Symbolic computation of periodic throughputs of timed event graphs

Stéphane Gaubert*
INRIA

Belgian-French-Netherlands'
Summer School on Discrete Event systems
Spa, 1993

Comments and Addendum, Oct. 1995. This lecture is an informal self-contained English version of the last chapter of the thesis [12]. The last section concerning the linear programming reduction has been published separately as a technical note "Resource Optimization and (min,+) Spectral theory", IEEE-TAC, (40)11, Nov. 1995.

Since this paper was written, we discovered related materials. It is worth noting that the Assertions 4.16–9 which compute the periodic throughput are a matrix counterpart of the modular algorithm –stated in terms of rational expressions– obtained in a different context by Cerin and Petit (Speedup of Recognizable Trace Languages, Proc. MFCS 93, Lect. Notes Comput. Sci. no. 711, Springer, 1993). We believe that the polyhedral simplifications used here could be applied as well to some symbolic or numerical computation of speedup of languages. We hope to present both problems in a unified framework elsewhere.

Abstract

We show that periodic throughputs of Timed Event Graphs with symbolic markings can be computed by Gaussian elimination algorithms performed in some dioid algebra of finite subsets of $\mathbb{R}^n$. Some related results about resource optimization are discussed.

Introduction

In this paper, we consider the computing of periodic throughputs of timed event graphs. Timed event graphs are a subclass of timed Petri nets which can be used to model deterministic discrete event dynamic systems subject to saturation and synchronization phenomena. Typically, we are concerned with flexible manufacturing systems, multiprocessor systems, transportation networks…These systems usually become modelizable with timed event graphs once the schedules have been fixed (so that performance evaluation of timed event graphs sometimes arises as a simple subproblem of scheduling). We shall not discuss modeling issues here (we shall just give some illustrative examples) and we refer the interested reader to [7, 1, 5, 4, 17, 20]. The most remarkable result about timed event graphs [6, 5] is certainly the following: *a timed event graph functioning at maximal speed reaches after a finite time a periodic regime*. More precisely, let x denote the *dater*

*INRIA, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cédex, France, e-mail: `Stephane.Gaubert@inria.fr`, tel: (33 1) 39.63.52.58, fax: (33 1) 39.63.53.30

function of a given transition of the graph. That is, $x(n)$ represents the date of the n-th firing of this transition (typically, such a firing represents some task or event, e.g. the production of a part, the entrance of a train on a portion of railway...). Then, there exists a constant λ (the *periodic throughput* or *production rate*), $c \in \mathbb{N}\backslash\{0\}$, $N \in \mathbb{N}$ such that

$$n \geq N \Rightarrow x(n+c) = c \times \lambda^{-1} + x(n) \ . \tag{0.0.a}$$

We call *cyclicity* the smallest value of c (the smallest value of N represent the length of the transient behavior). The name of *production rate* is justified, because we get from (0.0.a):

$$\lambda = \lim_{n\to\infty} \frac{n}{x(n)} = \text{ mean } \frac{\text{number of events}}{\text{time}} \ .$$

When the timed event graph is *strongly connected*, λ does not depend of the transition (i.e. all the transitions work at the same regime). The first practical problem about timed event graphs concerns the effective computation of the periodic throughput λ. This problem is solved by the following classical characterization. Let $N(c)$ denote the sum of the initial markings of the places of the circuit c of the graph, and $T(c)$ denote the sum of the holding times of the places of this circuit[1]. Then ([6, 5, 1]):

$$\lambda = \inf_{c} \frac{N(c)}{T(c)} \tag{0.0.b}$$

where the infimum is taken over all the circuits c of the graph. Indeed, the minimum is attained for some *elementary circuits*. Once $N(c)$ and $T(c)$ are given, computing λ from (0.0.b) reduces to a well known graph theoretical problem which can be solved by a mixture of Karp's algorithm [18], Lawler algorithms [21] and of the general algorithms for optimizing ratios given by Gondran and Minoux in the appendix V of [16]. This furnishes an efficient pseudo-polynomial method for computing λ.

A more difficult problem arises when some markings are *unknown*: we assume that the initial marking (number of tokens) of some place i is an indeterminate q_i. Such an indeterminate represents unknown resources (machines, pallets, processors, storage places, buffers). Then, $N(c) = N(c, q)$ becomes a function of the resource indeterminates $q_1, \ldots$, and the algorithms previously mentioned fail to compute the periodic throughput $\lambda = \lambda(q)$ given by (0.0.b). The motivation for computing the symbolic expression $\lambda(q)$ is clear. First of all, it provides a desirable knowledge about the system. Next, from such an expression, we may detect which resources are critical, and consider some *resource optimization* problems, which consist in maximizing the periodic throughput or minimizing some cost associated with the resources needed to reach a given production rate. These problems have been previously considered by Cohen, Dubois, Quadrat and Viot [6], Hillion and Proth [17], and Laftit, Proth and Xie [26, 20]. These contributions provide three kinds of techniques. In [6], an iterative algorithm was designed to maximize the periodic throughput. The main idea consists in finding the *critical circuits* in (0.0.b) -i.e. the circuits c attaining the infimum-. Then, the periodic throughput can be improved by adding some resources in these critical circuits. In [17], the authors considered minimizing some linear cost $J(q)$ (say an investment cost) under the constraint $\lambda(q) \geq \lambda_0$ (λ_0 denotes a minimal requested production rate). They observed that this was nothing but an integer linear programming problem, with unfortunately as many constraints as elementary circuits in the graph, making this formulation non effective. Therefore, the authors were led to giving some heuristic algorithms. In [26, 20] a noticeable progress was made: the authors showed -under additional assumptions on the graph and on the cost function- that the original linear problem reduces to an auxiliary one (with essentially as many constraints as edges in the graph), so that the exact solution can be obtained. Our purpose here is twofold.

[1] These quantities will be more precisely defined below. We just recall here that in timed event graphs, holding times are associated with places and firing times are associated with transitions. It is well known (see e.g. [1]) that times can be put only on places without loss of generality. We shall adopt this convention here. We warn the reader that some others authors use the dual convention.

- We show that is is possible to compute the symbolic expression $\lambda(q)$ without performing the naive enumeration of the elementary circuits. Our approach relies on some $(\max,+)$-algebraic characterization of production rates. We recall that in the spirit of the $(\max,+)$ approach [6, 7, 25, 1, 13, 12], timed event graphs can be represented by *rational series* over some $(\max,+)$-type algebra. Periodic throughputs appear as "poles" of these series. We show that these "poles" can be computed from the $(\max,+)$-"Newton polytopes" associated with these series. This amounts to working in some algebra of (finite) subsets of $\mathbb{R}^{p+2}$ -where p denotes the number of resources- together with some specific simplification rules. These computations also rely on very classical path algebraic techniques: roughly speaking, Gaussian elimination performs some enumeration of circuits. We have borrowed and adapted to our needs these Gaussian algorithms, used for a long time in operations research ([2, 16]) and in language theory ([19, 3, 28]).

- We show that the contribution of [26, 20] can be extended to general timed event graphs and general cost functions. We give a very simple algebraic proof based on some little $(\max,+)$ spectral theory: we show that the throughput constraint $\lambda(q) \geq \lambda_0$ is equivalent to the existence of a finite "sub-eigenvector" of a particular matrix (sub-eigenvectors are analogous to *potentials* in scheduling theory [4] and to *excessive functions* in potential theory). Then, this potential inequality translates to a set of linear constraints.

1 Two resource optimization problems

1.1 Example

We consider a flexible workshop with two part types and 3 machines, as shown on Figure 1. Processing times are given on the table below:

processing times	part 1	part 2	total charge
machine 1	9	2	11
machine 2	2	7	9
machine 3	1	10	11

This is a system of "flowshop" type. That is, the order of processing of the parts by the machines is the same for all the parts. Here, the first part of type 1 is conveyed on a pallet and processed on machines 1,2,3, in this order. Then the part exits the system and the pallet returns to machine 1 to take another type 1 part. The same treatment is made for parts of type 2. We assume that q_1 pallets are reserved for the exclusive transportation of type 1 parts, and that q_2 other pallets are reserved for type 2 parts. The initial position of the pallets is upstream machine 1. We observe that machines 1 and 3 are *bottleneck machines*, with processing cycles of 11 time units: this means that whatever the pallet allocation may be, the system will never produce more than 1 part over 11 units of time. However, if the quantities of pallets is not sufficient, the production can be worse than $1/11$. For instance, with one pallet for each part type (i.e. $q_2 = q_1 = 1$), we shall see that we have a periodic throughput of $1/19$. Here, our purpose consists in minimizing the quantities q_1 and q_2 in order to ensure that the bottleneck machines M_1 and M_3 work at maximal rate. We shall show -after some simple algebraic computations that we perform in detail in section §4.3- that the production rate $\lambda(q)$ is given by the following $(\min,+)$ function of the resources:

$$\lambda(q) = \min(\frac{1}{11}, \frac{q_1}{12}, \frac{1+q_1}{29}, \frac{q_2}{19}) \quad . \tag{1.1.a}$$

It should be noted that according to Formula (0.0.a), each term of the $\min$ corresponds to some circuit of the graph. E.g. $\frac{q_1}{12}$ corresponds to the circuit $c = (x_5, x_3, x_1)$ for which $N(c) = q_1$ and $T(c) = 12$. Similarly,

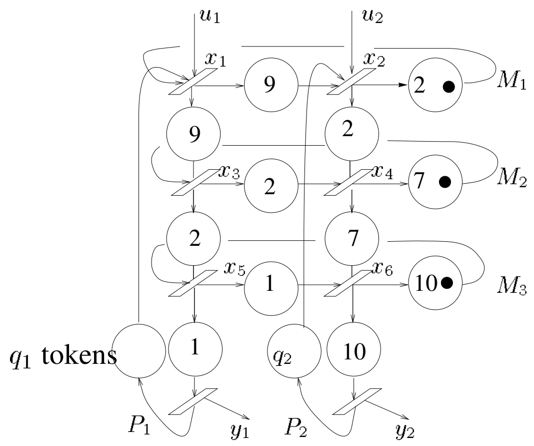

Figure 1: Flowshop with 2 part types and 3 machines.

the ratio $\frac{1+q_1}{29}$ corresponds to the circuit $(x_5, x_6, x_4, x_3, x_1)$). The workshop operates at maximal speed if each term in (1.1.a) is greater than the maximum throughput $\frac{1}{11}$, i.e. $\frac{q_1}{12} \geq \frac{1}{11}, \frac{1+q_1}{29} \geq \frac{1}{11}, \frac{q_2}{19} \geq \frac{1}{11}$, which gives the unique minimal integer valued allocation of pallets

$$q_1 = 2, q_2 = 2 \ .$$

This leads us to formulate more precisely the two following resources optimization problems.

1.2 First problem

Let $q_1, \ldots, q_p$ represent unknown resources and let $\lambda(q)$ denote the associated periodic throughput. Since the map $q \mapsto \lambda(q)$ is obviously nondecreasing, we have $\lambda(q) \leq \lambda(+\infty)$. We shall assume that the maximal periodic throughput $\overline{\lambda} = \lambda(+\infty)$ is finite (this means that there exists a cycle of tasks with finite fixed resources, e.g. in the above example, we have $\overline{\lambda} = 1/11$). The first resource optimization problem consists in finding one (or all) allocation of resources such that the system admits the maximal production rate $\overline{\lambda}$. That is, we look for a minimal $q \in \mathbb{N}^n$ such that

$$\lambda(q) = \overline{\lambda} \ .$$

1.3 Second problem

Let λ_0 denote a target throughput (that is, we require q to satisfy $\lambda(q) \geq \lambda_0$) and let us consider a linear cost associated with the resources:

$$J(q) = c_1 q_1 + \ldots + c_p q_p \ ,$$

(c_i represents the price of a unit of resource i, say, the price for buying a machine or a pallet). We consider minimizing the “design cost” $J(q)$ under the throughput constraint, i.e.

$$\min\{J(q) \mid \ q \in \mathbb{N}^p, \ \ \lambda(q) \geq \lambda_0\} \ . \tag{1.3.b}$$

We have only mentioned the first problem here because of its practical importance: it has been solved by the iterative algorithms described in [6, 17] which provides a (in general non unique) minimal allocation. It should be noted that these iterative algorithms do not require the symbolic computation of the periodic throughput $\lambda(q)$, although the knowledge of $\lambda(q)$ immediately gives the solution. Therefore, the symbolic and spectral techniques that we develop hereafter primarily concern the second problem (together with the computing of $\lambda(q)$ which is interesting in itself).

2 Some review of operatorial methods for timed event graphs

We briefly recall here the operatorial representation of timed event graphs introduced by Cohen, Moller, Quadrat and Viot [7] (see also [1]).

2.1 Dater functions and $(\max, +)$-linear operators

The first step consists in introducing the $(\max, +)$-algebra, i.e. the *semiring* $\overline{\mathbb{R}}_{\max} \stackrel{\text{def}}{=} \mathbb{R} \cup \{-\infty, +\infty\}$ equipped with $\max$ (denoted by $\oplus$, e.g. $1 \oplus 2 = 2$) and addition (denoted by $\otimes$, e.g. $1 \otimes 3 = 4$), with the convention $(-\infty) \otimes (+\infty) = (-\infty) + (+\infty) = -\infty$. We set $\varepsilon \stackrel{\text{def}}{=} -\infty$ and $e \stackrel{\text{def}}{=} 0$, in order to have $\varepsilon \oplus x = \max(-\infty, x) = x$ for all x (ε is the neutral for addition) and $e \otimes x = x$ (e is the unit). Moreover, the zero element ε is absorbing[2]: $\varepsilon \otimes x = x$. The main discrepancy by comparison with rings is that addition is idempotent: $x \oplus x = x$. Idempotent semirings are also called dioids: these structures enjoy many algebraic properties borrowed to rings and lattices. We refer the interested reader to [8, 16, 1, 23, 22]. Next, we introduce *dater functions*. A *dater* x is a nondecreasing map $\mathbb{Z} \to \overline{\mathbb{R}}_{\max}$ (for most applications, $x(n)$ represents the date of the n-th occurrence of some event, for instance, the date of the n-th firing of a transition). We shall denote by Dat the set of dater functions. Daters are equipped with pointwise max, denoted by $\oplus$:

$$(x \oplus x')(n) \stackrel{\text{def}}{=} x(n) \oplus x'(n) = \max(x(n), x'(n))$$

and addition, written as an external product:

$$\forall \lambda \in \overline{\mathbb{R}}_{\max}, \quad (\lambda x)(n) \stackrel{\text{def}}{=} \lambda \otimes x(n) = \lambda + x(n) \ .$$

Such a structure is known as a *moduloid* over $\overline{\mathbb{R}}_{\max}$ (moduloids are the analogue for dioids of modules for rings). Let us just notice that moduloids and dioids are canonically ordered by

$$a \preceq b \iff a \oplus b = b \ .$$

In $\overline{\mathbb{R}}_{\max}$, $\preceq$ coincides with the usual order, for $a \preceq b \iff \max(a, b) = b$. In general, $a \oplus b$ coincides with the least upper bound of $\{a, b\}$ for the order $\preceq$. We say that such an ordered structure is *complete* if any (infinite) subset A admits a least upper bound that we denote by $\bigoplus A$ or $\bigoplus_{a \in A} a$. $\overline{\mathbb{R}}_{\max}$ and Dat are obvious examples of complete ordered structures.

