Implementation of procedures for optimal control of timed event graphs with resource sharing [1]

Davide Zorzenon * **Germano Schafaschek** * **Dominik Tirpák** *
Soraia Moradi * **Laurent Hardouin** ** **Jörg Raisch** *,***

* *Control Systems Group, Technische Universität Berlin, Germany (e-mail: [zorzenon,schafaschek,moradi,raisch]@control.tu-berlin.de, dominik.tirpak@campus.tu-berlin.de)*
** *Laboratoire Angevin de Recherche en Ingénierie des Systèmes, Polytech Angers, Université d'Angers, France (e-mail: laurent.hardouin@univ-angers.fr)*
*** *Science of Intelligence, Research Cluster of Excellence, Berlin, Germany*

Abstract: It is a well-known fact that the dynamics of timed event graphs (TEGs), a subclass of timed Petri nets able to model delay and synchronization phenomena, admits linear representation in dioids of formal power series. In order to model and control systems presenting resource sharing phenomena, it is useful to extend the usual set of operations between series by including the Hadamard product, its residual, and its dual residual. Until now however, the characterization of the largest set of series for which the dual residual of the Hadamard product is defined was still incomplete. Rather than being a mere theoretical dilemma, this open problem delayed the development of reliable algorithms for optimal control. In this paper, we provide the solution to the problem, and discuss the implementation of procedures for computing the Hadamard product and affine operations. Such procedures have been recently implemented on top of the C++ library ETVO. Tests are conducted to evaluate their performance in solving optimal-control problems on TEGs with resource sharing and output-reference update.

Keywords: Petri nets, timed event graphs, resource sharing, optimal control

1. INTRODUCTION

Resource-sharing phenomena occur in a variety of applications. For example, in manufacturing, the same machine may serve parallel production lines but can only operate on a limited number of items at a time. Resource sharing can be modeled by general (timed) Petri nets but not by timed event graphs (TEGs). Nonetheless, TEGs are appealing because, as opposed to general Petri nets, their dynamics can be represented by linear equations in dioids, which has made it possible to develop an elegant theory for their analysis and control (see Baccelli et al. [1992], Hardouin et al. [2018]). The simplicity of TEG models in dioids has motivated several approaches to the extension of this theory towards encompassing TEGs with resource sharing, i.e., systems where each user competing for the resources is modeled by a TEG (e.g. Correia et al. [2009], Addad et al. [2010], Moradi et al. [2017]).

In Schafaschek et al. [2020], some of the authors of the present paper studied the optimal control of TEGs with resource sharing and output-reference update, namely, systems consisting of a number of TEGs that share resources and whose output-reference signals can change over time. Optimal control here is to be understood in terms of the just-in-time criterion, which aims at generating the input events as late as possible while guaranteeing that the output ones are never later than specified by the reference. Their approach is based on some operations on formal power series, including the Hadamard product, its residual, and its dual residual.

In this paper, we focus on the implementation of such operations. After recalling some preliminaries in Section 2, in Section 3 we give a characterization of the largest set for which the dual residual of the Hadamard product is defined. This result, which improves Proposition 3 of Hardouin et al. [2008], is of both theoretical and practical relevance; indeed, it allows to clarify the range of applicability of the operation and to simplify the implementation of reliable optimal-control procedures. In the same section, we present the algorithms for computing the Hadamard product and its residuals on monomials, polynomials, and periodic formal power series. Such algorithms have been implemented in C++, extending the set of routines provided by the library ETVO ((Event|Time)-Variant Operators), which is capable of representing and manipulating several classes of formal power series in dioids (Cottenceau et al. [2020]). The usefulness of the operations for solving optimal-control problems in discrete

[1] Support from Deutsche Forschungsgemeinschaft (DFG) via grant RA 516/14-1 and under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135 is gratefully acknowledged.

event systems is discussed in Section 4. Finally, the performance of our implementation of the Hadamard product and affine operations is evaluated in Section 5, where we report the computational time for solving optimal-control problems with different number of TEGs and resources, with and without output-reference update.

2. PRELIMINARIES

In this section, we recall some preliminary concepts from dioid and residuation theory. We refer to Baccelli et al. [1992], Hardouin et al. [2018] for an in-depth presentation.

2.1 Dioid theory

A dioid (or idempotent semiring) $(\mathcal{D}, \oplus, \otimes)$ is a set $\mathcal{D}$ equipped with two binary operations, $\oplus$ and $\otimes$, called respectively addition and multiplication, having the following properties. Addition is commutative, associative, idempotent (i.e., $a \oplus a = a \ \forall a \in \mathcal{D}$), and admits neutral (or zero) element ε; multiplication is associative, distributes over addition, admits neutral (or unit) element e, and ε is absorbing for multiplication (i.e., $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon \ \forall a \in \mathcal{D}$). As it is common in standard algebra, the multiplication symbol "$\otimes$" will be often omitted. Operation $\oplus$ induces an order relation $\preceq$, defined by $a \preceq b \ \Leftrightarrow \ a \oplus b = b$.

A dioid is complete if it is closed for infinite sums and if multiplication distributes over infinite sums, i.e., $a \otimes \left(\bigoplus_{x \in \mathcal{X}} x\right) = \left(\bigoplus_{x \in \mathcal{X}} a \otimes x\right)$, and $\left(\bigoplus_{x \in \mathcal{X}} x\right) \otimes a = \left(\bigoplus_{x \in \mathcal{X}} x \otimes a\right)$ for all $a \in \mathcal{D}$, $\mathcal{X} \subseteq \mathcal{D}$. Let $(\mathcal{D}, \oplus, \otimes)$ be a complete dioid. Its top element is defined by $\top = \bigoplus_{x \in \mathcal{D}} x$. The greatest lower bound $\wedge$ is defined, for all $a, b \in \mathcal{D}$, by $a \wedge b = \bigoplus_{\mathcal{D}_{ab}} x$, where $\mathcal{D}_{ab} = \{x \in \mathcal{D} \mid x \preceq a, \ x \preceq b\}$. Operation $\wedge$ is commutative, associative, idempotent, and admits $\top$ as neutral element. Moreover, in a complete dioid $(\mathcal{D}, \oplus, \otimes)$, the Kleene star operator * applied to $a \in \mathcal{D}$ yields $a^* = \bigoplus_{k \in \mathbb{Z}, k \geq 0} a^k$, where $a^0 = e$, and $a^{k+1} = a \otimes a^k$ for all $k \geq 0$.

Remark 1. The implicit equation $x = ax \oplus b$ over a complete dioid $\mathcal{D}$ admits $x = a^* b$ as least solution (see Baccelli et al. [1992]). $\diamond$

As in standard algebra, operations $\oplus$ and $\otimes$ can be extended to matrices as follows: for all $A, B \in \mathcal{D}^{m \times n}$ and $C \in \mathcal{D}^{n \times p}$, $A \oplus B \in \mathcal{D}^{m \times n}$, and $A \otimes C \in \mathcal{D}^{m \times p}$ are such that

$$(A \oplus B)_{ij} = A_{ij} \oplus B_{ij}, \quad (A \otimes C)_{ij} = \bigoplus_{k=1}^{n} A_{ik} \otimes C_{kj}.$$

If $(\mathcal{D}, \oplus, \otimes)$ is a complete dioid, then $(\mathcal{D}^{n \times n}, \oplus, \otimes)$, where $\oplus$ and $\otimes$ are extended as above, is also a complete dioid. Its zero (resp. top) element is the $n \times n$-matrix with all entries equal to ε (resp. $\top$), and its unit element is the $n \times n$-matrix with e's on the main diagonal and ε's elsewhere.

