j \oplus \overline{v}(n)_j \ , \tag{17}$$

and signal $\overline{x}_c(n+1)$ then depends on $\overline{v}(n)_j$.

Proposition 6 *For every $j \in H$, let variable $\overline{v}(n)_j$ denote the date at which transition j is authorized once n state-transitions have been observed (assuming $\overline{v}(0) = \alpha$), then $\overline{x}_c(n+1)_j$ corresponds to the maximum value of the set containing the dates from which transition j can occur consecutively to n state-transitions.*

Proof On one hand, it is easy to see that $\overline{x}_c(n+1) \geq \overline{x}(n+1)$ where $\overline{x}(n+1)_j$ (defined in Proposition 1 without considering exogenous influences) is the maximum completion date among sequences of length n and at the conclusion of which j can occur.
On the other hand, we have $\overline{x}_c(n+1) \geq \overline{B}\overline{v}(n)$ in which $[\overline{B}\overline{v}(n)]_j$ corresponds to the date from which transition j is authorized once n state-transitions have been observed.
From these two observations, we conclude that $\overline{x}_c(n+1)_j$ corresponds to the maximum among the dates from which transition j can occur consecutively to n state-transitions.

Remark 7 For conventional (boolean) automata, the influence of a control (supervision) is modeled by means of a (synchronous) product with the language [26]. The situation is comparable for the control of (max,+) automata based on their behavioral representation [17]. In these approaches, the control isn't considered as an exogenous input acting on the state of the system (i.e. an additive term in the state equation as in Eq. (15)), and this makes it difficult to adapt the numerous control results developed for (max,+) linear systems see for example [25, 8, 24, 1]).

Remark 8 With the approach described above, an input can influence a state-transition while assuming the worst-case for the past evolution (i.e. by considering that sequences with maximum completion dates have occurred). For example, if we consider a system with several possible schedules (and the schedule-choice is unknown), then the input is going to apply as if the worst schedule (with the largest makespan) had been selected until then. Afterwards, this representation makes it possible to compute control laws guaranteeing objectives for the system (typically some deadlines for the completion of given length sequences) whatever is the schedule applied in practice. It is a significant difference with the control law proposed in [17]: it is based on the behavioral representation and the control depends both on the current state and on the word denoting the sequence having lead to this state. Pursuing the previous illustration, the control is then specified according to the chosen schedule.
It is also worth mentioning that $\overline{v}(n)_j$ can be chosen to be equal to $\top = +\infty$, which means that a controllable transition j can be authorized at the end of infinitely large delay. This is equivalent to forbid this transition and logic control of the system can be tackled this way.

Note that recurrence (15) leads to

$$\overline{x}_c(n+1) = \bigoplus_{j=0}^{n} \overline{A}^j \overline{B}\overline{v}(n-j). \tag{18}$$

4.2 A control approach

The representation defined by Eqs. (15) and (18) is used to deal with the following control problem:

Find the *greatest* trajectory $\{\overline{v}(n)\}_{n\in\mathbb{N}}$ such that

$$\bigoplus_{j=0}^{n} \overline{A}^j \overline{B}\overline{v}(n-j) \preceq z(n) \tag{19}$$

where $[z(n)]_j$ corresponds to a deadline for the sequences of length n preceding transition j. In other words, we specify that all sequences of length n prefixing transition j have to be completed at date $[z(n)]_j$.

The motivations are as follows.

- The inequality (19) means that the input $\overline{v}$ is such that $\overline{x}_c(n+1)_j \leq z(n)_j$ for all $n \in \mathbb{N}$, $j \in H$. This means that the maximum among the dates from which transition j can occur consecutively to n state-transitions is less than the deadline $z(n)_j$.
- Seeking the greatest input $\{\overline{v}(n)\}_{n\in\mathbb{N}}$ means that we want to authorize the transitions at the latest.

In this sense, the input satisfies the so-called *just-in-time* criterion.