In conventional control theory, we usually consider linear operators (which associate signals to signals and satisfy the superposition principle). Here, we introduce $\mathcal{L}(\mathsf{Dat})$, the set of operators $\mathcal{S} : \mathsf{Dat} \to \mathsf{Dat}$ which satisfy the following properties

Linearity	$\forall u \in \mathsf{Dat}, \quad \mathcal{S}(u \oplus v) = \mathcal{S}(u) \oplus \mathcal{S}(v)$ $\forall \lambda \in \overline{\mathbb{R}}_{\max}, \forall u \in \mathsf{Dat}, \quad \mathcal{S}(\lambda u) = \lambda \mathcal{S}(u)$
Continuity	for all (infinite) family of daters $\{u_i\}_{i \in I}$, $\mathcal{S}(\bigoplus_{i \in I} u_i) = \bigoplus_{i \in I} \mathcal{S}(u_i)$.

[2] this is why we have chosen the convention $(-\infty) + (+\infty) = -\infty$.

Indeed additivity is a special case of continuity (when I is reduced to two elements). Continuity yields an "infinite superposition principle" which will play a role very similar to regularity assumptions in the conventional theory. The word continuity is justified if we consider some topology related with the order [14]. The set $\mathcal{L}(\mathsf{Dat})$ of linear continuous operators over Dat is naturally equipped with sum and composition product:

$$\forall f, g \in \mathsf{Dat}, \quad \begin{array}{lcl} [(f \oplus g)u](n) & \stackrel{\text{def}}{=} & [f(u)](n) \oplus [g(u)](n), \\ [(f \otimes g)(u)](n) & \stackrel{\text{def}}{=} & [f(g(u))](n) \ . \end{array}$$

Then, $(\mathcal{L}(\mathsf{Dat}), \oplus, \otimes)$ becomes a complete dioid. In the following, we shall of course drop the unuseful parentheses, writing $fgu(n)$ for $[f(g(u))](n)$.

2.2 Formal series

The two simplest (and most useful) examples of linear continuous operators on Dat are provided by shifts:

- shift in counting γ:
$$\gamma u\,(n) \stackrel{\text{def}}{=} u(n-1)$$
- shift in dating δ:
$$\delta u\,(n) \stackrel{\text{def}}{=} u(n) + 1 = 1 \otimes u(n) \ .$$

We observe that γ and δ commute and satisfy the two following simplification rules of paramount importance

$$\begin{array}{llcl} \text{(i)} & \gamma^n \oplus \gamma^m & = & \gamma^{\min(n,m)} \\ \text{(ii)} & \delta^t \oplus \delta^s & = & \delta^{\max(t,s)} \ . \end{array} \tag{2.2.a}$$

The second rule is obvious from the definition of $\oplus$, the first rule is valid because daters are non decreasing functions, for

$$\begin{array}{rcl} (\gamma^n \oplus \gamma^m)u(k) & = & \gamma^n u(k) \oplus \gamma^m u(k) \\ & = & \max(u(k-n), u(k-m)) \\ & = & u(k - \min(n,m)) \\ & = & \gamma^{\min(n,m)} u(k) \ . \end{array}$$

The two shifts operators γ and δ are fundamental because they provide a kind of basis for the linear operators we shall deal with:

2.2.1 Definition ($\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$). *We shall denote by* $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$ *the dioid of sums of the form*

$$s = \bigoplus_{(n,t)\in F} \gamma^n \delta^t, \quad \text{with} \quad F \subset \mathbb{Z}^2 \ . \tag{2.2.b}$$

Let $\mathbb{B} = \{\varepsilon, e\}$ denote the boolean dioid. Equivalently, $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$ can be defined as the *quotient* of $\mathbb{B}[[\gamma,\delta]]$ -the dioid of boolean formal series with integer exponents in the commuting indeterminates γ, δ- by the congruence generated by the simplification rules (2.2.a). Using the standard notation for formal series, (2.2.b) can also be written

$$s = \bigoplus_{n,t\in\mathbb{Z}} s_{nt}\gamma^n\delta^t \tag{2.2.c}$$

with

$$s_{nt} = \begin{cases} e & \text{if } (n,t) \in F \\ \varepsilon & \text{if } (n,t) \notin F. \end{cases}$$

It should be noted that representations of the form (2.2.b) or (2.2.c) are not unique (e.g. $e = e \oplus \delta^{-1} \oplus \delta^{-2} \oplus \dots$ due the simplification rules). Therefore, some little care is needed in order to define such notions as causality. We shall say that a series is *causal* if there exists *a* representation of the form (2.2.b) with nonnegative exponents (i.e. $F \subset \mathbb{N}^2$). Similarly, we shall say that s is a *polynomial* if there exists a representation of s as a finite sum of monomials. We shall denote by $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma,\delta]$ the subdioid of polynomials.

Since $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$ is defined by a *presentation* (i.e. by generators and relations), we have to solve the "*word problem*" which consists in deciding if two series are equal. The solution is provided by a simple observation from which we shall build a geometric representation of formal series. We first recall the the Kleene's star of a series s is defined by

$$s^* = e \oplus s \oplus s^2 \oplus \dots$$

(no problem of convergence here due to the completion of the dioid of boolean series).

2.2.2 Proposition *The two series $s, s' \in \mathbb{B}[[\gamma,\delta]]$ belong to the same coset in $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$ iff*

$$s\gamma^*(\delta^{-1})^* = s'\gamma^*(\delta^{-1})^* \ .$$

Let $\overline{s}$ denote the coset of s. As we noticed before, the simplification rules imply that $\overline{e} = \overline{(\delta^{-1})^*}$. Similarly, $\overline{e} = \overline{\gamma^*}$. Hence, $\overline{s\gamma^*(\delta^{-1})^*} = \overline{s}$, and the condition is clearly necessary. The sufficiency will become clear at once with the geometric interpretation.

We first define the *support map*[3]

$$\text{supp} : \begin{cases} \mathbb{B}[[\gamma,\delta]] & \longrightarrow & (\mathcal{P}(\mathbb{Z}^2), \cup, +) \\ s & \longmapsto & \text{supp}\, s \stackrel{\text{def}}{=} \{(n,t) \in \mathbb{Z}^2 \mid s_{nt} \neq \varepsilon\} \ . \end{cases}$$

This is plainly an isomorphism of dioids. The support of $\gamma^*(\delta^{-1})^*$ will play a particular role:

$$\mathcal{C} \stackrel{\text{def}}{=} \text{supp}\, \gamma^*(\delta^{-1})^* = \mathbb{N}^+ \times (\mathbb{N}^-) = \{(n,-t) \mid (n,t) \in \mathbb{N}^2\} \ .$$

Then, Proposition 2.2.2 is equivalent to saying that two boolean series s, s' belong to the same coset iff

$$\text{supp}\, s + \mathcal{C} = \text{supp}\, s' + \mathcal{C} \ .$$

2.2.3 Example The supports of the polynomials $p = e \oplus \gamma^2\delta^2 \oplus \gamma^2 \oplus \gamma^3\delta^3$ (small black squares) and $p' = e \oplus \gamma^2\delta^2 \oplus \gamma^3\delta^3$ (circles) are shown on Figure 2. These two polynomials belong to the same coset because their "south-east shadows" $\text{supp}\, s + \mathcal{C}$ and $\text{supp}\, s' + \mathcal{C}$ (in grey) are equal.

2.3 $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$ representation of timed event graphs

We next recall how timed event graphs are described by operators in $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$. Let us consider again the flexible workshop of Figure 1. Let $x_i(n)$ denote the *dater function* associated with the transition labeled x_i (e.g. $x_2(n)$ represents the date of the processing of the n-th type 2 part by machine 1). Similarly, $u_i(n)$

[3] $\mathcal{P}(\mathbb{Z}^2)$ denotes the set of subsets of $\mathbb{Z}^2$, $+$ denotes the vector sum.

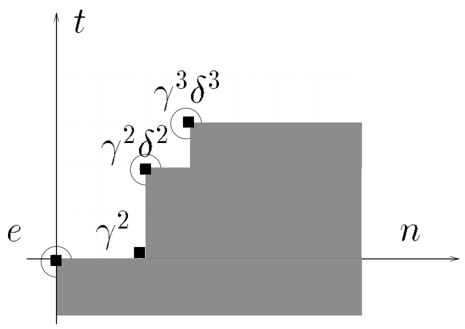

Figure 2: Support of polynomials

denotes the date of the n-th arrival of row materials for part i, and $y_i(n)$ the date of end of production of the n-th type i part. It should be clear that, for instance

$$x_2(n) \geq \max(10 + x_6(n - q_1), 9 + x_1(n))$$

i.e.

$$x_2 \geq \gamma^{q_1}\delta^{10}x_6 \oplus \delta^9 x_1 \ .$$

Writing analogous inequalities for the other transitions, we obtain a system of matrix inequalities which writes

$$x \geq Ax \oplus Bu, \quad y \geq Cx \ , \tag{2.3.d}$$

where A, B, C are matrices with *causal* entries in $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma, \delta]$. Here,

$$A = \begin{bmatrix} \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \gamma^{q_1}\delta & \varepsilon \\ \delta^9 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma^{q_2}\delta^{10} \\ \delta^9 & \varepsilon & \varepsilon & \gamma\delta^7 & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \gamma\delta^{10} \\ \varepsilon & \varepsilon & \varepsilon & \delta^7 & \delta & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} \tag{2.3.e}$$

$$C = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^{10} \end{bmatrix} \ .$$

A standard fix point argument shows that the least -i.e. the *earliest*- solution of (2.3.d) is given by

$$x = A^*Bu, \quad y = Hu$$

where

$$H \stackrel{\text{def}}{=} CA^*B = CB \oplus CAB \oplus CA^2B \oplus \dots \tag{2.3.f}$$

is the *transfer matrix* of the system. We next provide algorithms to compute transfers and show how the periodic throughputs can be obtained from the transfer representation (2.3.f).

2.4 From signals to formal series

In conventional control theory, there is a natural correspondence between signals and series (provided by some z-transform). The situation is similar for $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$. It is easily checked that the map

$$\mathrm{ser} : \begin{cases} \mathsf{Dat} & \longrightarrow \quad \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]] \\ d & \longmapsto \quad \mathrm{ser}\, d \stackrel{\text{def}}{=} \bigoplus_{n\in\mathbb{Z}} \gamma^n \delta^{d(n)} \end{cases} \tag{2.4.g}$$

(with the convention $\delta^{+\infty} \stackrel{\text{def}}{=} \delta^*, \delta^{-\infty} \stackrel{\text{def}}{=} \varepsilon$) is bijective. Indeed, if we equip daters with pointwise max (written $\oplus$) and convolution product:

$$d \otimes d'(k) \stackrel{\text{def}}{=} \bigoplus_{r+s=k} d(r) \otimes d'(s) \ ,$$

ser becomes a dioid isomorphism from $(\mathsf{Dat}, \oplus, \otimes)$ to $(\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma, \delta]], \oplus, \otimes)$. We let the reader check that the inverse isomorphism is given by

$$\text{dat} \stackrel{\text{def}}{=} \text{ser}^{-1} : \quad \text{dat}\, s(n) = \sup\{t \mid \gamma^n \delta^t \preceq s\} \ .$$

This allows to interpret the transfer H in terms of *impulse response*. Let e denote the unit for the convolution product, or "impulse"

$$e(n) = \begin{cases} -\infty & \text{if } n < 0 \\ 0 & \text{if } n \geq 0 \end{cases} \ .$$

The morphism property of ser gives in particular:

$$\text{ser}\, e = \bigoplus_{n \in \mathbb{N}} \gamma^n = \gamma^* = e \ .$$

Moreover, if $y = Hu$, then $\text{ser}\, y = H \text{ser}\, u$. Setting $h = \text{dat}\, H$ and $y = He$, we get $\text{ser}\, y = H \text{ser}\, e = H$, hence

$$y = \text{dat}\, \text{ser}\, y = \text{dat}\, H = h \ .$$

This means that $h(n)$ is equal the the date $y(n)$ of the n-th output of the system when the input is equal to the impulse e: h is the *impulse response* of the system.