Example 2. An example of complete dioid is the set $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$, with the *minimum* operation as $\oplus$ and standard addition as $\otimes$. With this notation, the complete dioid $\overline{\mathbb{Z}}_{\min} := (\overline{\mathbb{Z}}, \oplus, \otimes)$ is called the *min-plus algebra.* In $\overline{\mathbb{Z}}_{\min}$, $\varepsilon = +\infty$, $e = 0$, $\top = -\infty$, $\wedge$ corresponds to the *maximum* operation, and $\preceq$ corresponds to the standard $\geq$; this means that the order $\preceq$ is reversed with respect to the conventional one (e.g., $5 \preceq 2$). The dual dioid of $\overline{\mathbb{Z}}_{\min}$, denoted $\overline{\mathbb{Z}}_{\max}$, corresponds to the set $\overline{\mathbb{Z}}$ with the *maximum* operation as $\oplus$ and standard addition as $\otimes$; observe that the order in $\overline{\mathbb{Z}}_{\max}$ coincides with the standard one. Due to the absorbing property of ε, the result of $-\infty \otimes +\infty = +\infty \otimes -\infty$ is different in $\overline{\mathbb{Z}}_{\min}$ and $\overline{\mathbb{Z}}_{\max}$.

A mapping $\Pi : \mathcal{D} \to \mathcal{C}$, where $(\mathcal{D}, \oplus, \otimes)$ and $(\mathcal{C}, \oplus, \otimes)$ are two dioids, is *isotone* or *non-decreasing* (resp. *antitone* or *non-increasing*) if $\forall a, b \in \mathcal{D}$, $a \preceq b \Rightarrow \Pi(a) \preceq \Pi(b)$ (resp. $\Pi(a) \succeq \Pi(b)$).

Example 3. Another example of complete dioid is the algebra of counters. Let $s : \overline{\mathbb{Z}}_{\max} \to \overline{\mathbb{Z}}_{\min}$, $t \mapsto s(t)$, be an antitone mapping such that[2] $s(-\infty) = -\infty$ and $s(+\infty) = +\infty$. (Note that, due to the reverted order of $\overline{\mathbb{Z}}_{\min}$, such mappings are non-decreasing in the standard sense.) This kind of mappings can be used to represent the cumulative number of firings $s(t)$ of a transitions in a TEG up to and including time t. The δ-*transform* of s, called *counter*, is the non-increasing formal power series in δ with coefficients $s(t)$ in $\overline{\mathbb{Z}}_{\min}$ and exponents t in $\overline{\mathbb{Z}}_{\max}$, defined by

$$s = \bigoplus_{t \in \overline{\mathbb{Z}}} s(t) \delta^t.$$

As no ambiguity will occur, we indicate both the mapping and its δ-transform by the same symbol. The set of counters, denoted Σ, equipped with operations $\oplus$ and $\otimes$ defined by

$$(s \oplus s')(t) = s(t) \oplus s'(t) \quad \forall t \in \mathbb{Z},$$
$$(s \otimes s')(t) = \bigoplus_{\tau \in \mathbb{Z}} s(\tau) \otimes s'(t - \tau) \quad \forall t \in \mathbb{Z},$$

is a complete dioid.

Since counters are non-increasing and such that $s(-\infty) = -\infty$, $s(+\infty) = +\infty$, we can represent them compactly by omitting terms $-\infty \delta^{-\infty}$, $+\infty \delta^{+\infty}$, and all terms $s(t)\delta^t$ such that $s(t) = s(t+1)$. For instance,

$$-\infty \delta^{-\infty} \oplus \bigoplus_{-\infty < t \leq 1} -2\delta^t \oplus \bigoplus_{2 \leq t \leq 5} 3\delta^t \oplus \bigoplus_{t \geq 6} +\infty \delta^t$$

will be simply denoted $-2\delta^1 \oplus 3\delta^5$. With this simplified notation, the zero, unit, and top element of $(\Sigma, \oplus, \otimes)$ can be written, respectively, as $s_\varepsilon = +\infty \delta^{-\infty}$, $s_e = e\delta^0$, and $s_\top = -\infty \delta^{+\infty}$. Note that, given two counters $s, s' \in \Sigma$, $s \preceq s' \Leftrightarrow s(t) \preceq s'(t)$ for all $t \in \mathbb{Z}$, and their greatest lower bound is given by

$$(s \wedge s')(t) = s(t) \wedge s'(t) \quad \forall t \in \mathbb{Z}.$$

For algorithmic reasons, it is convenient to distinguish three increasingly larger classes of counters: monomials, of the form $n\delta^t$, polynomials, of the form $\bigoplus_{i=1}^{m} n_i \delta^{t_i}$ with $m > 0$, and ultimately periodic series. Series of the third kind are all those that can be written as $s = p \oplus qr^*$, where $p = \bigoplus_{i=1}^{m} n_i \delta^{t_i}$ is the transient part of s, $q = \bigoplus_{i=1}^{l} N_i \delta^{T_i}$ is the periodic pattern of s, whose periodicity is described by the monomial $r = \nu \delta^\tau$. When s represents the cumulative firings of a transition in a TEG, the sequence of firings specified by q repeats every τ time units and after ν firings of the corresponding transition. The ratio ν/τ is called *throughput*, and represents the average number of firings of the transition per unit time

[2] The importance of the end-point conditions on s is explained in [Baccelli et al., 1992, Chapter 5].

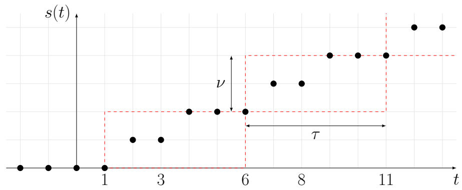

Figure 1. Series $(0\delta^1 \oplus 1\delta^3)(2\delta^5)^*$.

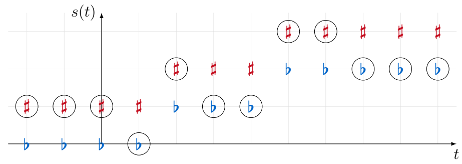

Figure 2. Series $\bar{x}$, $y \odot^\sharp a$, and $y \odot^\flat a$, indicated, respectively, by ◯'s, $\sharp$'s, and $\flat$'s, when $y = 1\delta^1 \oplus 3\delta^4 \oplus 5\delta^{+\infty}$ and $a = 0\delta^0 \oplus 1\delta^2 \oplus 2\delta^6 \oplus 3\delta^\infty$. Note that $\bar{x}$ is not a counter.

during the periodic regime. Due to the periodic behavior of ultimately periodic series, their representation in the form $p \oplus qr^*$ is not unique; however, every ultimately periodic series admits a unique canonical form, in which m (i.e., the number of monomials in the transient part p) is minimal. For example, the canonical form of series $0\delta^1 \oplus 1\delta^3 \oplus (2\delta^6 \oplus 3\delta^8)(2\delta^5)^*$, graphically represented in Figure 1, is $(0\delta^1 \oplus 1\delta^3)(2\delta^5)^*$.

2.2 Residuation theory

To solve control problems, it is often necessary to compute the inverse of a certain mapping. When the mapping is not invertible, sometimes it is possible to find the best under- and over-approximation of its inverse, called respectively its *residual* and *dual residual*.

Let $(\mathcal{D}, \oplus, \otimes)$ and $(\mathcal{C}, \oplus, \otimes)$ be two complete dioids, and $\Pi : \mathcal{D} \to \mathcal{C}$ an isotone mapping. The mapping Π is *residuated* (resp. *dually residuated*) if, for all $y \in \mathcal{C}$, set $\{x \in \mathcal{D} \mid f(x) \preceq y\}$ admits maximum (resp. $\{x \in \mathcal{D} \mid f(x) \succeq y\}$ admits minimum). In this case, the mapping $f^\sharp : \mathcal{C} \to \mathcal{D}$, $y \mapsto \bigoplus\{x \in \mathcal{D} \mid f(x) \preceq y\}$ (resp. $f^\flat : \mathcal{C} \to \mathcal{D}$, $y \mapsto \bigwedge\{x \in \mathcal{D} \mid f(x) \succeq y\}$) is called the residual (resp. dual residual) of f.