Remark 9 We can select z such that $z(n)_j = \top = +\infty$ (infinite deadline) for given transition j and for n greater than a given value N. This leads to a control input which delays indefinitely (i.e. forbids) controllable transition j. This can be useful

– for logic control purpose,
– or if $\{z(n)\}_{n\in\mathbb{N}}$ is partially known. In this case, it can be considered that beyond horizon N the deadlines are $+\infty$. When we deal with the control of a manufacturing system, this may mean that either only a finite number (N) of products must be delivered, or, the production schedule is not known beyond the N-th product to deliver. In both cases, the missing, or unknown future orders, are then supposed not to constrain the current running of the manufacturing system.

The following Propositions 7 and 8 give a positive answer to control problem (19), and can be seen as adaptations of results for (max,+) linear systems [2, §5.6], [20]. The results are based on residuation theory. Let us recall that

a mapping $f : \mathcal{C} \mapsto \mathcal{D}$, where $\mathcal{C}$ and $\mathcal{D}$ are ordered sets, is *residuated* if for all $y \in \mathcal{D}$, the least upper bound of subset $\{x \in \mathcal{C} \mid f(x) \preceq y\}$ exists and belongs to this subset. It is then denoted $f^\sharp(y)$. Mapping $f^\sharp : \mathcal{D} \mapsto \mathcal{C}$ is called the *residual* of f. In a complete dioid, mapping $x \mapsto a \otimes x$ is residuated; its residual is denoted $y \mapsto a \backslash y$. Some standard formulæ [2, §4.4] will be useful later on:

$$a \otimes (a \backslash x) \preceq x \ , \qquad (f.1)$$
$$a \backslash (x \wedge y) = (a \backslash x) \wedge (a \backslash y) \ , \qquad (f.2)$$
$$(a \otimes b) \backslash x = b \backslash (a \backslash x) \ . \qquad (f.3)$$

Proposition 7 *The trajectory* $\{\overline{v}_{opt}(n)\}_{n\in\mathbb{N}}$ *defined by*

$$n \in \mathbb{N},\ \overline{v}_{opt}(n) = \bigwedge_{i\geq 0} (\overline{A}^i \overline{B}) \backslash z(n+i) \qquad (20)$$

is the greatest solution of (19).

Proof 6 *Let* $\{\overline{v}(n)\}_{n\in\mathbb{N}}$ *be a solution of* (19)*. We have*

$$\begin{aligned}
&\forall n \in \mathbb{N}, && \bigoplus_{j=0}^{n} \overline{A}^j \overline{B}\,\overline{v}(n-j) \preceq z(n),\\
\Leftrightarrow\ &\forall n \in \mathbb{N}, \forall j \ s.t.\ j \geq 0 \ and\ j \leq n, && \overline{A}^j \overline{B}\,\overline{v}(n-j) \preceq z(n),\\
\Leftrightarrow\ &\forall n \in \mathbb{N}, \forall j \ s.t.\ j \geq 0 \ and\ j \leq n, && \overline{v}(n-j) \preceq (\overline{A}^j \overline{B}) \backslash z(n),\\
\Leftrightarrow\ &\forall n' \in \mathbb{N}, \forall j \ s.t.\ j \geq 0 \ and\ n' \geq 0, && \overline{v}(n') \preceq (\overline{A}^j \overline{B}) \backslash z(n'+j),\\
& && (with\ n' = n-j)\\
\Leftrightarrow\ &\forall n' \in \mathbb{N}, \forall j \geq 0, && \overline{v}(n') \preceq (\overline{A}^j \overline{B}) \backslash z(n'+j),\\
\Leftrightarrow\ &\forall n' \in \mathbb{N}, && \overline{v}(n') \preceq \bigwedge_{j\geq 0} (\overline{A}^j \overline{B}) \backslash z(n'+j) = \overline{v}_{opt}(n').
\end{aligned}$$

In the following proposition, we show that $\overline{v}_{opt}$ is a solution of a system of recurrent equations which proceed backwards.