2.4.1 Example For the system of Figure 1, with $q_1 = q_2 = 1$, it can be shown (using the Jordan algorithm described below) that the transfer is given by

$$H = \begin{bmatrix} \delta^{12} \oplus \gamma\delta^{29} \left(\gamma\delta^{19}\right)^* & \gamma\delta^{20} \left(\gamma\delta^{19}\right)^* \\ \delta^{28} \left(\gamma\delta^{19}\right)^* & \delta^{19} \left(\gamma\delta^{19}\right)^* \end{bmatrix}$$

The support function associated with H_{11} is shown on Figure 3. The impulse response $h_{11} = \text{dat}\, H_{11}$ is

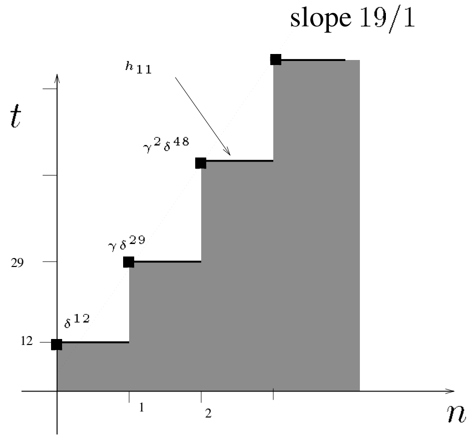

Figure 3: Impulse response

represented by the upper border of this set (black staircase line). This means that if the row materials of type

2 are non critical ($u_2 = \varepsilon$) and if an infinite quantity of row materials for part 1 becomes available at time 0 ($u_1 = e$), then the output function $n \mapsto y_1(n)$ is equal to h_{11}. In particular, we see that this function becomes periodic after $N = 1$. The periodic throughput is equal to the inverse of the asymptotic slope of h_{11}, i.e. $\lambda = \frac{1}{19}$.

2.5 Rationality and Periodicity

The only series we are interested in arise from finite dimensional systems of the form (2.3.f). This motivates the following definition:

2.5.1 Definition (Rationality). *The subdioid* $\mathrm{Rat}(\gamma, \delta)$ *of* rational series *is the least subdioid of* $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$ *containing causal polynomials and stable under the star operation.*

In the following, we shall allays use causal polynomials and we shall drop the term "causal" when no confusion occurs. We are interested in giving simple canonical forms for rational series (in order to perform effective computations).

2.5.2 Definition (periodicity). *The series* $s \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$ *is* periodic *(with periodicity* (ν, τ)*) if there exists two (causal) polynomials* p, q *such that*

$$s = p \oplus q(\gamma^\nu \delta^\tau)^* \quad . \tag{2.5.h}$$

The denomination "periodic" is justified by the following:

2.5.3 Proposition *The series* s *is periodic with periodicity* (ν, τ) *iff there exists* $N \in \mathbb{N}$ *such that*

$$n \geq N \Rightarrow \mathrm{dat}\, s(n + \nu) = \tau + \mathrm{dat}\, s(n) \quad . \tag{2.5.i}$$

In other words, after some transient behavior (whose length N essentially depends on the degrees of p and q), ν events occur every τ units of time. Let us show that (2.5.i) implies the periodicity of s. We have

$$s = \bigoplus_{0 \leq n \leq N-1} \gamma^n \delta^{\mathrm{dat}\, s(n)} \oplus \left(\bigoplus_{N \leq n \leq N+\nu-1} \gamma^n \delta^{\mathrm{dat}\, s(n)} \right) (\gamma^\nu \delta^\tau)^*$$

which is a representation of type (2.5.h). The converse implication is of the same vein and we do not prove it here.

The main theorem about rational series of $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$ is due to Cohen, Moller, Quadrat and Viot:

2.5.4 Theorem ([7]). *A series* $s \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$ *is rational iff it is periodic.*

See also [1, 12] for the proof. Obviously, a periodic series is rational (since $p \oplus q(\gamma^\nu \delta^\tau)^*$ is a finite rational expression). More generally, it is well known [10] that in commutative dioids, rational series can be written as finite sums $\bigoplus_i P_i Q_i^*$ where the P_i, Q_i are polynomials. What is more original here is that there is a *single* star of a *monomial*. The periodic representation (2.5.h) is of course not unique (e.g. $(\gamma\delta)^* = e \oplus \gamma\delta(\gamma\delta)^*$). However, some notion of minimality can be introduced ([13, 12]) which yields a canonical representation of series. This has allowed us to write a `MAPLE` package called `MAX` which implements these rational series. In particular, the examples of this paper have been checked with `MAX`. Here, we are not concerned with the arithmetics of rational series, and we refer the interested reader to [13, 12].

We shall need the definition of *minimal slope* and *ultimate slope* of rational series.

2.5.5 Definition (slopes). *Let $s = \bigoplus_{i \in I} \gamma^{n_i} \delta^{t_i}$. The* minimal slope *of s is by definition*

$$\sigma(s) = \inf_{t_i > 0,\ i \in I} \frac{n_i}{t_i} \quad .$$

If s is rational with periodicity (ν, τ), the ultimate slope *of s is the ratio*

$$\sigma_\infty(s) = \frac{\nu}{\tau} \quad .$$

It is easily checked that $\sigma(s)$ does not depend of the representation $s = \bigoplus_{i \in I} \gamma^{n_i} \delta^{t_i}$ and that even if p, q, (ν, τ) are not unique in (2.5.h), the ratio $\frac{\nu}{\tau}$ is unique, for

$$\sigma_\infty(s) = \lim_{n \to \infty} \frac{n}{\operatorname{dat} s(n)} \quad . \tag{2.5.j}$$

It follows from Proposition 2.5.3 that $\sigma_\infty(s)$ is equal to the periodic throughput of the impulse response of the SISO system with transfer s.

2.5.6 Example We have represented on Figure 4 the support of $s = e \oplus \gamma^3\delta^5 \oplus \gamma^8\delta^6 \oplus \gamma^9\delta^8(\gamma^2\delta^2)^*$. The minimal slope $\sigma(s)$ is attained by the monomial $\gamma^3\delta^5$. It should be noted that these slopes only depend of the convex hull of the graph of s (bold line). Therefore, these two dimensional representations of rational series can be seen as the analogous of *Newton polytopes* (or polygones) which occur in classical analysis. The asymptotic slope of the "Newton polygone" gives the periodic throughput. The analogy with conventional rational series is complete if we consider the following correspondence:

$$\begin{array}{lcl} \mathrm{Rat}(\gamma, \delta) & & \mathbb{C}(X) \\ \gamma^\nu \delta^\tau & \leftrightarrow & \tau X^\nu \\ (\gamma^\nu \delta^\tau)^* & \leftrightarrow & (1 - \tau X^\nu)^{-1} \end{array}$$

Let us assume for simplicity that the rational series $s = \bigoplus_n s_n X^n = P/Q \in \mathbb{C}(X)$ has a unique pole ρ of minimal modulus, and that $\rho \neq 0$. Then, the following is the counterpart of the "asymptotic slope" property (2.5.j)

$$\lim_{n \to \infty} \frac{\log |s_n|}{n} = -\log |\rho| \quad .$$

This shows that the periodic throughput $\lambda = \sigma_\infty(s)$ is analogous to the "minimal pole" of s. The main discrepancy with the usual algebra is that there is a unique (!) "pole" (i.e. a single star) in the canonical form (2.5.h).

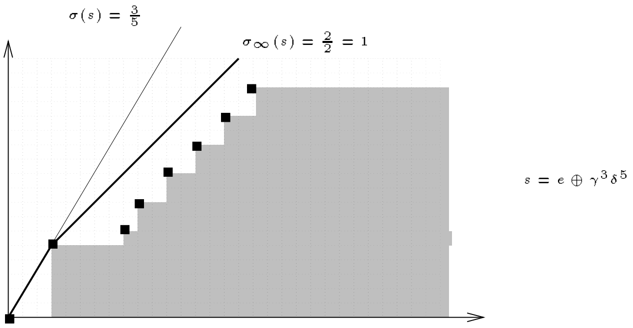

Figure 4: Slopes and Newton Polygon

We list the basic properties of slopes:

2.5.7 Properties *We have, for all rational series s, s':*

i $\sigma_\infty(s \oplus s') = \min(\sigma_\infty(s), \sigma_\infty(s'))$

ii $s, s' \neq \varepsilon \Rightarrow \sigma_\infty(ss') = \min(\sigma_\infty(s), \sigma_\infty(s'))$

iii $\sigma(ss') \geq \min(\sigma(s), \sigma(s'))$.

iv $\sigma_\infty(s) \geq \sigma(s)$,

v $\forall k \in \mathbb{N}\backslash\{0\},\ \sigma(s^k) = \sigma(s)$

vi $\sigma(s) = \sigma(s^*)$

vii $\sigma_\infty(s^*) = \sigma(s)$.

Properties (i) (iv) are obvious. Using the $\min$-additivity of the minimal slope, it is enough to prove (iii) for monomials (i.e. $s = \gamma^n\delta^t$ and $s' = \gamma^{n'}\delta^{t'}$). Then, it becomes a straightforward inequality.
(v): it follows from (iii) that $\sigma(s^k) \geq \sigma(s)$. If $\sigma(s) = \sigma(\gamma^n\delta^t) = n/t$ is attained for some monomial $\gamma^n\delta^t \preceq s$, then the result is clear, since $\sigma(s^k) \leq \sigma((\gamma^n\delta^t)^k) = (kn)/(kt) = \sigma(s)$. If the slope is not attained, we may use a limit argument.
(vi) follows readily from (v) and the fact that $\sigma(\bigoplus_i u_i) = \inf_i \sigma(u_i)$ for all (infinite) family u_i.
(vii) follows from (vi) and the fact that $\sigma_\infty(s^*) = \sigma(s^*)$. This together with the remaining properties is left to the reader (see also [12]). ∎

2.6 Computing stars of rational matrices

In the conventional algebra, the inverses of matrices are easily computed by using the Gauss algorithm. In the same spirit, there is a variety of Gaussian-type algorithm to compute the star of matrices in complete dioids. Some extensions of these algorithms will be discussed later on. Here, it is sufficient to recall the following "Jordan" algorithm:

2.6.1 Algorithm ([16]). *Let A be a $n\times n$-matrix with entries in a complete dioid $\mathcal{D}$, and define the matrices $A^{(0)}, \ldots, A^{(n)}$ by the following induction:*

$$\begin{array}{l} A^{(0)} = A \\ \text{for } i, j = 1, \ldots, n \\ A^{(k)}_{ij} = A^{(k-1)}_{ij} \oplus A^{(k-1)}_{ik}(A^{(k-1)}_{kk})^* A^{(k-1)}_{kj}. \end{array} \tag{2.6.k}$$

Then $A^{(n)} = A^+ \stackrel{\text{def}}{=} A \oplus A^2 \oplus A^3 \oplus \ldots$.

This computes A^+ instead of A^*, but it follows from $A^+ = AA^* = A^*A$ and $A^* = \mathrm{Id} \oplus A^+ = (\mathrm{Id} \oplus A)^+$ that the computations of A^* and A^+ are essentially equivalent. By analogy with the usual algebra, we shall call $A^{(k-1)}_{kk}$ the k-th pivot.

2.6.2 Definition (Graph of a matrix). *Let A be a $n\times n$ matrix. The (directed) graph associated with A has n vertices $\{1, \ldots, n\}$ and there is an edge $j \mapsto i$ iff $A_{ij} \neq \varepsilon$. A* path *of length k from j to i is a sequence of vertices $p = (i_1, \ldots, i_{k+1})$ such that $i_{k+1} = j$, $i_1 = i$, and the* weight

$$\mathsf{w}_A(p) \stackrel{\text{def}}{=} A_{i_1 i_2} \otimes \ldots \otimes A_{i_k i_{k+1}}$$

is different from ε. A circuit *c of length k is a path of length k such that $i_{k+1} = i_1$. We shall write $c = (i_1, \ldots, i_k)$ instead of $c = (i_1, \ldots, i_{k+1})$.*

We next give a characterization of the periodic behavior of A^* in function of the circuits of A. Let us extend the definition of the minimal slope σ to matrices as follows:

$$\sigma(A) \stackrel{\text{def}}{=} \inf_c \sigma(\mathsf{w}_A(c)) \tag{2.6.1}$$

where the infimum is taken over all the circuits of A. Indeed, since a non elementary circuit is a product of elementary circuits, Property 2.5.7,iii implies that the infimum is the same if we restrain (2.6.1) to elementary circuits. The following also are immediate consequencies of 2.5.7:

2.6.3 Properties

i $\forall k \in \mathbb{N}\backslash\{0\}$, $\sigma(A) = \sigma(A^k)$,

ii $\sigma(A^+) = \sigma(A)$.

2.6.4 Theorem *Let $A \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma, \delta]^{n\times n}$ be an irreducible matrix. Then*

$$\sigma_\infty(A^*_{ij}) = \sigma(A) = \min_c \sigma(\mathsf{w}_A(c)) \enspace .$$

In particular, the theorem asserts that the ultimate slope is independent of (ij) (which means that the whole graph works at the same speed).

Proof We first show that $\sigma_\infty(A^*_{ij})$ is independent of ij. Indeed, it is enough to show that $\sigma_\infty(A^*_{ij}) = \sigma_\infty(A^*_{i'j})$ and that $\sigma_\infty(A^*_{ij}) = \sigma_\infty(A^*_{ij'})$. The connectivity assumption implies that $A^+_{ii'} \neq \varepsilon$ and $A^+_{i'i} \neq \varepsilon$. Hence, taking the (ij) entry of $A^* \succeq A^+A^*$, we get

$$A^*_{ij} \succeq A^+_{ii'}A^*_{i'j} \enspace .$$

Hence, by 2.5.7,iii,

$$\sigma_\infty(A^*_{ij}) \leq \sigma_\infty(A^*_{i'j}) \enspace .$$

A symmetry argument shows that these two quantities are equal. The same argument applied to A^t shows that $\sigma_\infty(A^*_{ij}) = \sigma_\infty(A^*_{ij'})$, hence, $\sigma_\infty(A^*_{ij})$ is independent of ij.

Next, a well known result of the Perron-Frobenius theory (see e.g. [30]) states that, due to irreducibility assumption, there exists some power A^k of A such that A^k is block diagonal, all the diagonal blocks having non ε entries. Hence, from 2.6.3,i, we may assume that all the entries of A are different from ε.

Our proof relies on a combinatorial interpretation of the Jordan algorithm. Let $I \subset \{1, \dots, n\}$. We say that the path $p = (i = i_1, \dots, i_{k+1} = j)$ traverses exclusively I if all the intermediate vertices $i_2, \dots, i_k$ belong to I (but we allow the ends $i_1 = i$ and $i_{k+1} = j$ not to belong to I). Let us recall that $A^{(k)}$ denotes the matrix obtained at the k-th iteration of the Jordan algorithm 2.6.1. Then, the following lemma is easily checked by induction [12]:

2.6.5 Lemma $A^{(k)}_{ij}$ *is the sum of the weights of all the paths from j to i traversing exclusively* $\{1, \dots, k\}$.