We recall the following result, which will be used later to prove that a certain mapping is dually residuated.

Proposition 4. Let $(\mathcal{D}, \oplus, \otimes)$ and $(\mathcal{C}, \oplus, \otimes)$ be two complete dioids. An isotone mapping $f : \mathcal{D} \to \mathcal{C}$ is dually residuated if and only if $f(\top_\mathcal{D}) = \top_\mathcal{C}$ and $f(\bigwedge_{x \in \mathcal{X}} x) = \bigwedge_{x \in \mathcal{X}} f(x)$ for all $\mathcal{X} \subseteq \mathcal{D}$.

Example 5. Given a complete dioid $(\mathcal{D}, \oplus, \otimes)$ and an element $a \in \mathcal{D}$, the mapping $L_a : \mathcal{D} \to \mathcal{D}$, $x \mapsto a \otimes x$ is residuated. Its residual, called *left division* by a, is denoted by $L_a^\sharp(y) = a \backslash\!\!\!\circ\, y$. Therefore, $a \backslash\!\!\!\circ\, y$ corresponds to the greatest solution x of the inequality $a \otimes x \preceq y$.

3. THE HADAMARD PRODUCT OF FORMAL POWER SERIES

3.1 Definition and residuals

In this subsection, we define the Hadamard product and its residuals; these operations are useful for solving optimal-control problems for some interesting classes of discrete event systems, as will be discussed in the next section.

The Hadamard product of two counters $s_1, s_2 \in \Sigma$, denoted by $s_1 \odot s_2$, is defined by

$$(s_1 \odot s_2)(t) = s_1(t) \otimes s_2(t) \quad \forall t \in \mathbb{Z}.$$

In standard algebra, it corresponds to the element-wise addition of the coefficients of the corresponding series.

Remark 6. Note that, given two counters $x, a \in \Sigma$, the series $\Pi_a(x) = a \odot x$ is always a counter. On the other hand, given $y \in \Sigma$, the same is not always true for the series $\bar{x}$ defined by $\bar{x}(t) = y(t) - a(t)\ \forall t \in \mathbb{Z}$. The greatest counter less than or equal to $\bar{x}$ (in the sense of the order in Σ) is given by the residual of Π_a, $\Pi_a^\sharp(y) = y \odot^\sharp a$, and the least counter greater than or equal to $\bar{x}$, when defined, is given by the dual residual of Π_a, $\Pi_a^\flat(y) = y \odot^\flat a$. The difference between series $\bar{x}$, $y \odot^\sharp a$, and $y \odot^\flat a$ is shown through an example in Figure 2. Let us now formally characterize the operations $\odot^\sharp$ and $\odot^\flat$.

The following results, proved in Hardouin et al. [2008], shows that the Hadamard product is residuated.

Proposition 7. The mapping $\Pi_a : \Sigma \to \Sigma$, $x \mapsto a \odot x$ is residuated for any $a \in \Sigma$. Its residual is denoted by $\Pi_a^\sharp(y) = y \odot^\sharp a$, and corresponds to the greatest counter $x \in \Sigma$ that satisfies $a \odot x \preceq y$.

In general, however, the mapping Π_a is not dually residuated. Indeed, if for a certain $t \in \mathbb{Z}$ $a(t) = +\infty$ and $y(t) \neq +\infty$, the least solution x of $a \odot x \succeq y$ is not defined, as inequality $a(t) \otimes x(t) \succeq y(t)$ (in standard algebra, $+\infty + x(t) \leq y(t)$) does not admit solutions. Another situation in which Π_a is not dually residuated is when there exists $t \in \mathbb{Z}$ such that $a(t) = -\infty$ and $y(t) \neq +\infty$; in this case, inequality $a(t) \otimes x(t) \succeq y(t)$ (in standard algebra, $-\infty + x(t) \leq y(t)$) admits infinitely many solutions, but the infimum of the solution set, $+\infty$, does not belong to it. In the following, we will prove that these two are the only cases in which the dual residual of Π_a is not defined; but for doing so we first need the following lemma.

Lemma 8. The set Σ of counters endowed with operation $\oplus$ as addition and $\odot$ as multiplication, $(\Sigma, \oplus, \odot)$, forms a complete dioid.

Proof. It is easy to verify that the following properties hold: $\odot$ is associative and distributes over operation $\oplus$, its neutral element is $e_\odot = e\delta^{+\infty}$, and the zero element $s_\varepsilon = +\infty\delta^{-\infty}$ is absorbing for it. This proves that $(\Sigma, \oplus, \odot)$ is a dioid. We already know that Σ is closed under infinite $\oplus$; thus, to prove that the dioid is complete we only need to show that $\odot$ distributes over infinite $\oplus$. Observe that, since $\oplus$ operates component-wise in Σ, and $\otimes$ distributes over infinite $\oplus$ in $\overline{\mathbb{Z}}_{\min}$, we have that, $\forall t \in \mathbb{Z}, s \in \Sigma, \mathcal{X} \subseteq \Sigma$,

$$\left(s \odot \bigoplus_{x \in \mathcal{X}} x\right)(t) = s(t) \otimes \left(\bigoplus_{x \in \mathcal{X}} x\right)(t) = s(t) \otimes \left(\bigoplus_{x \in \mathcal{X}} x(t)\right)$$
$$= \bigoplus_{x \in \mathcal{X}} (s(t) \otimes x(t)) = \bigoplus_{x \in \mathcal{X}} (s \odot x)(t),$$

which is the last property we had to show. ∎

We recall from Hardouin et al. [2008] that $\odot$ is also commutative and distributes over finite $\wedge$; now we can prove the main result of the paper.

Proposition 9. For $a \in \Sigma$, let $\mathcal{D}_a = \{x \in \Sigma \mid x = s_\varepsilon \text{ if } \exists t \in \mathbb{Z} \text{ with } a(t) = -\infty\}$, and $\mathcal{C}_a = \{y \in \Sigma \mid y(t) = +\infty \ \forall t \in \mathbb{Z} \text{ such that } a(t) \in \{-\infty, +\infty\}\}$. The mapping $\Pi_a : \mathcal{D}_a \to \mathcal{C}_a$, $x \mapsto a \odot x$ is dually residuated for any $a \in \Sigma$. Its dual residual is denoted by $\Pi_a^\flat(y) = y \odot^\flat a$, and corresponds to the least counter $x \in \Sigma$ that satisfies $a \odot x \succeq y$.

Proof. To prove the theorem using Proposition 4, we first need to show that $(\mathcal{D}_a, \oplus, \odot)$ and $(\mathcal{C}_a, \oplus, \odot)$ are complete dioids; to do so, we consider different cases. If there is no $t \in \mathbb{Z}$ with $a(t) = \pm\infty$, then $\mathcal{D}_a = \mathcal{C}_a = \Sigma$, which forms a complete dioid when endowed with $\oplus$ and $\odot$ (cf. Lemma 8). If a value $t \in \mathbb{Z}$ with $a(t) = -\infty$ exists, then $a = s_\top$ and $\mathcal{D}_a = \mathcal{C}_a = \{s_\varepsilon\}$; but $(\{s_\varepsilon\}, \oplus, \odot)$ is a (trivial) complete dioid, with $\varepsilon = e = \top = s_\varepsilon$. If there is no $t \in \mathbb{Z}$ with $a(t) = -\infty$, but there exists one with $a(t) = +\infty$, then $\mathcal{D}_a = \Sigma$, and, due to the non-increasingness of counters, $\mathcal{C}_a = \{y \in \Sigma \mid y(t) = +\infty \ \forall t \geq t_a\}$, where t_a is the least (in the standard sense) integer t such that $a(t) = +\infty$. Then, $(\mathcal{C}_a, \oplus, \odot)$ is also a complete dioid, with $\varepsilon_{\mathcal{C}_a} = s_\varepsilon$, $e_{\mathcal{C}_a} = 0\delta^{t_a - 1}$ and $\top_{\mathcal{C}_a} = -\infty\delta^{t_a-1}$.