Proposition 8 *The trajectory $\{\overline{v}_{opt}(n)\}_{n\in\mathbb{N}}$ defined by (20) is the greatest solution $\overline{v}$ of:*

$$\begin{cases} \xi(n) = \overline{A}\circ\backslash\xi(n+1) \wedge z(n) \\ \overline{v}(n) = \overline{B}\circ\backslash\xi(n) \end{cases} \tag{21}$$

Proof 7 *Let us first show that $\{\overline{v}_{opt}(n)\}_{n\in\mathbb{N}}$ is a solution of* (21).
We define $\{\xi_{opt}(n)\}_{n\in\mathbb{N}}$ such that $\overline{v}_{opt}(n) = \overline{B}\circ\backslash\xi_{opt}(n)$, and then:

$$\xi_{opt}(n) = \bigwedge_{i\geq 0} \overline{A}^i \circ\backslash z(n+i) \quad \textit{(according to (f.3))}.$$

We then check that $\{\xi_{opt}(n)\}_{n\in\mathbb{N}}$ is a solution of the first equation in (21) :

$$\begin{aligned} \left[\overline{A}\circ\backslash\xi_{opt}(n+1)\right] \wedge z(n) &= \left[\overline{A}\circ\backslash(\bigwedge_{i\geq 0} \overline{A}^i \circ\backslash z(n+i+1))\right] \wedge z(n), \\ &= \bigwedge_{i\geq 0} \left[\overline{A}\circ\backslash(\overline{A}^i \circ\backslash z(n+i+1))\right] \wedge z(n), \; (using\ (f.2)) \\ &= \bigwedge_{i\geq 0} \left[(\overline{A}^{i+1} \circ\backslash z(n+i+1))\right] \wedge z(n), \;\; (using\ (f.3)) \\ &= \bigwedge_{i\geq 0} (\overline{A}^i \circ\backslash z(n+i)), \\ &= \xi_{opt}(n). \end{aligned}$$

Let us now show that any solution of (21) *is smaller than $\{\overline{v}_{opt}(n)\}_{n\in\mathbb{N}}$. Let $\{\overline{v}(n)\}_{n\in\mathbb{N}}$ be a solution of* (21)*, we have*

$$\begin{aligned} \overline{v}(n) &= \overline{B}\circ\backslash\Big(\overline{A}\circ\backslash\xi(n+1) \wedge z(n)\Big), \\ &= \overline{B}\circ\backslash\Big(\overline{A}\circ\backslash\left[\overline{A}\circ\backslash\xi(n+2) \wedge z(n+1)\right] \wedge z(n)\Big), \\ &= \overline{B}\circ\backslash\Big(\overline{A}^2\circ\backslash\xi(n+2) \wedge \overline{A}\circ\backslash z(n+1) \wedge z(n)\Big), \\ &\;\;\vdots \\ &= \overline{A}^i\overline{B}\circ\backslash\xi(n+i) \wedge \bigwedge_{j=0}^{i-1} \overline{A}^j\overline{B}\circ\backslash z(n+j), \qquad (\forall i > 0) \\ &\preceq \bigwedge_{j\geq 0} \overline{A}^j\overline{B}\circ\backslash z(n+j), \\ &\preceq \overline{v}_{opt}(n). \end{aligned}$$

5 Applications to a jobshop example

We consider a jobshop system (inspired by the example in [12])with two resources $\mathcal{R}_1$, $\mathcal{R}_2$ processing two jobs types $\mathcal{J}_1$, $\mathcal{J}_2$. There are six elementary tasks a, b, c, d, e and f whose durations are respectively 2, 2, 2, 1, 1 and 3. The production sequence for job $\mathcal{J}_1$ is abc, which means that the elementary tasks a, b and c have to be performed in this order in order to complete one job J_1. The production sequence for job $\mathcal{J}_2$ is def. Resources process tasks one by one. Tasks a and d (resp. c and f) are processed using resource $\mathcal{R}_1$ (resp. $\mathcal{R}_2$). Both resources $\mathcal{R}_1$ and $\mathcal{R}_2$ are required for tasks b and e. In addition, two occurrences of a same job cannot be processed simultaneously: for example in sequence $\mathcal{J}_1\mathcal{J}_1$ the first job must be completed before the second can start (but in the case of sequence $\mathcal{J}_1\mathcal{J}_2$ the second job can start before the first one is completed). We consider all the possible sequences with the earliest functioning rule (tasks are completed as soon as possible). We also assume that the system starts operating at date 0.