Let us define

$$\sigma^k \stackrel{\text{def}}{=} \min_{ij} \sigma(A^{(k)}_{ij}), \quad \pi_k = A^{(k-1)}_{kk}$$

We have

$$\sigma^1 = \sigma(\mathrm{tr}(A)), \qquad \sigma^{k+1} = \min(\sigma^k, \sigma(\pi_k)) \enspace .$$

Hence

$$\sigma_\infty(A^*_{nn}) = \sigma_\infty(A^+)_{nn} = \sigma^n = \sigma(\pi)$$

where

$$\pi = \pi_1 \oplus \ldots \oplus \pi_n \ .$$

The combinatorial interpretation 2.6.5 of $\pi_k = A_{kk}^{(k-1)}$ shows that π is a sum of weights of circuits, containing at least all the elementary circuits. Hence, $\sigma(\pi) = \sigma(A)$. ∎

We obtain as an immediate corollary the classical characterization of the periodic throughput (see Formula (0.0.b)):

2.6.6 Corollary *Let $N(c)$ and $T(c)$ denote the sum of initial markings and the sum of the holding times of a circuit c of a strongly connected timed event graphs. The periodic throughput λ is characterized by*

$$\lambda = \min_c \frac{N(c)}{T(c)}$$

where the min is taken over all the elementary circuits of the graph.

3 Coding resources with multivariate polynomials

3.1 The $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i, \delta]]$ dioid

In the preceding sections, the quantities of resources $q_1, \ldots, q_p$ were supposed to be fixed. Now, we consider $q_1, \ldots, q_p$ as indeterminates. This means that operators of the form $\gamma^{q_1}, \ldots, \gamma^{q_p}$ occur in the $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$ representation of the timed event graph (see for instance Formula (2.3.e)). Thus, we need handling in an efficient way polynomials (and series) with symbolic exponents. This leads us to introducing *additional indeterminates* $\gamma_1, \ldots, \gamma_p$: γ^{q_i} will be represented by γ_i. More formally, we introduce the following extension of $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$:

3.1.1 Definition ($\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_0, \ldots, \gamma_p, \delta]]$). *We shall denote by $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_0, \ldots, \gamma_p, \delta]]$ the dioid of all the sums of the form*

$$\bigoplus_{(n,t)\in F} \gamma_0^{n_0} \ldots \gamma_p^{n_p} \delta^t, \qquad F \subset \mathbb{Z}^{p+2}$$

together with the simplification rules

$$\forall i \in \{0, \ldots, p\}, \quad \gamma_i^r \oplus \gamma_i^m = \gamma_i^{\min(r,m)}$$
$$\delta^s \oplus \delta^t = \delta^{\max(s,t)} \ .$$

When it will be clear from the context that it is a dioid of formal series with $p+2$ indeterminates, we shall write $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i, \delta]]$ more simply. The $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$ dioid is obtained by specialization to $p = 0$ (with $\gamma = \gamma_0$). Causal series and polynomials are defined in a way similar to $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma, \delta]]$. The following solves the "word problem" for $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i, \delta]]$ (this is analogous to the Proposition 2.2.2).

3.1.2 Proposition *$\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i, \delta]]$ can be equivalently defined as the quotient of $\mathbb{B}[[\gamma_i, \delta]]$ by the congruence*

$$s \simeq s' \iff s\gamma_0^* \ldots \gamma_p^* (\delta^{-1})^* = s'\gamma_0^* \ldots \gamma_p^* (\delta^{-1})^* \ .$$

3.1.3 Example In $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_0, \gamma_1, \delta]$, we have $\gamma_0\gamma_1 \oplus \delta = \delta$, for $\gamma_0\gamma_1 \preceq e \preceq \delta$.

We next provide a geometric representation of $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i,\delta]]$. The support map of the series

$$s = \bigoplus_{n\in\mathbb{Z}^{p+1},t\in\mathbb{Z}} s_{nt}\gamma_0^{n_0}\dots\gamma_p^{n_p}\delta^t \in \mathbb{B}[[\gamma_i,\delta]]$$

is now defined by

$$\operatorname{supp} s = \{(n_0,n_1,\dots,n_p,t)\in\mathbb{Z}^{p+2} \mid s(n_0,\dots,n_p,t)\neq\varepsilon\} \enspace .$$

This yields an isomorphism of dioids from $\mathbb{B}[[\gamma_i,\delta]]$ to $\mathcal{P}(\mathbb{Z}^{p+2})$. It remains to introduce the following polyhedral cone which extends the "south-east" cone $\mathcal{C}$ of $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]]$:

$$\mathcal{C} = \mathbb{R}^+(1,0,\dots,0,0)+\dots+\mathbb{R}^+(0,\dots,0,1,0)+\mathbb{R}^+(0,\dots,0,-1)+ = (\mathbb{R}^+)^{p+1}\times\mathbb{R}^- \enspace . \qquad (3.1.a)$$

3.1.4 Corollary *Two series $s,s'\in\mathbb{B}[[\gamma_i,\delta]]$ belong to the same coset in $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i,\delta]]$ iff*

$$\operatorname{supp} s+\mathcal{C} = \operatorname{supp} s'+\mathcal{C} \enspace .$$

In other words, we can identify $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i,\delta]]$ to the dioid of subsets of $\mathbb{R}^{p+2}$ of the form $A+\mathcal{C}$ with $A\subset\mathbb{Z}^{p+2}$, equipped with union and vector sum. The unit of this dioid is equal to $\mathcal{C}$.

3.1.5 Example We have represented on Figure 5 the subset $\operatorname{supp} p+\mathcal{C}\subset\mathbb{R}^3$ associated with $p=\gamma_0\delta\oplus\gamma_1^2\delta\oplus\gamma_0^2\gamma_1^2\delta$. With each monomial $m=\gamma_0^{n_0}\gamma_1^{n_1}\delta^t$ is associated the cone $(n_0,n_1,t)+\mathcal{C}$ with $\mathcal{C}=(\mathbb{R}^+)^2\times\mathbb{R}^-$.

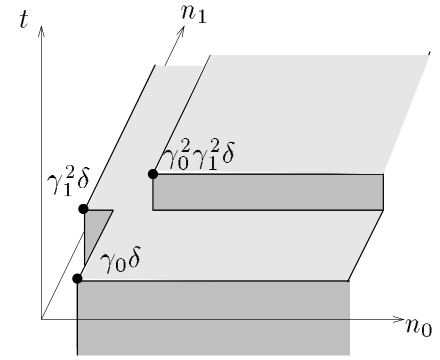

Figure 5: Representation of $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_0,\gamma_1,\delta]$

The fact that γ_i "represents" γ^{q_i} means algebraically that for all quantities of resources $q_1,\dots,q_p\geq 0$, we have an "evaluation" morphism

$$\phi_q : \begin{cases} \mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i,\delta]] & \longrightarrow & \mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma,\delta]] \\ \gamma_0 & \longmapsto & \gamma \\ \gamma_i & \longmapsto & \gamma^{q_i} \end{cases}$$

3.1.6 Counter example The quantities of resources must be nonnegative: it should be noted that ϕ_q is no more a dioid morphism for negative q. E.g., if $q=(q_1)$ with $q_1=-2$ we have $e=e\oplus\gamma_1$ in $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_0,\gamma_1,\delta]$ but $\phi_q(e)=e\neq\phi_q(e)\oplus\phi_q(\gamma_1)=e\oplus\gamma^{-2}=\gamma^{-2}$.

3.2 Slopes and periodic throughputs for multivariate polynomials

3.2.1 Definition (Slope in $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i, \delta]]$). *We define the slope $\sigma(s)$ of the series*

$$s = \bigoplus_i \gamma_0^{n_0^i} \gamma_1^{n_1^i} \dots \gamma_p^{n_p^i} \delta^{t^i} \in \mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i, \delta]]$$

from the slope σ already defined in $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma, \delta]]$ by

$$\sigma(s) \stackrel{\text{def}}{=} \sigma(\phi_q(s)) = \inf_{t^i > 0} \frac{n_0^i + n_1^i q_1^i + \dots + n_p^i q_p^i}{t^i} \ .$$

Thus, it is clear that $\sigma(s)$ does not depend on the representative $\bigoplus_i \gamma_0^{n_0^i} \gamma_1^{n_1^i} \dots \gamma_p^{n_p^i} \delta^{t^i}$. Moreover, the properties of the slope in $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma, \delta]]$ (and in particular (2.5.7,iii)) extend to $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_i, \delta]$. With the above definition of σ, is is clear from 2.6.4 that the symbolic periodic throughput of the autonomous system with matrix $A \in (\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma, \delta])^{n \times n}$ is equal to

$$\lambda(q) = \sigma(\bigoplus_c \mathtt{w}_A(c)) \tag{3.2.b}$$

where the sum is taken over all the (elementary) circuits of the graph associated with A.

3.2.2 Remark Set $q_0 = 1$, $F = \operatorname{supp} s$ and let $(\cdot|\cdot)$ denote the usual scalar product in $\mathbb{R}^{p+1}$. We have

$$\sigma(s) = \inf_{(n,t) \in F, \ t > 0} \frac{\langle q, n \rangle}{t} \ .$$

Thus, the slope appears as a non homogeneous extension of the classical support map which occurs in convex analysis [29],

$$\delta_F^\star(q) \stackrel{\text{def}}{=} \sup_{n \in F} \langle n, q \rangle$$

($\delta_F^\star$ denotes the convex conjugate of the convex indicator function of F, nothing to do with the shift δ). It is well known that the support of F only depends of its convex hull. Here, the slope only depends of the polyhedral cone generated by F.

3.2.3 Example Consider the system described by the following matrix with entries in $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma, \gamma_1, \delta]$:

$$A = \begin{bmatrix} \gamma\delta & \gamma_1 \\ \gamma\delta^{10} & \gamma^3\delta^3 \end{bmatrix}$$

By applying (3.2.b) we obtain the following expression of the periodic throughput in function of q_1:

$$\lambda(q_1) = \min(\frac{1}{1}, \frac{3}{3}, \frac{q_1 + 1}{10}) \ , \tag{3.2.c}$$

where the 3 terms of the minimum correspond to circuits (1), (2) et (12).

4 Constraint enumeration

We next give some algebraic algorithms which allow computing the periodic throughput of matrices with entries in $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_i, \delta]$.

4.1 Elementary paths

Recall that the graph associated with the $n \times n$ matrix A has been defined in 2.6.2. A path $p = (i_1, \dots, i_{k+1})$ is called *elementary* if $i_1, \dots, i_{k+1}$ are distinct. A circuit $c = (i_1, \dots, i_k)$ is elementary if $i_1, \dots, i_k$ are distinct. We shall denote by $\mathcal{P}_{ij}$ the set of elementary paths from j to i, and by $\mathcal{C}_i$ the set of elementary circuits with some vertex equal to i.

4.1.1 Definition **(A^{el}, $\pi(A)$, A^{e}).** *We define:*

- *The matrix of elementary paths* A^{el}

$$i \neq j : \quad A^{\mathsf{el}}_{ij} \stackrel{\text{def}}{=} \bigoplus_{p \in \mathcal{P}_{ij}} \mathsf{w}_A(p)$$

$$A^{\mathsf{el}}_{ii} \stackrel{\text{def}}{=} \bigoplus_{c \in \mathcal{C}_i} \mathsf{w}_A(c) \ .$$

- *The sum of the weights of the elementary circuits:*

$$\pi(A) \stackrel{\text{def}}{=} \mathrm{tr}(A^{\mathsf{el}}) = \bigoplus_{c \in \cup_i \mathcal{C}_i} \mathsf{w}_A(c) \ .$$

- A^{e} *the matrix which coincides with* A^{el} *outside the diagonal and with the identity on the diagonal.*

The following properties can be easily proved [12].

4.1.2 Properties *We have*

i $\pi(A \oplus B) \succeq \pi(A) \oplus \pi(B)$

ii *if* A *is triangular,* $\pi(A) = \mathrm{tr}(A)$

iii *If* D *is diagonal,* $\pi(D \oplus A') = \mathrm{tr}(D) \oplus \pi(A')$

iv $\pi(A) \preceq \pi(A^+) \preceq (\pi(A))^+$

4.1.3 Example Let $\mathcal{A}_k = (a_{ij})_{1 \leq i,j \leq k}$ denote the generic square matrix of size k. We have $\mathcal{A}_1^{\mathsf{el}} = (a_{11})$, $\mathcal{A}_1^{\mathsf{e}} = (e)$,

$$\mathcal{A}_2^{\mathsf{el}} = \begin{bmatrix} a_{11} \oplus a_{12}a_{21} & a_{12} \\ a_{21} & a_{22} \oplus a_{21}a_{12} \end{bmatrix},$$

$$\mathcal{A}_2^{\mathsf{e}} = \begin{bmatrix} e & a_{12} \\ a_{21} & e \end{bmatrix},$$

$$\pi(\mathcal{A}_2) = a_{11} \oplus a_{22} \oplus a_{12}a_{21} \ , \qquad (4.1.a)$$

$$\mathcal{A}_3^{\mathsf{el}} = \begin{bmatrix} \begin{array}{c} a_{11} \oplus a_{12}a_{21} \oplus a_{13}a_{31} \\ \oplus a_{12}a_{23}a_{31} \oplus a_{13}a_{32}a_{21} \end{array} & a_{12} \oplus a_{13}a_{32} & a_{13} \oplus a_{12}a_{23} \\ a_{21} \oplus a_{23}a_{32} & \begin{array}{c} a_{22} \oplus a_{21}a_{12} \oplus a_{23}a_{32} \\ \oplus a_{23}a_{31}a_{12} \oplus a_{21}a_{13}a_{31} \end{array} & a_{23} \oplus a_{21}a_{13} \\ a_{31} \oplus a_{32}a_{21} & a_{32} \oplus a_{31}a_{12} & \begin{array}{c} a_{33} \oplus a_{31}a_{13} \oplus a_{32}a_{23} \\ \oplus a_{12}a_{23}a_{31} \oplus a_{13}a_{32}a_{21} \end{array} \end{bmatrix}$$

$\pi(\mathcal{A}_3^{\mathsf{el}}) = a_{11} \oplus a_{22} \oplus a_{33} \oplus a_{12}a_{21} \oplus a_{13}a_{31} \oplus a_{23}a_{32} \oplus a_{12}a_{23}a_{31} \oplus a_{13}a_{32}a_{21}$.