Now that we have established that domain and codomain of Π_a are complete dioids, it needs to be shown that $\Pi_a(\top_{\mathcal{D}_a}) = \top_{\mathcal{C}_a}$: if a does not contain $\pm\infty$, $\top_{\mathcal{D}_a} = \top_{\mathcal{C}_a} = s_\top$, and $\Pi_a(s_\top) = s_\top$; if $a(t) = -\infty$ for some $t \in \mathbb{Z}$, $\top_{\mathcal{D}_a} = \top_{\mathcal{C}_a} = s_\varepsilon$ and $\Pi_a(s_\varepsilon) = s_\varepsilon$; if $a(t) = +\infty$ for all $t \geq t_a$, and for all $t \in \mathbb{Z}$ $a(t) \neq -\infty$, then $\top_{\mathcal{D}_a} = s_\top$, $\top_{\mathcal{C}_a} = -\infty\delta^{t_a-1}$, and $\Pi_a(s_\top) = a \odot s_\top = \top_{\mathcal{C}_a}$.

The final property to show is $\Pi_a\left(\bigwedge_{x\in\mathcal{X}} x\right) = \bigwedge_{x\in\mathcal{X}} \Pi_a(x)$ for all $\mathcal{X} \subseteq \mathcal{D}_a$. Note that Π_a is known to distribute over finite $\wedge$; in particular, if $a(t) = -\infty$ for some $t \in \mathbb{Z}$, then $\mathcal{X}$ is either $\emptyset$ or $\{s_\varepsilon\}$ and the property holds. Thus, we only need to consider the case in which $\mathcal{X}$ is infinite (and hence nonempty) and $a(t) \neq -\infty$ for all $t \in \mathbb{Z}$. For $t \in \mathbb{Z}$ such that $a(t) \notin \{-\infty, +\infty\}$,

$$\left(a \odot \bigwedge_{x\in\mathcal{X}} x\right)(t) = a(t) \otimes \left(\bigwedge_{x\in\mathcal{X}} x\right)(t) = a(t) \otimes \bigwedge_{x\in\mathcal{X}} x(t) = \bigwedge_{x\in\mathcal{X}} (a(t) \otimes x(t)) = \bigwedge_{x\in\mathcal{X}} (a \odot x)(t),$$

since $\wedge$ operates component-wise, and $\otimes$ distributes over infinite $\wedge$ in $\overline{\mathbb{Z}}_{\min}$; finally, for $t \in \mathbb{Z}$ such that $a(t) = +\infty$,

$$\left(a \odot \bigwedge_{x\in\mathcal{X}} x\right)(t) = +\infty \otimes \bigwedge_{x\in\mathcal{X}} x(t) = +\infty,$$

which equals

$$\bigwedge_{x\in\mathcal{X}} (a \odot x)(t) = \bigwedge_{x\in\mathcal{X}} (a(t) \otimes x(t)) = \bigwedge_{x\in\mathcal{X}} (+\infty \otimes x(t)) = +\infty.$$

■

Note that, for $y \odot^\flat a$ to be defined for any $y \in \Sigma$, it suffices that $a(t) \neq \pm\infty$ for all $t \in \mathbb{Z}$. This condition is not restrictive for application purposes, as $a(t)$ will typically denote the (finite) accumulated number of firings of a transition up to and including time t. Hence, the previous propositions guarantee the existence of the residual and dual residual of the Hadamard product for any case of practical interest. In the next subsection, we see how to compute the results of these operations.

3.2 Implementation on non-increasing formal power series

In order to implement operations on non-increasing formal power series, it is convenient to consider separately monomials, polynomials, and more general periodic series. The rules for computing these operations are reported in Table 1, whose interpretation is explained in the following. Due to space constraints, the proof of their correctness is provided in the separate technical report Zorzenon et al. [2022a]. The formulas show the rules to compute $\odot$, $\odot^\sharp$, and $\odot^\flat$ between monomials r, r', polynomials p, p', and periodic series s, s' defined in the caption of Table 1. Column "Convention $+\infty - \infty$" explains how to interpret the standard additions and subtractions contained in column "Monomials", when n or n' are $+\infty$ and $-\infty$.

The computation of the Hadamard product, its residual, and its dual residual on monomials and polynomials is straightforward and, for polynomials, the result can be obtained in time complexity $\mathcal{O}(mm')$; less trivial is the situation when considering general periodic series. An important observation is that applying the Hadamard product and its residuals on two periodic series s and s' yields another periodic series s'', with throughput ν''/τ'' and periodic behavior starting at the latest at time t''_p (the values of ν'', τ'', and t''_p being reported in the table). Consequently, to compute the result of operation $\circ \in \{\odot, \odot^\sharp, \odot^\flat\}$ between s and s', we can adopt the following procedure: obtain the polynomials, say $\tilde{p}$ and $\tilde{p}'$, composed by the first terms of series s and s' up to and including time $t''_\mathrm{p} + \tau'' - 1$; compute $\tilde{p}'' = \tilde{p} \circ \tilde{p}'$; define polynomials p'' and q'' such that elements of $\tilde{p}''$ with a δ-exponent less than t''_p belong to p'', and those with a δ-exponent between t''_p and $t''_\mathrm{p} + \tau'' - 1$ belong to q''; the result of $s \circ s'$ is then $s'' = p'' \oplus q''(\nu''\delta^{\tau''})^*$. Note that values of t''_p reported in the table are only upper bounds of the starting time of the periodic pattern of s''; a formula for the exact starting time is indeed not necessary for computing s''. The only inconvenience is that series $p'' \oplus q''(\nu\delta^{\tau''})^*$ obtained in this way may be not in canonical form, resulting in a transient part longer than necessary; nevertheless, rewriting a given series in canonical form is not computationally expensive.

The procedure described above, of time complexity $\mathcal{O}(\tilde{m}\tilde{m}')$ where $\tilde{p} = \bigoplus_{i=1}^{\tilde{m}} \tilde{n}_i \delta^{\tilde{t}_i}$ and $\tilde{p}' = \bigoplus_{j=1}^{\tilde{m}'} \tilde{n}'_j \delta^{\tilde{t}'_j}$, can be applied successfully for each operation. A simple formula for t_p is unknown to the authors for the residual of the Hadamard product, but an upper bound for the beginning of the periodic pattern of $s'' = s \odot^\sharp s'$ can be computed on the basis of the analysis of series $\bar{s}$, defined by $\bar{s}(t) = s(t) - s'(t)$ for all $t \in \mathbb{Z}$; we recall that series s'' is then the greatest counter less than or equal to $\bar{s}$ (see Remark 6). It turns out that we can take t''_p as $t''_\mathrm{p} = \bar{t}_\mathrm{p} + \kappa\tau''$, with $\bar{t}_\mathrm{p} = \max(T_1, T'_1)$,

$$\kappa = 1 + \max\left(0, \left\lceil \frac{\left(\max\limits_{i=\bar{t}_1}^{\bar{t}_\mathrm{p}-1} \bar{s}(i)\right) - \left(\max\limits_{j=\bar{t}_\mathrm{p}}^{\bar{t}_\mathrm{p}+\tau''-1} \bar{s}(j)\right)}{\nu''} \right\rceil\right),$$

and $\bar{t}_1 = \min(t_1, t'_1)$.