The system can be modeled by timed Petri net in figure 4 in which:

- timings are associated with transitions (notation a/τ means that transition labeled a has τ for firing time),
- a *preselection policy* is used to decide on which transition is to fire when a place has several output transitions (to be able to code all the possible choices),
- a token from the initial marking is supposed to have arrived in the Petri net at time instant 0.

This Petri net is proposed only for illustration (in order to clarify the explanations) since the jobshop is going to be studied by means of a (max,+) automaton model, in this case the automaton in figure 5. Let us point out that, to the best of our knowledge, there are only partial answers to the question on how to transform a Petri net into a (max,+) automaton (and vice-versa). As mentioned in Subsection 3.4, the approach from [12] associates a (max,+) automaton to any safe timed Petri net. This (max,+) automaton is generally non-deterministic and may admit a language which is larger than the language of the Petri (i.e., it recognizes sequences which aren't possible firing sequences). A procedure based on completion of heap automata in [11] can be used to build a deterministic automaton corresponding to the Petri for the jobshop. The recent contribution [22] proposes a recursive procedure which builds a deterministic (max,+) automaton equivalent to a safe timed Petri net. The equivalence corresponds to the facts that the automaton and the Petri net have the same language, and that the completion date of a firing sequence in the Petri net is the same as the one of the corresponding state-transitions sequence in the (max,+) automaton. In addition, it is shown that this procedure terminates if the oriented path between any two transitions contains at most one "conflict-place" (with more than one output transition). This condition is satisfied by the Petri net in figure 4 and this procedure[9] has been used to obtain (max,+) automaton in figure 5.

Let us first illustrate how the results from section 3 can be used to study the performance of the jobshop.

1. We can apply the representation introduced in Prop. 1 to find out the maximum completion date for a given number of jobs. As each of the jobs

[9] Please note that the complexity of this procedure remains to be shown.

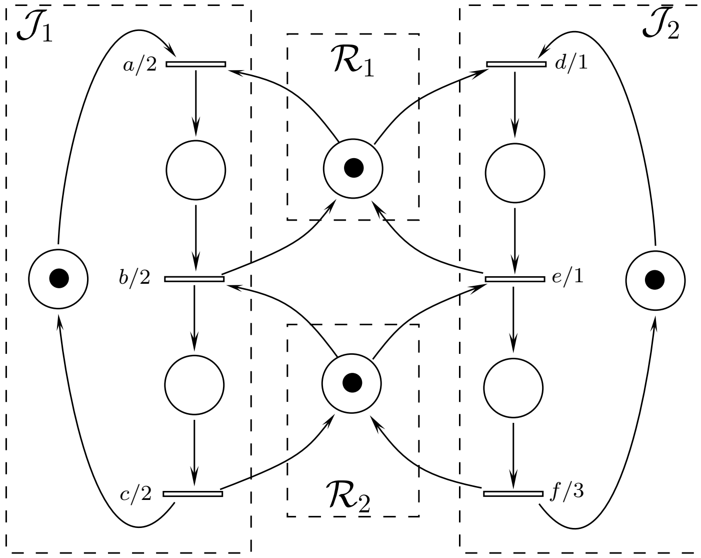

Fig. 4 Safe jobshop represented as a Petri net.

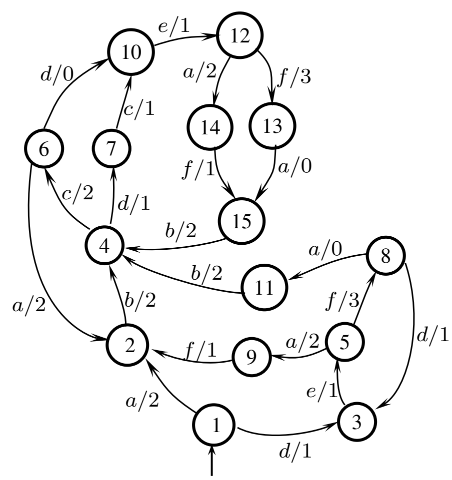

Fig. 5 Safe jobshop represented as a (max,+) automaton..