It should be clear that computing naively such matrices becomes desperate for large n (e.g. $\pi(\mathcal{A}_6)$ is the sum of 415 terms, and $\pi(\mathcal{A}_n)$ is the sum of about $\exp(1)\times(n-1)!$ monomials). However, we do not really need computing these matrices. It turns out that it is simpler to design some algebraic algorithms which compute the elementary paths together with non elementary ones (which are inessential for computing $\lambda(q)$). This leads us to introduce the following notions:

4.1.4 Definition *We say that the matrix* $\mathfrak{E}$ enumerates the paths *if*

$$A^{\mathsf{e}} \preceq \mathfrak{E} \preceq A^{+} \ .$$

We say that $\mathfrak{E}$ enumerates the circuits *if*

$$\mathrm{diag}(A^{\mathsf{el}}) \preceq \mathrm{diag}(\mathfrak{E}) \preceq \mathrm{diag}(A^{+})$$

and that the scalar $\mathfrak{e}$ *enumerates the circuits if* $\pi(A) \preceq \mathfrak{e} \preceq \pi(A)^{+}$.

In other words, $\mathfrak{e}$ must contain all the elementary circuits (given by $\pi(A)$), but it may also contain all the inessential products[4] of circuits contained in $\pi(A)^{+}$.

4.1.5 Example Here are obvious example of $n\times n$ matrices which enumerate the paths and the circuits:

$$B=\mathrm{Id}\oplus A\oplus\ldots\oplus A^{n},\quad C=(\mathrm{Id}\oplus A)^{2^{k}} \text{ with } 2^{k}\geq n \ .$$

Hence, $\mathrm{tr}(B)$ and $\mathrm{tr}(A)$ enumerate the circuits.

The interest of these notions arises from the following results:

4.1.6 Proposition *For all autonomous system with representation* $A\in(\mathcal{M}^{\mathrm{in}}_{\mathrm{ax}}[\gamma_i,\delta])^{n\times n}$*, we have for any* $\mathfrak{e}$ *enumerating the circuits:*

$$\lambda=\sigma(\mathfrak{e}) \ .$$

Proof We have, by definition of $\mathfrak{e}$, $\pi(A)\preceq\mathfrak{e}\preceq\pi(A)^{+}$, hence, using the anti-isotony of the slope map and the fact that $\sigma(b^{+})=\sigma(b)$: $\lambda=\sigma(\pi(A))\geq\sigma(\mathfrak{e})\geq\sigma((\pi(A))^{+})=\lambda$. ∎

Thus, computing the periodic throughput reduces to computing enumerating matrices. We have the following algorithm inspired by the "zero square *gerbier*" of Benzaken [3] and from Gondran et Minoux [16],Chapter 3, Algorithm 3':

4.1.7 Algorithm *Consider the sequence of* $n\times n$ *matrices:*

$$A^{(0)}=A$$

$$A^{(k)}_{ij}=\begin{cases} A^{(k-1)}_{ij}\oplus A^{(k-1)}_{ik}A^{(k-1)}_{kj} & \textit{if } i\neq k \textit{ and } j\neq k\\ A^{(k-1)}_{ij} & \textit{if } i=k \textit{ or } j=k.\end{cases}$$

Then, the matrix $A^{(n)}$ *enumerates the paths and circuits.*

The main difference with the Jordan algorithm 2.6.1 is that the stars of the pivots which would produce non elementary paths have been cancelled.

Given a matrix A and $I,J\subset\{1,\ldots,n\}$, we shall denote by $A_{[I|J]}$ the submatrix with row indices in I and column indices in J. The following property is the counterpart of the classical "Schur complement formula"

$$\det\begin{bmatrix} A_{[I|I]} & A_{[I|J]}\\ A_{[J|I]} & A_{[J|J]}\end{bmatrix}=\det A_{[I|I]}\ \det(A_{[J|J]}-A_{[J|I]}A^{-1}_{[I|I]}A_{[I|J]}) \ . \qquad (4.1.b)$$

[4] Recall that if $C=\bigoplus_i C_i$, then $C^{+}=C\oplus C^{2}\oplus\ldots$ is equal to the sum of all possible products of the C_i.

4.1.8 Proposition *Consider a partition $\{1,\ldots,n\} = I \cup J$. For all matrix $\mathfrak{E}_{[I|I]}$ which enumerates the paths of $A_{[I|I]}$, the polynomial*

$$c = \pi(A_{[I|I]}) \oplus \pi\left(A_{[J|J]} \oplus A_{[J|I]}\ \mathfrak{E}_{[I|I]}\ A_{[I|J]}\right) \tag{4.1.c}$$

enumerates the circuits of A.

Proof Let us consider the entries of A as noncommuting indeterminates[5]. Then the weight of the path $p = (i_1,\ldots,i_{k+1})$ is equal to the monomial $a_{i_1 i_2}\ldots a_{i_k i_{k+1}}$. Since there is a one to one correspondence between paths and monomials, we shall identify a path to its weight. Let $B = A_{[J|J]} \oplus A_{[J|I]}\mathfrak{E}_{[I|I]}A_{[I|J]}$. We have to show that B_{kl} is a sum of paths from l to k. Indeed,

$$B_{kl} = A_{kl} \oplus \bigoplus_{r,s\in I} A_{kr}\mathfrak{E}_{rs}A_{sl} \quad . \tag{4.1.d}$$

It follows from the definition of $\mathfrak{E}$ that $\mathfrak{E}_{rs}$ is a sum of paths from s to r, hence, $A_{kr}\mathfrak{E}_{rs}A_{sl}$ is a path from l to k: this shows that $B_{kl} \preceq A^+_{kl}$. Hence, $\pi(B) \preceq \pi(A^+) \preceq \pi(A)^+$ (by 4.1.2). This implies that $c \preceq \pi(A)^+$.

To prove that $c \succeq \pi(A)$, it remains to show that all the elementary circuits are contained in (4.1.c). This is left to the reader (see also [12]). ∎

If we only want to compute $\pi(A)$ but do not require computing enumerating matrices, we have an improved version of Algorithm 4.1.7 which involves matrices of decreasing size.

4.1.9 Algorithm *Define the matrices $A^{(i)}$ and the scalars π_i as follows:*

$$\begin{aligned}
A^{(0)} &= A \quad \pi_0 = \mathrm{tr}A \\
A^{(k)}_{ij} &= A^{(k-1)}_{ij} \oplus A^{(k-1)}_{ik}A^{(k-1)}_{kj} \quad \text{if } i \neq j \text{ and } i,j > k \\
A^{(k)}_{ii} &= A^{(k-1)}_{ik}A^{(k-1)}_{ki} \\
\pi_k &= \pi_{k-1} \oplus \mathrm{tr}A^{(k)}_{[k+1\ldots n|k+1\ldots n]}
\end{aligned}$$

Then, the polynomial π_{n-1} enumerates the circuits.

This is analogous to computing determinants by Gaussian triangularization.

Proof of the algorithm. It follows from 4.1.2,(iii) (with $D = \mathrm{diag}(A)$, $A'_{ij} = A_{ij}$ outside the diagonal and ε otherwise), together with the bloc elimination given in 4.1.8 with the order 1 pivot $\mathfrak{E} = e$. ∎

4.1.10 Example For the matrix of Example 3.2.3, we obtain

$$\pi_0 = \gamma\delta \oplus \gamma^3\delta^3, \quad a^{(1)} = \begin{bmatrix} \gamma\delta & \gamma_1 \\ \gamma\delta^{10} & \gamma\delta^{10}\gamma_1 \end{bmatrix},$$

$$\pi_1 = \pi_0 \oplus a^{(1)}_{22} = \gamma\delta \oplus \gamma^3\delta^3 \oplus \gamma\delta^{10}\gamma_1 \quad . \tag{4.1.e}$$

Let us recall that the *permanent* function is given by

$$\mathrm{perm}\, A = \bigoplus_{\sigma\in S_n} \bigotimes_{i=1}^{n} A_{i\sigma(i)}$$

where S_n denotes the group of permutations of $\{1,\ldots,n\}$. The following result, whose version for determinants is well know to combinatorialists [27], shows that enumerating polynomials generalize to some extent characteristic polynomials:

[5] In other words, we work in the dioid of formal series with boolean coefficients $\mathbb{B}\langle\langle a_{ij}\rangle\rangle$

4.1.11 Proposition *The polynomial* perm $(\mathrm{Id} \oplus A)$ *enumerates the circuits of* A.

4.1.12 Remark This last point gives some metaphysical reason for which products become sums when we pass from the standard version of the Schur complement formula (4.1.b) to the dioid one (4.1.c): $\pi(A)$ is the analogous of $\det(\mathrm{Id} \oplus A)$ and we have $(e \oplus a)(e \oplus b) = e \oplus a \oplus b \oplus \ldots$, that is, $x \mapsto e \oplus x$ acts locally[6] as a morphism from the additive structure to the multiplicative one.

4.1.13 Example By applying algorithm 4.1.9 to the matrix A (see (2.3.e)) representing the flexible workshop described in §1.1, we obtain the following enumerating polynomial:
$c = \delta^{11}\gamma \oplus \delta^{12}\gamma_1 \oplus \delta^{19}\gamma_2 \oplus \delta^{29}\gamma\gamma_1 \oplus \delta^{30}\gamma\gamma_2 \oplus \delta^{33}\gamma^2\gamma_2 \oplus \delta^{40}\gamma^2\gamma_1 \oplus \delta^{42}\gamma\gamma_1\gamma_2 \oplus \delta^{53}\gamma^2\gamma_1\gamma_2 \oplus \delta^{62}\gamma^3\gamma_1\gamma_2$
which is a sum of 10 monomials. Hence, the periodic throughput is equal to the $\min$ of the 10 following affine functions:
$\lambda(q) = \min(1/11, \frac{q_1}{12}, \frac{q_2}{19}, 1/29+\frac{q_1}{29}, 1/30+\frac{q_2}{30}, \frac{2}{33}+\frac{q_2}{33}, 1/20+\frac{q_1}{40}, 1/42+\frac{q_1}{42}+\frac{q_2}{42}, \frac{2}{53}+\frac{q_1}{53}+\frac{q_2}{53}, \frac{3}{62}+\frac{q_1}{62}+\frac{q_2}{62})$

We observe that the above expression in redundant (i.e. that some affine terms can be dropped without changing the value of $\lambda(q)$). E.g. let us consider

$$u = \min(\frac{1}{11}, \frac{1}{29} + \frac{1}{29}q_1), \qquad v = \frac{1}{20} + \frac{q_1}{40} \enspace .$$

It is easily checked that $\forall q_1 \geq 0$, we have $v \geq u$. Hence, we can drop v from the expression of λ. After some similar simplifications, we obtain the formula (1.1.a).

The fact that these enumeration algorithms produce redundant terms lead to a combinatorial increase of the size of polynomials, which makes these algorithms ineffective as soon as the size of matrices is greater than a few units. In the remaining part of this paper, we show that some *simplification rules* are valid. This allows reducing the size of the polynomials occurring in the Gaussian algorithms defined above.

4.2 Some polyhedral congruences

We have already noticed that at each step of the Gaussian algorithm, the sum of the pivots π_k is a sum of weights of circuits of the graph. We next show that this knowledge allows us to apply some simplifications rules.

We associate with a polynomial

$$c = \bigoplus_{i=1}^{\ell} c_i \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i, \delta], \qquad c_i = \gamma_0^{n_0^i} \ldots \gamma_p^{n_p^i} \delta^{t_i}$$

the polyhedral cone

$$\mathcal{C}_c \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} (\operatorname{supp} c_i) \times \mathbb{R}^+ = \{\sum_{i=1}^{\ell} x_i(n_0^i, \ldots, n_p^i, t^i) \mid x \in (\mathbb{R}^+)^\ell\} \enspace .$$

4.2.1 Definition *We call* congruence modulo the polyhedral cone $\mathcal{C}_c$ *the relation* $\mathcal{R}_c$ *defined over* $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i, \delta]]$ *by:*

$$x\mathcal{R}_c y \iff \operatorname{supp} x + \mathcal{C} + \mathcal{C}_c = \operatorname{supp} y + \mathcal{C} + \mathcal{C}_c \enspace .$$

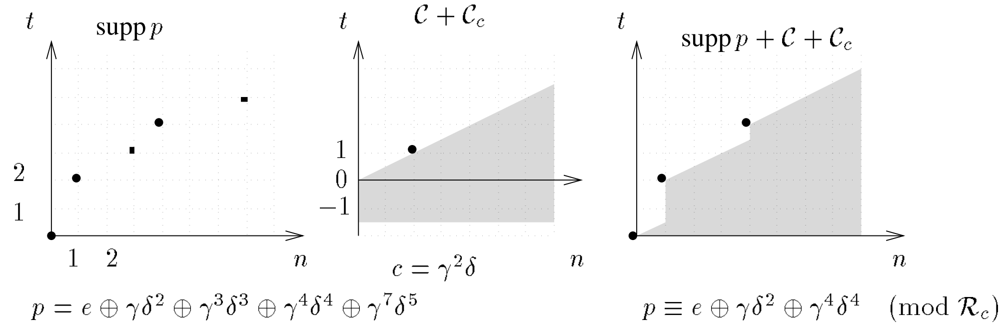

Figure 6: Polyhedral congruence in dimension 2

4.2.2 Example We have represented on Figure 6 the polynomial $p = e \oplus \gamma\delta^2 \oplus \gamma^3\delta^3 \oplus \gamma^4\delta^4 \oplus \gamma^7\delta^5$ (black circles and squares), and its minimal representative modulo $\mathcal{R}_{\gamma^2\delta}$, $p_0 = e \oplus \gamma\delta^2 \oplus \gamma^4\delta^4$ (blacks circles only). The set supp $p + \mathcal{C} + \mathcal{C}_c$ is represented in grey. The unique minimal representative p_0 is obtained by taking the extremal points of this subset.

In the following we shall denote by $\oplus$ the $\min$ of the slopes.