Table 1. Rules for computing $\odot$, $\odot^\sharp$, and $\odot^\flat$ between monomials $r = \nu\delta^\tau$ and $r' = \nu'\delta^{\tau'}$, polynomials $p = \bigoplus_{i=1}^{m} n_i\delta^{t_i}$ and $p' = \bigoplus_{j=1}^{m'} n'_j\delta^{t'_j}$, and periodic series $s = p \oplus qr^*$ and $s' = p' \oplus q'r'^*$, where $q = \bigoplus_{i=1}^{l} N_i\delta^{T_i}$ and $q' = \bigoplus_{j=1}^{l'} N'_j\delta^{T'_j}$.

	Convention $+\infty - \infty$	Monomials	Polynomials	Periodic series		
				τ''	ν''	t''_{p}
$\odot$	$+\infty$	$(\nu+\nu')\delta^{\min(\tau,\tau')}$	$\bigoplus_{i=1}^{m}\bigoplus_{j=1}^{m'}(n_i\delta^{t_i} \odot n'_j\delta^{t'_j})$	$\mathrm{lcm}(\tau,\tau')$	$\tau''\left(\frac{\nu}{\tau}+\frac{\nu'}{\tau'}\right)$	$\max(T_1, T'_1)$
$\odot^\sharp$	$-\infty$	$\begin{cases}(\nu-\nu')\delta^\tau & \text{if } \tau < \tau',\\ (\nu-\nu')\delta^{+\infty} & \text{otherwise}\end{cases}$	$\bigwedge_{j=1}^{m'}\bigoplus_{i=1}^{m}(n_i\delta^{t_i} \odot^\sharp n'_j\delta^{t'_j})$	$\mathrm{lcm}(\tau,\tau')$	$\tau''\left(\frac{\nu}{\tau}-\frac{\nu'}{\tau'}\right)$	see text
$\odot^\flat$	$+\infty$	$\begin{cases}(\nu-\nu')\delta^\tau & \text{if } \tau \leq \tau',\\ \text{undefined} & \text{otherwise}\end{cases}$	$\begin{cases}\bigoplus_{i=1}^{m}\bigwedge_{j=1}^{m'}(n_i\delta^{t_i} \odot^\flat n'_j\delta^{t'_j}) & \text{if } t_m \leq t'_m,\\ \text{undefined} & \text{otherwise}\end{cases}$ ignoring undefined results on monomials	$\mathrm{lcm}(\tau,\tau')$	$\tau''\left(\frac{\nu}{\tau}-\frac{\nu'}{\tau'}\right)$	$\max(T_1, T'_1)$

4. APPLICATIONS IN DISCRETE EVENT SYSTEMS

In this section, we recall recent results on the control of a class of discrete event systems where the Hadamard product and its residuals play a prominent role. Sections 4.2 and 4.3 are mainly based on Moradi et al. [2017] and Schafaschek et al. [2020], respectively, to which the reader may refer for details. We start with a brief overview of the basic modeling formalism and related control theory.

4.1 Timed event graphs – modeling and control

Timed event graphs (TEGs) are timed Petri nets in which each place has exactly one upstream and one downstream transition and all arcs have weight 1. With each place p is associated a *holding time*, representing the minimum amount of time every token needs to spend in p before it can contribute to the firing of its downstream transition. In a TEG, we can distinguish input transitions (those that are not affected by the firing of other transitions), output transitions (those that do not affect the firing of other transitions), and internal transitions (those that are neither input nor output transitions). In this paper, we shall limit our discussion to SISO TEGs, i.e., TEGs with only one input and one output transition, which we denote respectively by u and y; internal transitions are denoted by x_i. An example of a SISO TEG is shown in Fig. 3.

We henceforth assume that TEGs operate under the *earliest firing rule*, which states that every internal and output transition fires as soon as it is enabled.

With each transition x_i, we associate a non-increasing mapping $x_i : \overline{\mathbb{Z}}_{\max} \to \overline{\mathbb{Z}}_{\min}$, for simplicity denoted by the same symbol, where, for every $t \in \mathbb{Z}$, $x_i(t)$ represents the accumulated number of firings of x_i up to and including time t. Similarly, we associate mappings u and y with input and output transitions, respectively. By inspection of Fig. 3, one can see that, at any time t, $x_1(t)$ cannot

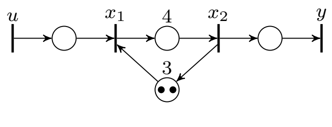

Figure 3. Simple example of a SISO TEG.

exceed (in the standard sense) the minimum between $u(t)$ and $x_2(t-3)+2$. This can be expressed as

$$(\forall t \in \overline{\mathbb{Z}}_{\max}) \quad x_1(t) \succeq u(t) \oplus 2x_2(t-3)\,. \tag{1}$$

Under the earliest firing rule, (1) turns into equality and, through the δ-transform, can be written in Σ as

$$x_1 = u \oplus 2\delta^3 x_2\,.$$

We can obtain similar relations for x_2 and y and, defining $x = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$, write

$$x = \begin{bmatrix} s_\varepsilon & 2\delta^3 \\ 0\delta^4 & s_\varepsilon \end{bmatrix} x \oplus \begin{bmatrix} 0\delta^0 \\ s_\varepsilon \end{bmatrix} u\,, \qquad y = \begin{bmatrix} s_\varepsilon & 0\delta^0 \end{bmatrix} x\,.$$

In general, a TEG can be described by implicit equations over Σ of the form

$$x = Ax \oplus Bu\,, \qquad y = Cx\,. \tag{2}$$

From Remark 1, the least solution of (2) is given by

$$x = A^*Bu \qquad \text{and} \qquad y = CA^*Bu\,, \tag{3}$$

where $G = CA^*B$ is often called the *transfer function* of the system. For instance, for the system from Fig. 3 we obtain the transfer function $G = 0\delta^4(2\delta^7)^*$.

Now, suppose equations (2) model a TEG to be controlled and let an output-reference $z \in \Sigma$ be given. We aim at a *just-in-time* input u, i.e., one that fires, by each time instant, the least possible number of times while guaranteeing that the output transition y fires at least as many times as specified by z. In other words, we seek the greatest (in the order of Σ) u such that $y = Gu \preceq z$. Based on Example 5, the solution is directly obtained by

$$u_{\mathrm{opt}} = G \mathbin{\backslash\!\!\!\circ}\, z\,. \tag{4}$$

4.2 Control of systems with resource sharing

Consider a system consisting of TEGs $S^1, \ldots, S^K$ sharing a resource with finite but arbitrary capacity, as shown in Fig. 4. β may, in general, be a TEG (or, in simple cases, just a single place) describing the capacity of the resource as well as the minimum delay between release and allocation events. H^k represents the internal dynamics of S^k. For simplicity, let us assume that input transitions (u^k) are connected to resource-allocation transitions (x_A^k) via a single place with zero delay and no initial tokens, the same being true for the connection between resource-release transitions (x_R^k) and output transitions (y^k). This

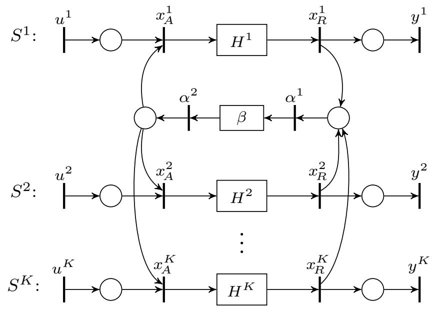

Figure 4. A number of TEGs with a shared resource.

implies $y^k = x_R^k$ and, for just-in-time inputs u^k, also $x_A^k = u^k$.