J_1 and J_2 consists of three tasks, n jobs are represented by $3n$ transitions in the automaton. In addition, we are interested with sequences leading to states 6, 8 and 13 since these are sequences for which both $\mathcal{J}_1$ and $\mathcal{J}_2$ are completed. For example, we obtain for $n = 10$:

$$\bigoplus_{j \in H_6 \cup H_8 \cup H_{13}} [\overline{x}(31)]_j = 60,$$

which means that the maximum execution time for 10 jobs is 60 units of time. This corresponds to the makespan of the worst schedule, i.e. 10 jobs $\mathcal{J}_1$.

2. We can apply the representation introduced in Prop. 3 (the automaton is strongly unambiguous) to evaluate the minimum completion date for n jobs. As for the previous item, we are interested by sequences of length $3n$ leading to states 6, 8 and 13. In $\overline{\mathbb{R}}_{\min}$, we obtain for $n = 10$:

$$\bigoplus_{j \in H_6 \cup H_8 \cup H_{13}} [\underline{x}(31)]_j = 41$$

which means that the minimum makespan for 10 jobs is equal to 41 units of time. This corresponds to the makespan of the best schedule, i.e. $\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1$ whose timing is depicted in figure 6.
3. The representation introduced in Prop. 4 can be used to evaluate the maximum number of jobs that can be performed until a date t. We focus on sequences leading to states 6, 8 and 13 and we obtain for $t = 41$:

$$\bigoplus_{j \in H_6 \cup H_8 \cup H_{13}} [\overline{z}(41)]_j = 30$$

This result is coherent with the one obtained at the previous item, since it reveals that 10 jobs can be completed before time instant 41.

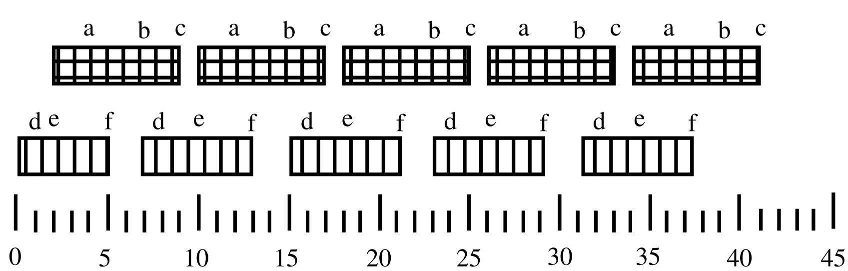

Fig. 6 Timing for schedule $\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1\mathcal{J}_2\mathcal{J}_1$.

Let us now illustrate how results from Section 4 (Prop. 7 and 8) can contribute to the control of this system.
We consider the following reference input z:

$$[z(n)]_j = \begin{cases} 70 & \text{if } j \in H_6 \cup H_8 \cup H_{13} \text{ and } 0 \le n \le 30, \\ T & \text{otherwise.} \end{cases}$$

The control objective can then be interpreted as follows: delay the system as much as possible while guaranteeing that 10 jobs are completed at t=70. As discussed in Remark 9, value $T = +\infty$ is used to code unknown deadlines or undesired occurrences.

- We first assume that all the transitions are controllable, that is $\overline{B}$ is equal to identity matrix I_n. Using recurrence (21) we obtain :

  $$\overline{v}_{opt}(0)_{(1,a,2)} = 10.$$

  This means that if the first job is $\mathcal{J}_1$, then its processing is authorized from date 10. In this case, even if the worst schedule is applied (that is ten times $\mathcal{J}_1$ with 60 as completion time), 10 jobs are completed at the latest at date 70.
  We also obtain,

  $$\overline{v}_{opt}(0)_{(1,d,3)} = 13$$

  which gives the starting date in the case where the first job is $\mathcal{J}_2$. Like this, even if 9 jobs $\mathcal{J}_1$ are consecutively processed (with a processing time equal to 57), then 10 jobs are also completed at the latest at date 70.