4.2.3 Lemma *Assume that $\pi(A) \succeq c$. For all matrix A' which is the same coset (componentwise) as A modulo $\mathcal{R}_c$, we have $\sigma(\pi(A')) \oplus \sigma(c) = \sigma(\pi(A)) \oplus \sigma(c) = \sigma(A)$.*

The assumption that $\pi(A) \succeq c$ means that a subset of circuits associated with c is already known. It immediately follows that:

4.2.4 Theorem *Each step of the Gaussian algorithm can be performed modulo the polyhedral congruence associated with the circuits already enumerated.*

For instance, in algorithm 4.1.9, we have the following decreasing chain of dioids (the arrows denote canonical projections):

$$\begin{array}{cl}
\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_i,\delta]/\mathcal{R}_{\text{tr}(A)} & \text{step } 0 \\
\downarrow & \\
\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_i,\delta]/\mathcal{R}_{\pi_1} & \text{step } 1 \\
\downarrow & \\
\vdots & \\
\downarrow & \\
\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_i,\delta]/\mathcal{R}_{\pi_{n-2}} & \text{step } n-1.
\end{array}$$

Each step of the algorithm brings some more knowledge about the circuits, and thus, induces additional simplification rules. We think that it is enough to consider the following heuristic argument and the following examples to understand this result.

4.2.5 Example We consider a matrix A with entries in $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma,\delta]$ (this is the case $p = 0$). We assume that at a certain step of the Gaussian algorithm 4.1.9, we have $A_{ij}^{(k)} = e \oplus \gamma^2\delta^2$ and that the circuit $c = \gamma\delta^3$ is already known. Observe that $e \oplus \gamma^2\delta^2 \mathcal{R}_c e$ because $\gamma^2\delta^2 \preceq e \otimes c^2$. The polynomial C which gives the sum of all the circuits obtained at the end of the algorithm can be written as

$$C = c \oplus uA_{ij}^{(k)} \oplus v \qquad (4.2.f)$$

[6] for us, it means that the small perturbation consists in adding non elementary circuits.

where u and v are polynomials. We have

$$\sigma(C) = \min(\sigma(c), \sigma(uA_{ij}^{(k)}), \sigma(v)) \ . \tag{4.2.g}$$

Let us assume for simplicity that u is a monomial, i.e. $u = \gamma^n\delta^t$. Then

$$\sigma(uA_{ij}^{(k)}) = \min(\sigma(\gamma^n\delta^t), \sigma(\gamma^2\delta^2\gamma^n\delta^t)) \ .$$

We have due to 2.5.7,iii: $\sigma(\gamma^2\delta^2\gamma^n\delta^t) \geq \min(\sigma(\gamma^2\delta^2), \sigma(\gamma^n\delta^t)) \geq \min(\sigma(c), \sigma(\gamma^n\delta^t))$. This shows that $\sigma(\gamma^n\delta^t\gamma^2\delta^2)$ is redundant in (4.2.g). Hence, we may have replaced $A_{ij}^{(k)} = e \oplus \gamma^2\delta^2$ by its minimal representative modulo $\mathcal{R}_c$, $A_{ij}^{'(k)} = e$ without modifying the final value of $\sigma(C)$.

4.2.6 Example Consider the periodic throughput associated with the matrix:

$$A = \begin{bmatrix} \gamma\delta & \gamma \oplus \gamma^4\delta \\ \gamma\delta^{10} \oplus \gamma^2\delta^{11} & \gamma^3\delta^3 \end{bmatrix} \ . \tag{4.2.h}$$

The diagonal circuits are immediately enumerated, hence:

$$c = \gamma\delta \oplus \gamma^3\delta^3 \ .$$

Modulo the congruence associated with c, we have $\gamma \oplus \gamma^4\delta \equiv \gamma$ (because $\gamma^4\delta \preceq \gamma(\gamma\delta)$), and $\gamma\delta^{10} \oplus \gamma^2\delta^{11} \equiv \gamma\delta^{10}$. Hence, the following matrix has the same periodic throughput as A.

$$A' = \begin{bmatrix} \gamma\delta & \gamma \\ \gamma\delta^{10} & \gamma^3\delta^3 \end{bmatrix} \ .$$

Hence

$$\lambda = \sigma(A'_{11} \oplus A'_{22} \oplus A'_{21}A'_{12}) = \sigma(\gamma\delta \oplus \gamma^3\delta^3 \oplus \gamma^2\delta^{10}) = \frac{1}{5} \ .$$

If we apply directly Formula (4.1.a) to the matrix (4.2.h) we obtain

$$\begin{aligned} \pi(A) &= \gamma\delta \oplus \gamma^3\delta^3 \oplus (\gamma \oplus \gamma^4\delta)(\gamma\delta^{10} \oplus \gamma^2\delta^{11}) \\ &= \gamma\delta \oplus \gamma\delta^{10} \oplus \gamma^3\delta^{11} \oplus \gamma^6\delta^{12} \end{aligned} \tag{4.2.i}$$

which yields of course the same $\lambda = \sigma(A)$. The important point here is that some redundant monomials have been simplified *after* the enumeration in the naive algorithm (4.2.i) and *before* with the polyhedral congruence. Hence, for large matrices, only the second approach is possible.

4.2.7 Example Let $p = \gamma_0\delta \oplus \gamma_0\gamma_1\delta^2 \in \mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_0, \gamma_1, \delta]$, and $c = \gamma_1^2\delta^6$. We have $p \ \mathcal{R}_c \ \gamma_0\delta$ because $\gamma_0\gamma_1\delta^2 \preceq \gamma_0\delta(c)^{\frac{1}{2}}$, i.e. in $\mathbb{R}^3$,

$$\begin{aligned} &(2,1,1) = (1,1,0) + \frac{1}{2}(6,0,2) + 2(-1,0,0) \\ &\quad \subset (1,1,0) + \mathbb{R}^+(6,0,2) + \mathbb{R}^+(-1,0,0) + \mathbb{R}^+(0,1,0) + \mathbb{R}^+(0,0,1) \ . \end{aligned}$$

Proof of Lemma 4.2.3. In the following, we shall identify monomials to points of $\mathbb{R}^{p+2}$, according to the bijection:

$$\gamma_0^{n_0} \ldots \gamma_p^{n_p}\delta^t \xrightarrow{\imath} (n_0, \ldots, n_p, t) \in \mathbb{R}^{p+2} \ . \tag{4.2.j}$$

The map $\imath$ is extended to polynomials by additivity. The power m^s of a monomial m (the exponent s can be noninteger) is by definition

$$m^s \stackrel{\text{def}}{=} \imath^{-1}(s \times \imath(m))$$

i.e.

$$(\gamma_0^{n_0} \dots \gamma_p^{n_p} \delta^t)^s \stackrel{\text{def}}{=} \gamma_0^{s\times n_0} \dots \gamma_p^{s\times n_p} \delta^{s\times t} \quad .$$

In the following, we shall omit the $\imath$ sign when no confusions arises. The key of the result is the following simple remark:

4.2.8 Lemma *For all monomials a,b, for all $s, r \geq 0$,*

$$\sigma(a^s b^r) \preceq \sigma(a) \oplus \sigma(b) = \min(\sigma(a), \sigma(b)) \quad .$$

With the usual notation, (4.2.8) becomes:

$$\sigma(s \times a + r \times b) \geq \min(\sigma(a), \sigma(b))$$

(recall that for slopes, $u \preceq v \iff u \oplus v = \min(u, v) = v$, i.e. $u \geq v$). This is nothing but a simple inequality which is left to the reader.

Let us define the following preorder relation:

$$a \preceq_c b \iff \operatorname{supp} a \subset \operatorname{supp} b + \mathcal{C} + \mathcal{C}_c$$

The transitivity of $\preceq_c$ follows from $\mathcal{C} + \mathcal{C} = \mathcal{C}$ and $\mathcal{C}_c + \mathcal{C}_c = \mathcal{C}_c$. Observe that $a \preceq_c b$ and $b \preceq_c a$ iff $a\mathcal{R}_c b$. We naturally extend $\preceq_c$ to matrices (componentwise).

4.2.9 Lemma $A' \preceq_c A \Rightarrow \sigma(A') \preceq \sigma(A) \oplus \sigma(c)$.

Let us assume that 4.2.9 holds and that $A'\mathcal{R}_c A$. Then $A' \preceq_c A$, hence,

$$\sigma(A') \oplus \sigma(c) \preceq \sigma(A) \oplus \sigma(c)$$

and by symmetry,

$$\sigma(A') \oplus \sigma(c) = \sigma(A) \oplus \sigma(c) = \sigma(A)$$

because $\pi(A) \succeq c$. This shows Lemma 4.2.3.

Proof of Lemma 4.2.9. Assume that $A' \preceq_c A$. Let $\alpha = (i_1, \dots, i_m)$ be a circuit (with the convention $i_{m+1} = i_1$). We have $A'_{i_v i_{v+1}} \preceq_c A_{i_v i_{v+1}}$ hence, taking the product for $v = 1 \dots m$ and using the compatibility of $\preceq_c$ with product, we get $\mathsf{w}_{A'}(\alpha) \preceq_c \mathsf{w}_A(\alpha)$. We may clearly assume $\mathsf{w}_{A'}(\alpha)$ is reduced to single monomial (i.e. a single point). We thus have

$$\mathsf{w}_{A'}(\alpha) \preceq \mathsf{w}_A(\alpha)\, c_1^{s_1} \dots c_\ell^{s_\ell} \tag{4.2.k}$$

for some $s \in (\mathbb{R}^+)^\ell$. Indeed, with conventional notation of $\mathbb{R}^{p+2}$, (4.2.k) rewrites

$$\mathsf{w}_{A'}(\alpha) \in \mathsf{w}_A(\alpha) + s_1 \times c_1 + \dots + s_\ell \times c_\ell + \mathcal{C} \quad .$$

We have from (4.2.k):

$$\sigma(\mathsf{w}_{A'}(\alpha)) \preceq \sigma(\mathsf{w}_A(\alpha)\, c_1^{s_1} \dots c_\ell^{s_\ell}) \quad .$$

Adding $\sigma(c)$ to this inequality and using 4.2.8, we get

$$\sigma(\mathsf{w}_{A'}(\alpha)) \oplus \sigma(c) \preceq \sigma(\mathsf{w}_A(\alpha)) \oplus \sigma(c) \quad .$$

Taking the summation over all the elementary circuits α, we get

$$\sigma(\pi(A')) \oplus \sigma(c) \preceq \sigma(\pi(A)) \oplus \sigma(c) = \sigma(\pi(A)) \quad .$$

This concludes the proof of 4.2.3. ∎

4.3 Fundamental example

We consider again the flexible workshop represented in Figure 1 and corresponding to the matrix

$$a = \begin{bmatrix} \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \gamma_1\delta & \varepsilon \\ \delta^9 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma_2\delta^{10} \\ \delta^9 & \varepsilon & \varepsilon & \gamma\delta^7 & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \gamma\delta^{10} \\ \varepsilon & \varepsilon & \varepsilon & \delta^7 & \delta & \varepsilon \end{bmatrix}$$

We apply the enumeration algorithm 4.1.9. We only write at each step the useful part of the matrix $a^{(k)}$:

$$a^{(0)} = a \qquad \pi^{(0)} = \varepsilon$$

$$a^{(1)} = \begin{bmatrix} \gamma\delta^{11} & \varepsilon & \varepsilon & \gamma_1\delta^{10} & \gamma^2\delta^{10} \\ \gamma\delta^{11} & \varepsilon & \gamma\delta^7 & \gamma_1\delta^{10} & \varepsilon \\ \delta^2 & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \varepsilon & \varepsilon & \gamma\delta^{10} \\ \varepsilon & \varepsilon & \delta^7 & \delta & \varepsilon \end{bmatrix} \qquad \pi^{(1)} = \gamma\delta^{11}$$

$$a^{(2)} = \begin{bmatrix} \varepsilon & \gamma\delta^7 & \gamma_1\delta^{10} & \gamma\gamma_1\delta^{21} \\ \delta^2 & \varepsilon & \gamma_1\delta^{12} & \gamma_2\delta^{12} \\ \delta^2 & \varepsilon & \varepsilon & \gamma\delta^{10} \\ \varepsilon & \delta^7 & \delta & \varepsilon \end{bmatrix} \qquad \pi^{(2)} = \gamma\delta^{11}$$

Let us detail the computation of $a^{(2)}_{35}$. We have $a^{(2)}_{35} = a^{(1)}_{35} \oplus a^{(1)}_{32}a^{(1)}_{25} = \gamma_1\delta^{10} \oplus \gamma\delta^{11}\gamma_1\delta^{10} \equiv \gamma_1\delta^{10}$ $(\bmod\ \mathcal{R}_{\pi^2})$, because $\gamma\delta^{11}\gamma_1\delta^{10} \preceq \gamma_1\delta^{10}\pi^2$. A similar argument shows that

$$a^{(3)} = \begin{bmatrix} \gamma\delta^9 & \gamma_1\delta^{12}(\oplus\gamma_1\delta^{12}) & \gamma_2\delta^12(\oplus\gamma\gamma_2\delta^{23}) \\ \gamma\delta^9 & \gamma_1\delta^{12} & \gamma\delta^{10}\oplus\gamma\gamma_2\delta^{23} \\ \delta^7 & \delta & \varepsilon \end{bmatrix} \qquad \pi^{(3)} = \gamma\delta^{11}\oplus\gamma_1\delta^{12}$$

(parenthesis refer to the inessential terms).

$$a^{(4)} = \begin{bmatrix} \gamma_1\delta^{12}(\oplus\gamma\gamma_1\delta^{21}) & \gamma\delta^{10}\oplus\gamma\gamma_2\delta^{23}(\oplus\gamma\gamma_2\delta^{21}) \\ \delta\oplus\gamma_1\delta^{19} & \gamma_2\delta^{19} \end{bmatrix} \qquad \pi^{(4)} = \gamma\delta^{11}\oplus\gamma_1\delta^{12}\oplus\gamma_2\delta^{19}$$

$a^{(5)} = (\delta\oplus\gamma_1\delta^{19})(\gamma\delta^{10}\oplus\gamma\gamma_2\delta^{23}) = \gamma\delta^{11}\oplus\gamma\gamma_1\delta^{22}\oplus(\gamma\gamma_2\delta^{24})\oplus\gamma\gamma_2\delta^{42}$, $\pi^{(5)} = \gamma\delta^{11}\oplus\gamma_1\delta^{12}\oplus\gamma_2\delta^{19}\oplus\gamma\gamma_1\delta^{29}$. Hence, we obtain the final minimal expression of λ:

$$\lambda = \sigma(\pi^{(5)}) = \min(\frac{1}{11}, \frac{q_1}{12}, \frac{1+q_1}{29}, \frac{q_2}{19}) \ , \tag{4.3.1}$$

as it was claimed in the introduction.