Clearly, the overall system from Fig. 4 is not a TEG, as there are places with several upstream or downstream transitions. In particular, the relationship among counters x_A^k and x_R^k, $k \in \{1, \ldots, K\}$, cannot be described by linear equations (2). With the help of the Hadamard product, however, this relationship can be expressed as

$$\beta \otimes \Big(\bigodot_{k=1}^{K} x_R^k \Big) \preceq \bigodot_{k=1}^{K} x_A^k \,. \tag{5}$$

Let the input-output behavior of each S^k, including the resource and ignoring all other subsystems, be described by $y^k = G^k u^k$, and assume respective output-references z^k are given. It should be clear that, due to the limited capacity of the resource, in general it is not possible for all subsystems to achieve the same just-in-time schedule as in the case without resource sharing. One way to settle the dispute is introducing a priority policy. We henceforth assume, without loss of generality, that the subsystems are indexed according to their priority level, meaning S^k has higher priority than S^{k+1} for all $k \in \{1, \ldots, K-1\}$. The priority policy then dictates that, for each $k \in \{2, \ldots, K\}$ and for all $i \in \{1, \ldots, k-1\}$, S^k cannot interfere with the performance of S^i.

Hence, the optimal input for S^1 can be computed neglecting any competition for the resource, which amounts to (4), i.e., $u^1_{\text{opt}} = G^1 \circ\!\!\!\!\backslash\, z^1$. Denote the corresponding resource-allocation and release schedules by $x^1_{A_{\text{opt}}}$ and $x^1_{R_{\text{opt}}}$, respectively. Then, based on (5) and making use of the fact that the Hadamard product is residuated, we obtain the optimal inputs u^k_{opt} successively for $k = 2, \ldots, K$ by computing the greatest fixed point of respective mappings $\Phi^k : \Sigma \to \Sigma$,

$$\Phi^k(u) = H^k \circ\!\!\!\!\backslash \Big[\Big(\beta \circ\!\!\!\!\backslash \big(\bigodot_{i=1}^{k-1} x^i_{A_{\text{opt}}} \odot u \big) \Big) \odot^\sharp \bigodot_{i=1}^{k-1} x^i_{R_{\text{opt}}} \Big] \wedge\ G^k \circ\!\!\!\!\backslash\, z^k \ \wedge\ u \,. \tag{6}$$

The procedure is summarized in Algorithm 1.

Example 10. In order to illustrate the method, let us consider the example of a freight railway station, adapted from Correia et al. [2009]. The station has two tracks which are used by three types of trains, as modeled in Figure 5. Input transition u^k and output transition y^k represent, respectively, the arrival and departure of a train of type k, $k \in \{1, 2, 3\}$. Upon arrival, a train can only enter the station (transitions x_A^k) if there is a track available. There is a minimum delay of 3 time units between the release of a track (transitions x_R^k) and its subsequent allocation. On each train of type 1, a container must be loaded. There is a single loader crane which can load one container at a time; provided this crane is available, the loading process starts as soon as the train enters the station, and it finishes with the firing of transition x_L^1. Trains of type 2 do not stop at this station, but need to use one of the tracks to travel through. Trains of type 3, in turn, must unload a container. It is assumed that there are always cranes available for the unloading process, which starts as soon as a train enters the station. However, the previous container must be removed from the unloading area before a new one can be deposited, so that an unloading operation can only be concluded (transition x_U^1) once every 2 time units.

Algorithm 1: Control of TEGs with resource sharing

Data: $K \in \mathbb{N}$, $\beta \in \Sigma$ and G^k, H^k, $z^k \in \Sigma$ for $1 \le k \le K$
Result: Optimal inputs $u^k_{\text{opt}} \in \Sigma$ respecting z^k, for $1 \le k \le K$
$u^1_{\text{opt}} = G^1 \circ\!\!\!\!\backslash\, z^1;\ x_{\text{AH}} = u^1_{\text{opt}};\ x_{\text{RH}} = H^1 \otimes u^1_{\text{opt}};$
for $k = 2$ **to** K **do**
  $x_0 = s_\varepsilon;\ a = s_\top;$
  while $x_0 \neq a$ **do**
    $x_0 = a;$
    $a = H^k \circ\!\!\!\!\backslash \left[\big(\beta \circ\!\!\!\!\backslash (x_0 \odot x_{\text{AH}}) \big) \odot^\sharp x_{\text{RH}} \right] \wedge (G^k \circ\!\!\!\!\backslash\, z^k) \wedge x_0;$
  end
  $u^k_{\text{opt}} = x_0;$
  $x_{\text{AH}} = x_{\text{AH}} \odot x_0;\ x_{\text{RH}} = x_{\text{RH}} \odot (H^k \otimes x_0);$
end

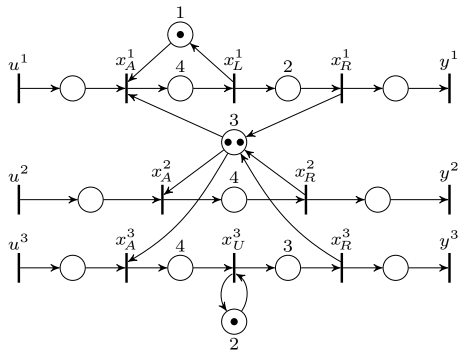

Figure 5. Model of freight railway station with two tracks.

References are given in terms of train departures and can be encoded by the following counters: $z^1 = 0\delta^{19} \oplus 1\delta^{27} \oplus 2\delta^{59} \oplus 4\delta^{+\infty}$; $z^2 = 0\delta^{24} \oplus 1\delta^{40} \oplus 2\delta^{52} \oplus 3\delta^{+\infty}$; $z^3 = 0\delta^{47} \oplus 2\delta^{+\infty}$. They can be read as follows: $\kappa\delta^\tau \oplus \kappa'\delta^{\tau'}$ means a total of κ' departures is required by time $\tau \otimes 1$. The transfer functions for the subsystems are $G^1 = 0\delta^6(1\delta^5)^*$, $G^2 = 0\delta^4(2\delta^7)^*$, and $G^3 = (0\delta^7 \oplus 1\delta^9)(2\delta^{10})^*$. Applying Algorithm 1, we obtain the optimal inputs $u^1_{\text{opt}} = 0\delta^{13} \oplus 1\delta^{21} \oplus 2\delta^{48} \oplus 3\delta^{53} \oplus 4\delta^{+\infty}$, $u^2_{\text{opt}} = 0\delta^{14} \oplus 1\delta^{36} \oplus 2\delta^{46} \oplus 3\delta^{+\infty}$, and $u^3_{\text{opt}} = 0\delta^{28} \oplus 1\delta^{38} \oplus 2\delta^{+\infty}$.

4.3 Control of systems with resource sharing and output-reference update

In practice, it may be necessary to update the reference for the output of a system during run-time, for instance when customer demand is increased and a new production objective must be taken into account. Consider the system from Fig. 4 and assume every subsystem S^k is operating optimally with respect to its own output-reference z^k, according to the priority-based strategy introduced in Section 4.2. Now, suppose that at time $T \in \mathbb{Z}$ each S^k has its reference z^k updated to $z^{k\prime}$ (with the possibility that $z^{k\prime} = z^k$ for some of them). Let us now investigate how to optimally update the inputs u^k.