- Let us now consider that all transitions with events a, b and c (i.e. tasks for job $\mathcal{J}_1$) are uncontrollable. In that case, matrix B isn't equal to identity matrix I_n, since coefficients corresponding to uncontrollable transitions are equal to $\varepsilon = -\infty$.

  The schedules with only jobs $\mathcal{J}_1$ are uncontrollable, in the sense that the processing cannot be delayed. For example, we obtain:

  $$\overline{v}_{opt}(n)_{(1,a,2)} = T, \forall n \in \mathbb{N},$$

  knowing that $B_{j,(1,a,2)} = \varepsilon, \forall j$ (transition $(1, a, 2)$ is uncontrollable) and so $\overline{v}_{opt}(n)_{(1,a,2)}$ has no influence.

  Any other schedule is controllable in the sense that one occurrence of job $\mathcal{J}_2$ (with controllable associated transitions) is sufficient to meet the control objective. For example, we obtain

  $$\overline{v}_{opt}(0)_{(1,d,3)} = 13,$$

  which is the latest processing time for $\mathcal{J}_2$ as first job so that 10 jobs are completed at the latest at date 70. We also get

  $$\overline{v}_{opt}(27)_{(9,f,2)} = 65,$$

  which means that the last transition of $\mathcal{J}_2$ as 9th job is delayed such that $\mathcal{J}_1$ as 10th job is completed at 70. In fact, last transitions to complete 10 jobs are then $9 \xrightarrow{f} 2 \xrightarrow{b} 4 \xrightarrow{c} 6$ with duration equal to 5 (see figure 7).

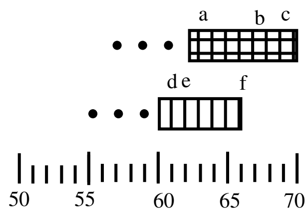

Fig. 7 Timing for a schedule with $\mathcal{J}_2\mathcal{J}_1$ as 9th and 10th jobs and with the just-in-time control.

6 Conclusion

We have proposed new representations for (max,+) automata in order to describe their extremal behaviors. We have shown that these representations can be applied to efficiently evaluate performances of systems. In future work, results from the spectral theory of (max,+) matrices should be exploited to enrich our approach. Using representation of the worst-case behavior in terms of sequence durations, we have illustrated how exogenous inputs can be taken into account. Taking advantage from the fact that the model is then strictly similar to state equations of (max,+) linear systems, a control approach has been sketched out. It would be interesting to study the control of the other extremal behaviors. It is also envisaged to adapt other control laws developed for (max,+) and (min,+) linear systems.

References

1. S. Amari, I. Demongodin, J.-J. Loiseau, and C. Martinez. Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transaction on Automatic Control*, 57(8), 2012.
2. F. Baccelli, G. Cohen, G.J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity*. Wiley, 1992.
3. E. Badouel, A. Bouillard, P. Darondeau, and J. Komenda. Residuation of tropical series: rationality issues. In *joint 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'11)*, pages 3855–39861, Orlando, USA, 2011.
4. M.-P. Béal, E. Czeizler, J. Kari, and D. Perrin. Unambiguous automata. *Mathematics in Computer Science*, 1(4):625–638, 2008.
5. S. Bernardi and J. Campos. Computation of performance bounds for real-time systems using time petri nets. *IEEE Transactions on Industrial Informatics*, 5(2):168–180, 2009.
6. P. Butkovic, R.A. Cuninghame-Green, and S. Gaubert. Reducible spectral theory with applications to the robustness of matrices in max-algebra. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1412–1431, 2010.