4.4 Effective computation modulo polyhedral congruences

We shall not give here the algorithms needed to make effective the polyhedral congruences: these are studied in detail in [12] and have been implemented in `MAPLE` and `SCILAB`. We just give some idea of the tools needed in order to perform simplifications and of the complexity that we may expect.

Let us recall that monomials of $\mathcal{M}_{\rm ax}^{\rm in}[\gamma_i,\delta]$ are identified to points of $\mathbb{R}^{p+2}$ according to the correspondence (4.2.j). It should be clear that the simplification rule

$$a\oplus b=b \quad \text{in } \mathcal{M}_{\rm ax}^{\rm in}[\gamma_i,\delta]/\mathcal{R}_c \tag{4.4.m}$$

holds iff we have

$$a\in b+\mathcal{C}+\mathcal{C}_c \quad \text{in } \mathbb{R}^{p+2}. \tag{4.4.n}$$

In other words, it is enough to determine if the point a belongs to the polyhedral cone $b+\mathcal{C}+\mathcal{C}_c$. Assume that the polynomial c is a sum of ℓ monomials $c_1,\ldots,c_\ell$. With the identification between points and monomials, we have

$$\mathcal{C}_c=\sum_{i=1}^{\ell} c_i\times\mathbb{R}^+ \ .$$

The test (4.4.n) can be performed by using standard LP algorithms, since this is equivalent to determine if the following set of constraints admits an admissible point $y\in(\mathbb{R}^+)^l$:

$$\begin{array}{l}\forall i, 1\leq i\leq p+1,\ \ \sum_{j=1}^{\ell}(c_j)_i y_j+b_i\leq a_i\\ \sum_{j=1}^{\ell}(c_j)_{p+2}y_j+b_{p+2}\geq a_{p+2}\ \ .\end{array}$$

Then, the unique minimal representative of polynomials modulo $\mathcal{R}_c$ can be obtained by an inductive use of the rewriting rule (4.4.m) (see [12]).

Since simplification modulo polyhedral congruences involves solving linear programs in dimension $p+2$, the number p of unknown resources must of course be small. A case of particular interest is $p=0$ (no unknown resources): then, simplification modulo $\mathcal{R}_c$ has a pseudo-polynomial complexity. More precisely, we have the following results:

4.4.1 Proposition ([12]). *Let p,q,c be causal polynomials in $\mathcal{M}_{\rm ax}^{\rm in}[\gamma,\delta]$. The computation of the minimal form of $p\oplus q$ modulo $\mathcal{R}_c$ requires* $O((\max(\deg_\gamma p,\deg_\gamma q))^2)$ elementary comparisons. *The computation of pq requires $O((\deg_\gamma p+\deg_\gamma q)^2)$ elementary comparisons.*

4.4.2 Corollary ([12]). *Let $A\in(\mathcal{M}_{\rm ax}^{\rm in}[\gamma,\delta])^{n\times n}$ and $\deg_\gamma A=\max_{ij}\deg_\gamma A_{ij}$. The algorithms 4.1.7 and 4.1.9, performed modulo 4.2.4, require $O(n^5(\deg_\gamma A)^2)$ elementary comparisons.*

We next describe some useful aggregation methods.

5 Some refinements

5.1 Aggregation

Consider a partition $\{1,\ldots,n\}=I\cup J$ of the set of vertices of the graph. The matrix A writes

$$A=\begin{bmatrix} A_{II} & A_{IJ}\\ A_{JI} & A_{JJ}\end{bmatrix} \ .$$

We get via 4.1.8

$$\pi(A)=\pi(A_{II})\oplus\pi\left(A_{JJ}\oplus A_{JI}A'_{II}A_{IJ}\right) \tag{5.1.a}$$

for all matrix A'_{II} which enumerates the paths of A_{II}. This can be seen as an aggregation result. If the matrix A_{II} is non symbolic (i.e. if it only contains the indeterminates γ et δ), the enumerated matrix A'_{II}

is easily obtained. Thus, the complexity of the computation is related with the size of the residual matrix $A_{JJ} \oplus A_{JI}A'_{II}A_{IJ}$, which corresponds to the number of resources. In other words, we can deal with timed event graphs of great size as soon as the number of indeterminate resources is small.

We illustrate this approach by obtaining a closed formula for a workshop with 2 parts and n identical machines.

5.2 Periodic throughput of a flowshop with 2 part types an n identical machines

We consider the flowshop at the left side of Figure 7, analogous to that of §4.3. That is, we assume that the machine i processes part 1 during a units of time, then part 2 during b u.t., and that a et b are independent of i. Initially, q_j pallets reserved for type j parts ($j = 1, 2$) and wait for machine 1 to start.

5.2.1 Theorem *The periodic throughput of the Flowshop with n identical machines and two part types given on Figure 7,(a) is equal to:*

$$\lambda(q_1, q_2) = \min(\frac{1}{a+b}, \frac{q_1}{na}, \frac{q_2}{nb}, \frac{1+\min(q_1, q_2)}{2a+2b+(n-2)\max(a,b)}) \ . \tag{5.2.b}$$

Proof We have with the notation of Figure 7:

$$\begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} = \begin{bmatrix} \varepsilon & \gamma\delta^b \\ \delta^a & \varepsilon \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} \oplus \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \oplus \begin{bmatrix} \gamma_1\delta^a & \varepsilon \\ \varepsilon & \gamma_2\delta^b \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} \ , \tag{5.2.c}$$

$$i > 1 : \quad \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} = \begin{bmatrix} \varepsilon & \gamma\delta^b \\ \delta^a & \varepsilon \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} \oplus \begin{bmatrix} \delta^a & \varepsilon \\ \varepsilon & \delta^b \end{bmatrix} \begin{bmatrix} x_{1i-1} \\ x_{2i-1} \end{bmatrix} \ .$$

$$y_1 = \delta^a x_n, \quad y_2 = \delta^b x_n,$$

We perform an ad hoc elimination of the variables x_{1i}, x_{2i} for $i = 2 \ldots n$, which is indeed equivalent to

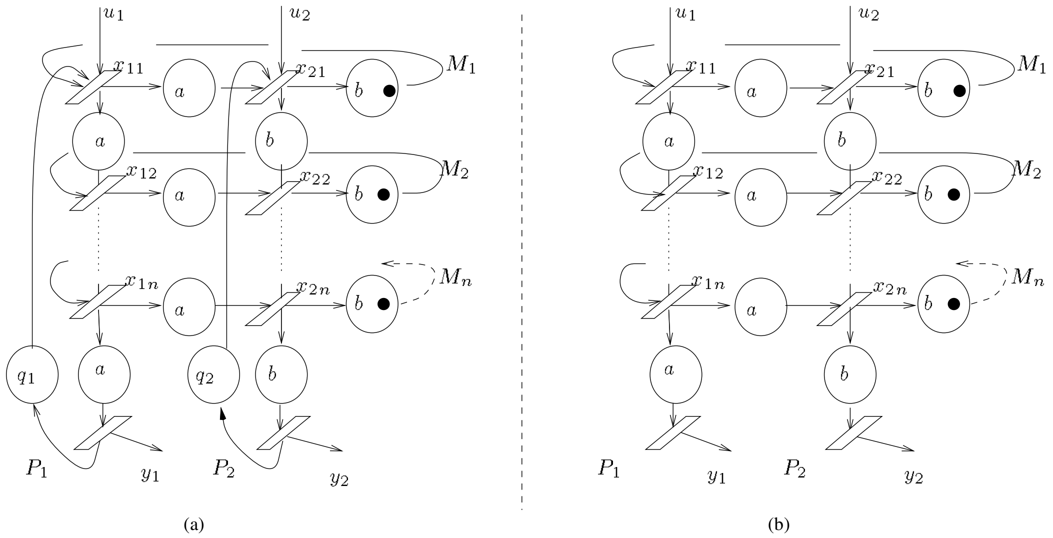

Figure 7: Flexible workshop with n identical machines

algorithm (5.1.a). It is sufficient to compute the transfer H from u to y of the open loop system at the right of Figure 7,(b). Then, the transfer H' of the closed loop system will be obtained by solving:

$$X_1 = DY \oplus U, \ \ Y = HX_1, \quad \text{with} \ \ D = \text{diag}(\gamma_1, \gamma_2)$$

i.e.

$$H' = H(DH)^* \ .$$

For simplicity, we shall write dates as scalars i.e. $\gamma^k\delta^a \simeq a\gamma^k$. Then, a simple computation gives the transfer between machines $i-1$ and i:

$$X_i = \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} = (ab\gamma)^* \begin{bmatrix} a & ab\gamma \\ ab & b \end{bmatrix} \begin{bmatrix} x_{1i-1} \\ x_{2i-2} \end{bmatrix} \ .$$

We thus have enumerated a first circuit:

$$\pi_1 = ab\gamma^* \ .$$

Modulo the congruence associated with π_1, we have $(ab\gamma)^* = e$, and we can write

$$X_i = \mathcal{H}X_{i-1} \text{ with } \mathcal{H} = \begin{bmatrix} a & ab\gamma \\ ab & b \end{bmatrix} \begin{bmatrix} x_{1i-1} \\ x_{2i-2} \end{bmatrix}$$

Hence:

$$Y = \mathcal{H}^n X_1 \ .$$

An easy computation shows that

$$\mathcal{H}^2 = \begin{bmatrix} a^2 \oplus a^2b^2\gamma & ab(a\oplus b)\gamma \\ ab(a \oplus b) & b^2 \oplus a^2b^2\gamma \end{bmatrix}$$

$$\mathcal{H}^3 = \begin{bmatrix} a^3 \oplus a^2b^2(a\oplus b)\gamma & ab(a\oplus b)^2\gamma \oplus a^3b^3\gamma^2 \\ ab(a\oplus b)^2 \oplus a^3b^3\gamma & b^3 \oplus a^2b^2(a\oplus b)\gamma \end{bmatrix}$$

Modulo the congruence associated with π_1, we have

$$ab(a\oplus b)^2 \oplus a^3b^3\gamma = ab(a\oplus b)^2$$

because $a^3b^3\gamma \preceq ab(a\oplus b)(ab\gamma)$. This gives the simplified matrix

$$\mathcal{H}^3 = \begin{bmatrix} a^3 \oplus a^2b^2(a\oplus b)\gamma & ab(a\oplus b)^2\gamma \\ ab(a\oplus b)^2 & b^3 \oplus a^2b^2(a\oplus b)\gamma \end{bmatrix}$$

and more generally

$$\mathcal{H}^n = \begin{bmatrix} a^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma & ab(a\oplus b)^{n-1}\gamma \\ ab(a\oplus b)^{n-1} & b^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma \end{bmatrix} \ .$$

Finally

$$\lambda(q_1,q_2) = \sigma \begin{bmatrix} \gamma_1(a^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma) & \gamma_1 ab(a\oplus b)^{n-1}\gamma \\ \gamma_2 ab(a\oplus b)^{n-1} & \gamma_2(b^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma) \end{bmatrix} \oplus \sigma(ab\gamma) \ .$$

Hence, $\lambda = \sigma(\pi_1) \oplus \sigma(\pi)$ where π is given by 4.1.3:

$$\pi = \gamma_1 a^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma\gamma_1 \oplus \gamma_2 b^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma\gamma_2 \oplus a^2b^2(a\oplus b)^{2n-2}\gamma\gamma_1\gamma_2 \ .$$

Assume for instance that $a \succeq b$. Then, $\gamma_1 a^n \otimes a^2b^2(a\oplus b)^{n-2}\gamma\gamma_2 \succeq a^2b^2(a\oplus b)^{2n-2}$. This shows that the last term of π is inessential. A similar argument holds if $a \preceq b$. This gives the final formula

$$\pi_1 \oplus \pi = ab\gamma \oplus \gamma_1 a^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma\gamma_1 \oplus \gamma_2 b^n \oplus a^2b^2(a\oplus b)^{n-2}\gamma\gamma_2$$

∎

5.3 Complexity estimation

The algorithm 4.2.4 is relevant when the number p of indeterminate resources is small, and when the size of the graph is large, The preceding section exhibits a limit case with two indeterminate resources and a graph of arbitrary size. More generally, let us consider a system as shown on Figure 8. diag(γ_i) denotes a $p \times p$

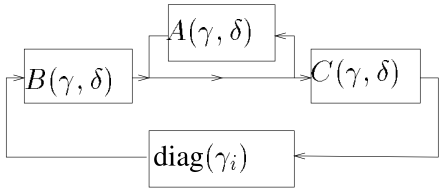

Figure 8: Standard form of a system with p indeterminate resources

diagonal matrix whose diagonal entries are $\gamma_i,\ i = 1, \ldots, p$. A is an $n \times n$ matrix with entries in $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma, \delta]$, which represent some large part of the graph with constant resources. B and C denote two matrices with monomial entries in γ and δ, so that the indeterminate resources are concentrated in the matrix diag(γ_i). The reader will easily convince himself that we can always reduce to this standard form, perhaps after adding some transitions. Applying (5.1.a) -with $A_{II} \stackrel{\text{def}}{=} A$, $A_{IJ} \stackrel{\text{def}}{=}$ diag$(\gamma_i)C$, $A_{JI} \stackrel{\text{def}}{=} B$, $A_{JJ} = \varepsilon$-, we are led to enumerating the circuits of $A'' \stackrel{\text{def}}{=} BA'$diag$(\gamma_i)C$, where the matrix A' enumerates the paths of A. From 4.4.2, computing A'' requires $O(n^5(\deg_\gamma A)^2)$ operations. The number of monomials of each entry of A'' is bounded above by $n^2 \deg_\gamma A$. It remains to enumerate the circuits of A''. Since A'' is produced by some elimination algorithm, it is likely to be full, and we have $O((p-1)!)$ elementary circuits, each of them being sum of at most $(n^2 \deg_\gamma A)^p$ monomials. Hence, the total number of monomials (without simplifications) of the final enumerating polynomial will be $O((p-1)!(n^2 \deg_\gamma A)^p)$. The only interest of this very coarse bound is to show that given a fixed number of unknown resources p, the complexity is a polynomial function of the size of the graph n.