For the purpose of the present discussion, let us fix an arbitrary $k \in \{1, \ldots, K\}$, and define the index sets $I_k = \{1, \ldots, k-1\}$ for $k \neq 1$, $J_k = \{k+1, \ldots, K\}$ for $k \neq K$, and $I_1 = J_K = \emptyset$. Define also the auxiliary mapping $r_T : \Sigma \to \Sigma$,

$$[r_T(s)](t) = \begin{cases} s(t), & \text{if } t \leq T\,; \\ \varepsilon\,, & \text{if } t > T\,, \end{cases}$$

and its residual

$$[r_T^\sharp(s)](t) = \begin{cases} s(t), & \text{if } t \leq T\,; \\ s(T), & \text{if } t > T\,. \end{cases}$$

As in Section 4.2, we seek the input $u_{\text{opt}}^{k\prime}$ which leads to an output as close as possible to $z^{k\prime}$ while observing the priority scheme. This implies the combined, already updated allocation and release schedules of higher-priority subsystems (i.e., of all S^i with $i \in I_k$) must be taken as a hard restriction. These can be expressed by the terms

$$\mathcal{H}_A^k = \bigodot_{i \in I_k} x_{A_{\text{opt}}}^{i\prime} \quad \text{and} \quad \mathcal{H}_R^k = \bigodot_{i \in I_k} x_{R_{\text{opt}}}^{i\prime}.$$

Furthermore, we require minimum interference from lower-priority subsystems (i.e., all S^j with $j \in J_k$). This means we have to respect past inputs (and consequent resource allocations) in these subsystems, but may ignore future ones. Recall that $u_{\text{opt}}^j(t)$ is the accumulated number of firings originally scheduled for u^j up to time t. Respecting the past means that the firings which have already occurred by time T (when the new references are received) cannot be revoked. These firings are relevant because the resulting resource releases may take place after T, thus influencing the availability of the resource. On the other hand, the prospective input firings that have not taken place by time T can still be postponed and hence, from the point of view of S^k, ignored. So, for the sake of determining the new optimal input $u_{\text{opt}}^{k\prime}$ with minimum interference from S^j, $j \in J_k$, we set $u^j(t) = u_{\text{opt}}^j(t)$ for $t \leq T$ and $u^j(t) = u_{\text{opt}}^j(T)$ for $t > T$, which is precisely captured by the counter $r_T^\sharp(u_{\text{opt}}^j)$. Since u_{opt}^j is a just-in-time input, under the assumptions in place we have $u_{\text{opt}}^j = x_{A_{\text{opt}}}^j$. The combined resource allocations and releases by lower-priority subsystems resulting from the inputs $r_T^\sharp(u_{\text{opt}}^j)$, $j \in J_k$, can then be expressed respectively by the terms

$$\mathcal{L}_A^k = \bigodot_{j \in J_k} r_T^\sharp(x_{A_{\text{opt}}}^j) \quad \text{and} \quad \mathcal{L}_R^k = \bigodot_{j \in J_k} \left(H^j \otimes r_T^\sharp(x_{A_{\text{opt}}}^j)\right).$$

Thus, based on (5) and on the foregoing discussion, the conditions for the updated allocation schedule x_A^k — namely, respecting the performance of higher-priority subsystems and ensuring minimum interference from lower-priority ones — can be expressed as

$$\beta \otimes \left(\mathcal{H}_R^k \odot (H^k \otimes x_A^k) \odot \mathcal{L}_R^k\right) \preceq \mathcal{H}_A^k \odot x_A^k \odot \mathcal{L}_A^k\,. \qquad (7)$$

Note that, as we look for a just-in-time input for S^k, we can in fact replace x_A^k with u^k in (7). Then, defining the mapping $\Psi^k : \Sigma \to \Sigma$,

$$\Psi^k(u^k) = H^k \diagdown\!\!\!\circ \left[\left(\beta \diagdown\!\!\!\circ (\mathcal{H}_A^k \odot u^k \odot \mathcal{L}_A^k)\right) \odot^\sharp (\mathcal{H}_R^k \odot \mathcal{L}_R^k)\right],$$

through straightforward manipulations one can see that (7) is equivalent to $u^k \preceq \Psi^k(u^k)$.

One last condition is that the past inputs of S^k itself must also be preserved, which amounts to requiring $r_T(u^k) = r_T(u_{\text{opt}}^k)$. The problem of determining the new optimal input $u_{\text{opt}}^{k\prime}$ $(= x_{A_{\text{opt}}}^{k\prime})$ with respect to a reference $z^{k\prime}$ given at time T can then be formulated as follows: find the greatest element of the set

$$\mathcal{F}^{k\prime} = \left\{u^k \in \Sigma \mid G^k \otimes u^k \preceq z^{k\prime} \text{ and } u^k \preceq \Psi^k(u^k) \text{ and } r_T(u^k) = r_T(u_{\text{opt}}^k)\right\}.$$

However, set $\mathcal{F}^{k\prime}$ may turn out to be empty, meaning the new reference $z^{k\prime}$ is too demanding and needs to be relaxed. In Schafaschek et al. [2020], it has been shown that there exists a least counter $\underline{u}^k$ satisfying $\underline{u}^k \preceq \Psi^k(\underline{u}^k)$ and $r_T(\underline{u}^k) = r_T(u_{\text{opt}}^k)$. This least counter can be obtained thanks to the dual residual of the Hadamard product; it is the least fixed point of mapping $\Upsilon^k : \Sigma \to \Sigma$,

$$\Upsilon^k(u^k) = \left[\left(\beta \otimes \left(\mathcal{H}_R^k \odot (H^k \otimes u^k) \odot \mathcal{L}_R^k\right)\right) \odot^\flat (\mathcal{H}_A^k \odot \mathcal{L}_A^k)\right] \oplus r_T(u_{\text{opt}}^k) \oplus u^k\,.$$

Based thereon, the least $z^{k\prime\prime} \succeq z^{k\prime}$ such that the set $\mathcal{F}^{k\prime\prime}$ (defined as $\mathcal{F}^{k\prime}$, only replacing $z^{k\prime}$ with $z^{k\prime\prime}$) is nonempty is

$$z^{k\prime\prime} = z^{k\prime} \oplus (G^k \otimes \underline{u}^k)\,.$$

Note that, if $\mathcal{F}^{k\prime} \neq \emptyset$, we have $G^k \otimes \underline{u}^k \preceq z^{k\prime}$ and hence $z^{k\prime\prime} = z^{k\prime}$.

In conclusion, the sought optimal input $u_{\text{opt}}^{k\prime}$ can be obtained as the greatest fixed point of $\Gamma^k : \Sigma \to \Sigma$,

$$\Gamma^k(u^k) = G^k \diagdown\!\!\!\circ z^{k\prime\prime} \wedge \Psi^k(u^k) \wedge r_T^\sharp(u_{\text{opt}}^k) \wedge u^k\,. \qquad (8)$$

The method is realized by Algorithm 2.

Example 11. For the railway station from Example 10, suppose the demand for trains of type 1 is updated at time $T = 30$; more specifically, one additional departure is required by time 60, i.e., $z^{1\prime} = 0\delta^{19} \oplus 1\delta^{27} \oplus 2\delta^{59} \oplus 5\delta^{+\infty}$ (whereas $z^{2\prime} = z^2$ and $z^{3\prime} = z^3$). We apply Algorithm 2 to obtain the updated inputs $u_{\text{opt}}^{1\prime} = 0\delta^{13} \oplus 1\delta^{21} \oplus 2\delta^{43} \oplus 3\delta^{48} \oplus 4\delta^{53} \oplus 5\delta^{+\infty}$, $u_{\text{opt}}^{2\prime} = 0\delta^{14} \oplus 1\delta^{36} \oplus 2\delta^{41} \oplus 3\delta^{+\infty}$, $u_{\text{opt}}^{3\prime} = 0\delta^{28} \oplus 1\delta^{57} \oplus 2\delta^{+\infty}$. In this case, $z^{1\prime}$ and $z^{2\prime}$ are feasible, but $z^{3\prime}$ is not and must be relaxed to $z^{3\prime\prime} = 0\delta^{47} \oplus 1\delta^{64} \oplus 2\delta^{+\infty}$.