7. J. Cochet-Terrasson, S. Gaubert, G. Cohen, M. Mc Gettrick, and J.-P. Quadrat. Numerical compuatation of spectral elements in max-plus algebra. In *IFAC Conference System Structure and Control*, pages 667–674, Nantes, France, 1998.
8. B. De Schutter and T.J.J. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049 – 1056, 2001.
9. M. Droste, W. Kuich, and H. Vogler (Eds.). *Handbook of Weighted Automata*. Springer, 2009.
10. S. Gaubert. Performance Evaluation of (max,+) Automata. *IEEE Transactions on Automatic Control*, 40(12):2014–2025, 1995.
11. S. Gaubert and J. Mairesse. Asymptotic analysis of heaps of pieces and application to timed petri nets. In *Proceedings of the 8th International Workshop on Petri Nets and Performance Models*, pages 158–169, Zaragoza, Spain, 1999.
12. S. Gaubert and J. Mairesse. Modeling and Analysis of Timed Petri Nets using Heaps of Pieces. *IEEE Transactions on Automatic Control*, 44(4):683–698, 1999.
13. B. Heidergott, G.J. Olsder, and J. van der Woude. *Max Plus at work*. Princeton, 2006.
14. L. Houssin. Cyclic jobshop problem and (max,plus) algebra. In *18th IFAC World Congress*, pages 2717–2721, Milan, Italy, 2011.
15. D. Kirsten and S. Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *Symposium on Theoretical Aspects of Computer Science*, pages 589–600, 2009.
16. I. Klimann, S. Lombardy, J. Mairesse, and C. Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327:349–373, 2004.
17. J. Komenda, S. Lahaye, and J.-L. Boimond. Supervisory Control of (max,+) Automata: A Behavioral Approach. *Discrete Event Dynamic Systems*, 19(4):525–549, 2009.
18. J. Komenda, S. Lahaye, and J.-L. Boimond. Synchronous Composition of Interval Weighted Automata. In *International Workshop on Discrete Event Systems (WODES 2010)*, Berlin, Allemagne, August 2010.
19. D. Krob. The equality problem for rational series with multiplicities in the tropical semirings is undecidable. *International Journal of Algebra and Computation*, pages 405–425, 1994.
20. S. Lahaye, J.-L. Boimond, and L. Hardouin. Optimal Control of (Min,+) Linear Time-Varying Systems. In *8th International Workshop on Petri Nets and Performance Models (PNPM 1999)*, pages 170–178, 1999.
21. S. Lahaye, J. Komenda, and J.-L. Boimond. Compositions of (max,+) automata. In *International Workshop on Discrete Event Systems (WODES 2012)*, Guadalajara, 2012.
22. S. Lahaye, J. Komenda, and J.-L. Boimond. Modeling of timed Petri nets using deterministic (max,+) automata. In *Submitted to 52nd IEEE Conference on Decision and Control*, 2013.

23. S. Lombardy and J. Sakarovitch. Sequential ? *Theoretical Computer Science*, 359(1-2):224–244, 2006.
24. C.A. Maia, L. Hardouin, R. Santos Mendes, and B. Cottenceau. Optimal Closed-Loop Control for Timed Event Graphs in Dioid. *IEEE Transaction on Automatic Control*, vol. 48:2284–2287, 2003.
25. E. Menguy, J.-L. Boimond, L. Hardouin, and J.-L. Ferrier. Just in time control of linear systems in dioid: cases of an update of reference input and uncontrollable input. *IEEE Transaction on Automatic Control*, vol. 45(11):2155–2159, 2000.
26. P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of The IEEE*, 77:81–98, 1989.
27. R. Su, J.H. van Schuppen, and J.E. Rooda. The synthesis of time optimal supervisors by using heaps-of-pieces. *IEEE Transactions on Automatic Control*, 57(1):105–118, 2012.
28. R. Su and G.J. Woeginger. String execution time for finite languages: Max is easy, min is hard. *Automatica*, 47(10):2326–2329, 2011.
29. W. Weyerman and S. Warnick. Monotonically improving error bounds for a sequence of approximations for makespan minimization of batch manufacturing systems. In *IEEE Conference on Decision and Control*, pages 5288–5293, 2007.