6 Spectral approach

In this section, we develop a completely different point of view about resource optimization. We have previously characterized the periodic throughput λ as the “pole” of the transfer matrix. We have also seen that it could be computed from some “characteristic polynomial” (see 4.1.11), which suggests that it is related with some $(\max, +)$ spectral theory. Indeed, it can be shown that λ is equal to the *eigenvalue* of the dynamic matrix of the $(\min, +)$ linear recurrent representation of the system. We shall not develop this point of view here, and refer the reader to [1]. We next show how the throughput constraints $\lambda(q) \geq \lambda_0$ can be formulated as a potential constraint (existence of a sub-eigenvector) in the $(\min, +)$ algebra.

6.1 Sub-eigenvectors

In this section, we give an alternative approach which is completely elementary, and for which a simpler notation can be used.

6.1.1 Notation T_{ij} denotes the sojourn times in place[7] (ij). N_{ij} denotes the initial marking (number of tokens) in this place. If this initial marking is known, N_{ij} is a constant. Otherwise, the place (ij) corresponds

[7] i.e. the place downstream transition j and upstream transition i

to some indeterminate resource l, and we set $N_{ij} = q_l$, where q_l is the indeterminate quantity of resource l. We shall write $N(q)$ instead of N to emphasize this dependency. We assume that the firings of transitions is instantaneous, and that the graph to be strongly connected. We shall denote by $\lambda(q)$ the periodic throughput as a symbolic expression of the resources.

We need introducing the $(\min, +)$ algebra, i.e. the set $\mathbb{R} \cup \{+\infty\}$ equipped with $\min$ (denoted by $\oplus$) and addition (denoted by $\otimes$). This is a dual structure of the $\overline{\mathbb{R}}_{\max}$ dioid described above. The zero element is $\varepsilon \stackrel{\text{def}}{=} +\infty$. Observe that by comparison with $\overline{\mathbb{R}}_{\max}$, we have withdrawn the $-\infty$ element, so that every non ε element is invertible. This is essential in order to develop a Perron-Frobenius like spectral theory. Let us recall that the natural order relation in $\mathbb{R}_{\min}$

$$a \preceq b \iff a \oplus b = \min(a, b) = b \iff a \geq b$$

is precisely the dual of the usual order. The *spectral radius* $\rho(C)$ [15, 1, 8, 9] of a $n \times n$ matrix C with entries in $\mathbb{R}_{\min}$ is defined by

$$\rho(C) = \min_{1 \leq k \leq n} \min_{i_1, \dots, i_k} \frac{C_{i_1 i_2} + C_{i_2 i_3} + \ldots + C_{i_k i_1}}{k} \tag{6.1.a}$$

$$= \bigoplus_{k=1}^{n} \left(\operatorname{tr}(A^k)\right)^{\frac{1}{k}} \ . \tag{6.1.b}$$

Here is the main theorem:

6.1.2 Theorem *The following assertions are equivalent:*

i $\lambda(q) \geq \lambda_0$

ii $\rho(C(q)) \preceq e$ *in* $\mathbb{R}_{\min}$, *where the matrix* $C(q)$ *is defined by* $C_{ij}(q) = N_{ij}(q) - \lambda_0 T_{ij}$.

iii there exists a vector u *with entries in* $\mathbb{R}_{\min}$ *such that* $C(q) \otimes u \preceq u$.

Condition (iii) rewrites as follows in the conventional algebra:

$$\forall i, \quad \min_j (C_{ij}(q) + u_j) \geq u_i \ ,$$

i.e.

$$\forall i, j, \quad N_{ij}(q) - \lambda_0 T_{ij} + u_j \geq u_i \ . \tag{6.1.c}$$

We observe that the constraints which appear in (6.1.c) are linear functions of u_i and q_l. Hence, we get from 6.1.2:

6.1.3 Corollary *The resource optimization problem of §1.3 is equivalent to the following linear programming problem with integer and real variables:*

$$\min\{J(q) \mid q \in \mathbb{N}^p, \ \exists u \in \mathbb{R}^n, \ \forall i, j, \ N_{ij}(q) - \lambda_0 T_{ij} + u_j \geq u_i\} \ .$$

The proof is based on the following *sub-eigenvector* result which is reminiscent of Wielandt's proof of the Perron-Frobenius theorem (as it is told for instance in [11, 24, 30]).

6.1.4 Lemma (sub-eigenvector). *Let* $A \in (\mathbb{R}_{\min})^{n \times n}$ *be irreducible, let* $\lambda \in \mathbb{R}_{\max}$. *The following assertions are equivalent:*

i there exists $u \in \mathbb{R}^n_{\min} \backslash \{\varepsilon\}$ *such that* $Au \preceq \lambda u$, *ii* $\rho(A) \preceq \lambda$.

This is just one half of the spectral theorem for dioids (see [15, 9]). However, we think that it is useful to include the very simple proof.

Proof of Lemma 6.1.4. We only prove the result when $\lambda \neq \varepsilon$ (the case $\lambda = \varepsilon$ is almost obvious).

(i)⇒(ii). We have

$$\forall k, \quad A^k u \preceq \lambda^k u,$$

hence

$$\forall i, k \quad A^k_{ii} u_i \preceq \lambda^k u_i$$

and since all the u_i are different from ε (easy proof), we get after cancellation

$$\forall i, k \ \ (A^k_{ii})^{\frac{1}{k}} \preceq \lambda \ . \tag{6.1.d}$$

Summing all these inequalities:

$$\rho(A) = \bigoplus_{k=1}^{n} (\mathrm{tr} A^k)^{\frac{1}{k}} = \bigoplus_{i=1}^{n} (A^k_{ii})^{\frac{1}{k}} \preceq \lambda \ . \tag{6.1.e}$$

(ii)⇒(i) Conversely, let $\tilde{A} = \lambda^{-1} A$. Since $\rho(\tilde{A}) = \lambda^{-1}\rho(A) \preceq e$, $\tilde{A}^*$ is finite. Since $\tilde{A}\tilde{A}^* = \tilde{A}^+ \preceq \tilde{A}$, any column of $\tilde{A}^*$ is a subeigenvector. ∎

Proof of Theorem 6.1.2. (i)⇔(ii). Given the matrix B, recall that the the weight $\mathsf{w}_B(\alpha)$ of the circuit $\alpha = (i_1, \ldots, i_k)$ is defined by

$$\mathsf{w}_B(\alpha) = B_{i_1 i_2} + \ldots + B_{i_k i_1} \ .$$

The periodic throughput writes (see 2.6.6):

$$\lambda(q) = \min_{\alpha \text{ circuit}} \frac{\mathsf{w}_N(\alpha)}{\mathsf{w}_T(\alpha)} \ . \tag{6.1.f}$$

The following assertions are clearly equivalent:

	$\lambda(q) \geq \lambda_0$	
For all circuit α,	$\frac{\mathsf{w}_N(\alpha)}{\mathsf{w}_T(\alpha)} \geq \lambda_0$	(by (6.1.f))
For all circuit α,	$\mathsf{w}_N(\alpha) \geq \lambda_0 \mathsf{w}_T(\alpha)$	
For all circuit α,	$\mathsf{w}_N(\alpha) - \lambda_0 \mathsf{w}_T(\alpha) \geq 0$	
For all circuit α,	$\mathsf{w}_C(\alpha) \geq 0$	
	$\rho(C) \geq 0.$	(from (6.1.b)).

We have shown (i)⇔(ii). The equivalence of (ii) and (iii) is an immediate consequence of the sub-eigenvector Lemma 6.1.4. ∎

This extends to general timed event graphs and arbitrary linear cost function the result given by Proth and Xie [26, 20] for a class of invariant[8] cost functions and for a subclass of timed event graphs[9].

[8] i.e. invariant by firing of transitions.

[9] times are only put on transitions and it is assumed that all the transitions are recycled.

6.2 Concluding remarks

Let us compare the different approaches that we have discussed. When the number p of unknown resources is of the same size than the number of places of the graph, the linear programming approach 6.1.3 has to be retained. When the number of unknown resources is small with respect to the size of the graph, the aggregation technique described in §5.1 gives a linear programming problem of type (1.3.b) with a smaller number of unknowns (equal to the number of resources): in this case, the symbolic approach of §5.1 seems preferable. For instance, in [12], we have considered a more realistic example of flowshop with 8 machines and 6 part types (hence 48 transitions), for which the final expression of the periodic throughput involved only 25 constraints (although the intermediate simplifications took about half an hour of `MAPLE` and `SCILAB` on a Sun Sparcstation 2). In the current state of the art (and of the implementation), it is possible to compute the symbolic throughput of timed event graphs with at most 6 or 7 indeterminate resources and reasonable size (in some cases, this size can be almost arbitrary large, due to the aggregation techniques that we have described). Some efficient algorithms might also be designed by combining the symbolic and iterative approach. For instance, it should be possible to consider a unique resource q_i as an indeterminate (the others being given arbitrary fixed values), then to compute the periodic throughput $\lambda(q_i)$ (which is simple because there is a unique resource) and then to optimize $J(q)$ with respect only to q_i. Then, we might optimize with respect to another resource q_j ($j \neq i$), and so on. This class of algorithms remains to be explored.

References

[1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. Wiley, 1992.

[2] R.C. Backhouse and B.A. Carré. Regular algebra applied to path finding problems. *J. of the Inst. of Maths and Appl.*, 15:161–186, 1975.

[3] C. Benzaken. Structures algébrique des cheminements: pseudo-treillis, gerbiers de carré nul. In G. Biorci, editor, *Network and Switching Theory*, pages 40–47. Academic Press, 1968.

[4] J. Carlier and P. Chretienne. *Problèmes d'Ordonnancement: modélisation, complexité, algorithmes*. Masson, Paris, 1988.

[5] P. Chretienne. *Les Réseaux de Petri Temporisés*. Thèse Université Pierre et Marie Curie (Paris VI), Paris, 1983.

[6] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. Analyse du comportement périodique des systèmes de production par la théorie des dioïdes. Rapport de recherche 191, INRIA, Le Chesnay, France, 1983.

[7] G. Cohen, P. Moller, J.P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *IEEE Proceedings: Special issue on Discrete Event Systems*, 77(1), Jan. 1989.

[8] R.A. Cuninghame-Green. *Minimax Algebra*. Number 166 in Lectures notes in Economics and Mathematical Systems. Springer, 1979.

[9] P. Dudnikov and S. Samborskiĭ. Endomorphisms of semimodules over semirings with an idempotent operation. *Math. in USSR, Izvestija*, 38(1), 1992. translation of Izv. Akad. Nauk SSSR Ser. Mat. 55, 1991.

[10] S. Eilenberg and M.P. Schützenberger. Rational sets in commutative monoids. *J. Algebra*, 13:173–191, 1969.

[11] F.R. Gantmacher. *Theorie des Matrices, volume II*. Dunod, 1966.

[12] S. Gaubert. *Théorie des systèmes linéaires dans les dioïdes*. Thèse, École des Mines de Paris, July 1992.

[13] S. Gaubert and C. Klimann. Rational computation in dioid algebra and its application to performance evaluation of discrete event systems. In G. Jacob and F. Lamnabhi-Lagarrigue, editors, *Algebraic computing in control*, number 165 in Lecture Notes in Control and Inf. Sci. Springer, 1991.

[14] G. Gierz, K.H. Hofmann, K. Keimel, J.D Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer, 1980.

[15] M. Gondran and M. Minoux. Valeurs propres et vecteurs propres en théorie des graphes. In *Problèmes combinatoires et théorie des graphes*, number 260 in Colloques internationaux CNRS, Orsay, 1976.

[16] M. Gondran and M. Minoux. *Graphes et algorithmes*. Eyrolles, Paris, 1979. Engl. transl. *Graphs and Algorithms*, Wiley, 1984.

[17] H.P. Hillion and J.M. Proth. Performance evaluation of job-shop systems using timed event-graphs. *IEEE Trans. on Automatic Control*, 34(1):3–9, Jan 1989.

[18] R.M. Karp. A characterization of the minimum mean-cycle in a digraph. *Discrete Maths.*, 23:309–311, 1978.

[19] S.C. Kleene. Représentation of events in nerve nets and finite automata. In C. Shannon J. Mc Carthy, editor, *Automata studies*. Princeton University Press, 1956.

[20] S. Laftit, J.M. Proth, and X.L. Xie. Optimization of invariant criteria for event graphs. *IEEE Trans. on Automatic Control*, 37(6):547–555, 1992.

[21] L. Lawler. Optimal cycles in doubly weighted directed graphs. In *Journées internationales d'étude sur la théorie des graphes*, pages 209–213, Rome, 1966.

[22] V. Maslov and S. Samborskiĭ, editors. *Idempotent analysis*, volume 13 of *Adv. in Sov. Math.* AMS, RI, 1992.

[23] V.P. Maslov. *Méthodes Operatorielles*. Mir, Moscou, 1973. trad. fr. 1987.

[24] H. Minc. *Nonnegative matrices*. Wiley, 1988.

[25] M. Plus. Linear systems in $(\max, +)$-algebra. In *Proceedings of the 29th Conference on Decision and Control*, Honolulu, Dec. 1990.

[26] J.M. Proth and X.L. Xie. Les critères invariants dans un graphe d'événements déterministes. *CRAS, Série I*, 313:797–800, 1991.

[27] C. Reutenauer. Ensembles libres de chemins dans un graphe. *Bull. Soc. Math. Fr.*, 114(2):135–152, 1986.

[28] C. Reutenauer. *Aspects mathématiques des réseaux de Petri*. Masson, Paris, 1989.

[29] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, New Jersey, 1970.

[30] E. Seneta. *Non-negative matrices and Markov chains*. Springer series in statistics. Springer, 1981.