5. PERFORMANCE EVALUATION

The following tests were performed in order to assess the computational time for solving optimal-control problems on TEGs with resource sharing, using our recently-developed C++ functions, available at Zorzenon et al. [2022b]. We consider K TEG subsystems that share a resource with capacity m. The internal dynamics of each subsystem S^k, $k \in \{1, \ldots, K\}$, is randomly generated as $H^k = 0\delta^{t_k} \oplus 1\delta^{t_k+\omega_{k,1}} \oplus 2\delta^{t_k+\omega_{k,1}+\omega_{k,2}}(1\delta^{\tau_k})^*$, where

Algorithm 2: Control of TEGs with resource sharing and output-reference update

Data: $K \in \mathbb{N}$, $T \in \mathbb{Z}$, $\beta \in \Sigma$ and H^k, G^k, $z^{k\prime}$, $x^k_{A_{\text{opt}}} \in \Sigma$ for $1 \leq k \leq K$

Result: New optimal inputs $u^{k\prime}_{\text{opt}} \in \Sigma$ respecting $z^{k\prime}$ (or $z^{k\prime\prime}$), for $1 \leq k \leq K$

$\mathcal{H}^k_A = \mathcal{H}^k_R = \mathcal{L}^K_A = \mathcal{L}^K_R = 0\delta^{+\infty}$;

for $k = K-1$ **to** 1 **do**

  $\mathcal{L}^k_A = \mathcal{L}^{k+1}_A \odot r^\sharp_T(x^{k+1}_{A_{\text{opt}}})$;

  $\mathcal{L}^k_R = \mathcal{L}^{k+1}_R \odot (H^{k+1} \otimes r^\sharp_T(x^{k+1}_{A_{\text{opt}}}))$;

end

for $k = 1$ **to** K **do**

  $x_0 = s_\top$; $a = s_\varepsilon$;

  while $x_0 \neq a$ **do**

    $x_0 = a$;

    $a = r_T(x^k_{A_{\text{opt}}}) \oplus x_0 \oplus \left[\left(\beta \otimes \left(\mathcal{H}^k_R \odot (H^k \otimes x_0) \odot \mathcal{L}^k_R\right)\right) \odot^\flat \left(\mathcal{H}^k_A \odot \mathcal{L}^k_A\right)\right]$;

  end

  if $G^k \otimes x_0 \preceq z^{k\prime}$ **then** $z = z^{k\prime}$;

  else $z = z^{k\prime} \oplus (G^k \otimes x_0)$;

  $x_0 = s_\varepsilon$; $a = s_\top$;

  while $x_0 \neq a$ **do**

    $x_0 = a$;

    $\Psi = G^k \,\lozenge\, z \wedge H^k \,\lozenge\, \left[\left(\beta \,\lozenge\, (\mathcal{H}^k_A \odot x_0 \odot \mathcal{L}^k_A)\right) \odot^\sharp (\mathcal{H}^k_R \odot \mathcal{L}^k_R)\right]$;

    $a = x_0 \wedge \Psi \wedge r^\sharp_T(x^k_{A_{\text{opt}}})$;

  end

  $u^{k\prime}_{\text{opt}} = x_0$; $\mathcal{H}^k_A = \mathcal{H}^k_A \odot x_0$; $\mathcal{H}^k_R = \mathcal{H}^k_R \odot (H^k \otimes x_0)$;

end

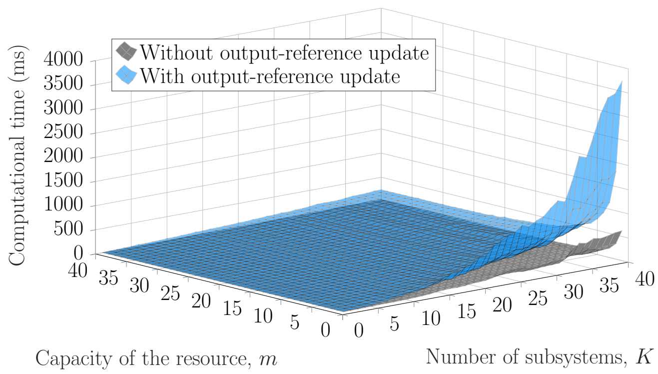

Figure 6. Computational time for solving the optimal-control problem for TEGs with resource sharing and with or without output-reference update.

$t_k, \tau_k, \omega_{k,i}$ are drawn from the discrete uniform distribution $\mathcal{U}(1, 20)$ for $i = 1, 2$. The dynamics of the resource is simply taken as $\beta = m\delta^1$, which corresponds to a single place with m initial tokens and unitary holding time. References $z^k = 0\delta^{T_k} \oplus 1\delta^{+\infty}$ and $z^{k\prime} = 0\delta^{T'_k} \oplus 1\delta^{T'_k+\Omega_{k,1}} \oplus 2\delta^{T'_k+\Omega_{k,1}+\Omega_{k,2}} \oplus 3\delta^{+\infty}$ are taken as polynomials consisting of two and four monomials respectively, and parameters $T_k, T'_k, \Omega_{k,i}$ are generated from the distribution $\mathcal{U}(1, 20)$ for $i = 1, 2$. At time $T = 20$, the output references $z^1, \ldots, z^K$ are updated to $z^{1\prime}, \ldots, z^{K\prime}$. Note that transfer functions G^k can be computed from H^k and β.

The results of the tests, performed both with and without updating the output references and considering different values of K and m (between 1 and 40), are shown in Figure 6. From the plot, we can draw the following considerations. The algorithms converge more rapidly to the solution when the capacity of the resource is large; this should not be surprising, since it is the scarcity of resources that forces low-priority subsystems to change their inputs in order to accommodate the optimal behavior of subsystems with higher priority. If the capacity of the resource is infinite, the system acts as K independent TEGs; hence, the optimal inputs can simply be computed using (4). The output-reference update introduces additional computational costs, that are even more pronounced at low resource capacity: in this case, the worst computational time (3717 ms) is achieved for $K = 40$ and $m = 1$, whereas for $K = 40$ and $m = 40$ the optimal-control problem is solved in less than one tenth of the time (245 ms).

REFERENCES

Addad, B., Amari, S., and Lesage, J.J. (2010). Linear time-varying (max,+) representation of conflicting timed event graphs. *IFAC Proceedings Volumes*, 43(12), 300–305. doi:https://doi.org/10.3182/20100830-3-DE-4013.00050. 10th IFAC Workshop on Discrete Event Systems.

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

Correia, A., Abbas-Turki, A., Bouyekhf, R., and El Moudni, A. (2009). A dioid model for invariant resource sharing problems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 39(4), 770–781. doi:10.1109/TSMCA.2009.2019867.

Cottenceau, B., Hardouin, L., and Trunk, J. (2020). A C++ toolbox to handle series for event-variant/time-variant (max,+) systems.

Hardouin, L., Cottenceau, B., Lagrange, S., and Le Corronc, E. (2008). Performance analysis of linear systems over semiring with additive inputs. In *2008 9th International Workshop on Discrete Event Systems*, 43–48. doi:10.1109/WODES.2008.4605920.

Hardouin, L., Cottenceau, B., Shang, Y., and Raisch, J. (2018). Control and state estimation for max-plus linear systems. *Foundations and Trends® in Systems and Control*, 6(1), 1–116.

Moradi, S., Hardouin, L., and Raisch, J. (2017). Optimal control of a class of timed discrete event systems with shared resources, an approach based on the hadamard product of series in dioids. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 4831–4838. IEEE.

Schafaschek, G., Hardouin, L., and Raisch, J. (2020). Optimal control of timed event graphs with resource sharing and output-reference update. *at - Automatisierungstechnik*, 68(7), 512–528. doi:doi:10.1515/auto-2020-0051. URL `https://doi.org/10.1515/auto-2020-0051`.

Zorzenon, D., Schafaschek, G., Tirpák, D., Moradi, S., Hardouin, L., and Raisch, J. (2022a). Algorithms for the hadamard product, its residual, and its dual residual in the dioid of counters. Sumbitted to arxiv.

Zorzenon, D., Schafaschek, G., Tirpák, D., Moradi, S., Hardouin, L., and Raisch, J. (2022b). Hadamard product for counters. URL `https://github.com/davidezorzenon/Hadamard_product_counters`.