Chapitre IX

Calcul formel du taux de production et optimisation des ressources

Introduction

Nous considérons le problème dit "d'optimisation des ressources", qui consiste à minimiser le nombre de ressources (machines, palettes, processeurs, places de stocks, mémoires tampons ...) d'un système de manière à réaliser un taux de production fixé. Ce problème a d'abord été traité par Cohen, Dubois, Quadrat et Viot [17], qui présentent un algorithme itératif donnant *une* répartition minimale des ressources. Dans le cas particulier où le taux de production visé est le taux du processus le plus lent, on parvient de de la sorte à saturer le "processus goulot". Hillion et Proth [52] ont considéré un second problème analogue, qui consiste à minimiser un coût linéaire associé aux ressources, toujours sous la contrainte d'un taux de production donné. Ils ramènent ce problème à un problème de programmation linéaire en nombres entiers, dont les contraintes sont indicées par l'ensemble des circuits du graphe. Constatant que cette approche était rendue ineffective par l'impossibilité apparente d'énumérer les circuits (en trop grand nombre), ils formulent des heuristiques pour ce problème. Nous attaquons dans ce chapitre le seul problème d'énumération des contraintes du programme linéaire, dans l'esprit des techniques d'énumération algébrique des circuits d'un graphe dues à Benzaken, Backhouse & Carré, Gondran & Minoux [5, 4, 47]. Nous adaptons ainsi un algorithme de type Gaussien. La clé de la méthode consiste à introduire des congruences polyédriques, faisant appel à des algorithmes de programmation quadratique ou de programmation linéaire. On peut alors simplifier les données intermédiaires, et obtenir le simplexe final sous forme réduite. On obtient de la sorte l'expression formelle du taux de production en fonction des ressources. Nous avons appliqué cette méthode à l'atelier étudié en 1983 par Cohen, Dubois, Quadrat et Viot. Pour cet exemple réel à 8 machines et 6 pièces, le problème d'énumération des circuits, inabordable par l'algorithme de Gauss sans les congruences polyédriques, donne finalement le taux de production comme min de 25 contraintes linéaires non redondantes. On peut alors prouver simplement que l'allocation des ressources obtenue par l'algorithme itératif cité plus haut est l'*unique* allocation minimale. Proth et Xie [83] ont par ailleurs récemment montré que certains problèmes d'optimisation de ressources se ramenaient à la résolution d'un nouveau problème de programmation linéaire mixte, dont le nombre de contraintes est de l'ordre du nombre de places et non plus du nombre de circuits. L'algorithme de Proth et Xie n'est cependant valable que pour une sous classe assez restrictive de systèmes et de critères. Nous montrons que le problème d'optimisation des ressources est équivalent à la recherche d'un sous-vecteur propre, ce qui généralise l'algorithme de Proth et Xie à des systèmes quelconques. Nous concluons en comparant les deux approches.

1 Les problèmes d'optimisation des ressources

1.1 Exemple

On considère l'atelier flexible avec deux types de pièces et 3 machines représenté sur la figure IX.1. On donne la table suivante des temps de passage des pièces sur les machines:

temps de passage	pièce 1	pièce 2	charge totale
machine 1	9	2	11
machine 2	2	7	9
machine 3	1	10	11

L'atelier flexible est de type *flowshop*, i.e. les pièces sont supposées passer sur les machines dans le même ordre. Un système analogue a été étudié en VIII,4. La pièce 1 passe donc successivement sur les machines 1,2,3. Idem pour la pièce 2. On suppose que dans la configuration initiale, q_1 palettes réservées aux pièces 1 et q_2 palettes réservées aux pièces 2 sont plaçées en amont de la machine 1. Les deux machines 1 et 3 sont menantes, i.e. ont la charge maximale de 11 unités de temps. Quelle que soit l'allocation des palettes, on ne pourra donc jamais dépasser la fréquence des machines menantes, c'est-à-dire une pièce en moyenne toutes les 11 unités de temps. On cherche ici à minimiser les nombres q_1 et q_2 pour saturer les machines menantes M_1 et M_3. Nous montrerons dans ce chapitre qu'à l'aide d'un calcul algébrique simple (nous l'effectuons à la main dans la section §3.4), on obtient l'expression *formelle* du taux de production λ en fonction des ressources, soit en l'occurrence:

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

On notera que ces contraintes se lisent en termes de circuits du graphe. $\frac{q_1}{12}$ correspond au circuit (x_5, x_3, x_1), $\frac{1+q_1}{29}$ au circuit $(x_5, x_6, x_4, x_3, x_1)$. Pour saturer la machine menante, il faut que chacun des termes de λ soit plus grand que le taux de production maximal $\frac{1}{11}$, soit $\frac{q_1}{12} \geq \frac{1}{11}$, $\frac{1+q_1}{29} \geq \frac{1}{11}$, $\frac{q_2}{19} \geq \frac{1}{11}$, ce qui donne l'allocation entière minimale des palettes saturant les machines M_1 et M_3:

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

1.2 Premier problème

Soient $q_1, \dots, q_n$ les quantités des ressources inconnues, et $\lambda(q)$ le taux de production associé. L'application $q \mapsto \lambda(q)$ est évidemment croissante, et donc le taux de production $\lambda(q)$ est majoré par $\lambda(+\infty)$. On suppose que le taux de production maximal $\overline{\lambda} = \lambda(+\infty)$ est fini (dans l'exemple ci-dessus, il s'agit du taux de production des machines menantes, i.e. des processus sur lesquels les ressources n'ont pas d'effet). Le premier problème d'optimisation des ressources consiste à trouver une (ou toutes) les allocations des ressources saturant les processus menants. En d'autres termes, on cherche les vecteurs $q \in \mathbb{N}^n$ minimaux tels que

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

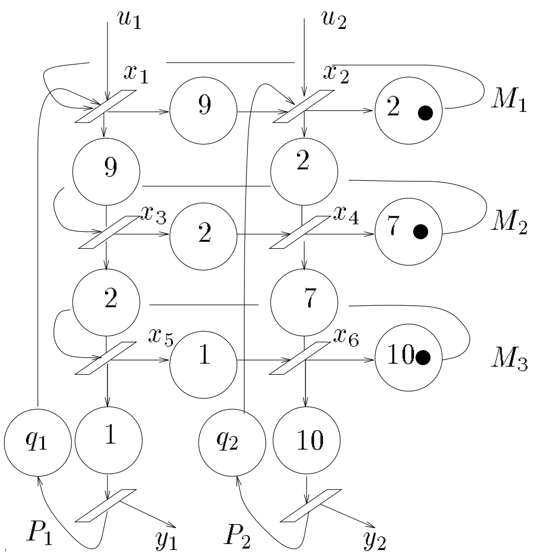

Figure IX.1: "Flowshop" 2 pièces 3 machines, nombre de palettes à optimiser

1.3 Second problème

Etant donné un taux de production souhaité λ_0, on cherche à minimiser un coût linéaire associé aux ressources nécessaires pour atteindre ce taux de production, soit le coût

$$J(q) = p_1 q_1 + \ldots + p_n q_n \ ,$$

où p_i désigne le prix d'une unité de ressource i (par exemple, le prix d'installation d'une machine ou d'une palette). On verra plus loin que $\lambda(q)$ est un min de formes affines. On a donc le problème de programmation linéaire en nombres entiers suivant

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

2 Polynômes multivariables et ressources

Nous introduisons un dioïde à $p+2$ indéterminées, utile pour l'étude des systèmes à p ressources inconnues. Ce dioïde est la généralisation naturelle de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$.

2.1 Le dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$

2.1.1 Définition (Dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_0,\ldots,\gamma_p,\delta]]$) *On note $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_0,\ldots,\gamma_p,\delta]]$ le dioïde quotient de $\mathbb{B}[[\gamma_0,\gamma_1,\ldots,\gamma_p,\delta]]$ par l'application*

$$\varphi : \varphi(a) = a\gamma_0^*\gamma_1^*\ldots\gamma_p^*(\delta^{-1})^* \ . \tag{2.1.a}$$

Lorsque qu'il sera clair au vu du contexte qu'il s'agit d'un dioïde de séries formelles à $p+2$ indéterminées, on s'autorisera la notation $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$.

En d'autres termes, $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$ est le dioïde des séries formelles à coefficients booléiens en les variables commutatives $\gamma_0,\gamma_1,\ldots,\gamma_p,\delta$, muni des règles de simplification suivantes:

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

Dans le cas $p=0$, on retrouve le dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$, On notera $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$ le sous-dioïde des polynômes. La théorie développée en B,1.2.10 s'applique. La relation associée à φ est une *bonne congruence*. En particulier, tout élément de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$ admet un unique représentant minimal modulo φ.

2.1.2 Exemple Dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_0,\gamma_1,\delta]$, on a $\gamma_0\gamma_1\oplus\delta=\delta$.

2.2 Représentation géométrique

On a vu que le dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$ des séries formelles en γ et δ est isomorphe au dioïde des "cônes sud-est" de $\mathbb{Z}^2$ (cf. VII,(2.2.c)). Nous donnons de même une représentation géométrique de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$ à l'aide d'un dioïde de parties de $\mathbb{R}^{p+2}$.

2.2.1 Dioïdes $\mathcal{P}(\mathbb{R}^{p+2}),\mathcal{P}(\mathbb{Z}^{p+2})$ On notera $\mathcal{P}(\mathbb{R}^{p+2})$ le dioïde des parties de $\mathbb{R}^{p+2}$, munies de l'union et de la somme vectorielle. L'application support, définie comme suit

$$\left\{\begin{array}{ccc}\mathbb{B}[[\gamma_0,\ldots,\gamma_p,\delta]] & \longrightarrow & \mathcal{P}(\mathbb{Z}^{p+2})\subset\mathcal{P}(\mathbb{R}^{p+2})\\ s & \longmapsto & \mathrm{supp}s=\{(n_0,n_1,\ldots,n_p,t)\in\mathbb{Z}^{p+2}\mid\ s(n_0,\ldots,n_p,t)\neq\varepsilon\}\end{array}\right.\tag{2.2.a}$$

est un isomorphisme de dioïdes.

2.2.2 Dioïdes $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^{p+2}]],\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{Z}^{p+2}]]$ Soit le cône de $\mathbb{R}^{p+2}$:

$$\mathcal{C}=\mathbb{R}^+(1,0,\ldots,0,0)+\ldots+\mathbb{R}^+(0,\ldots,0,1,0)+\mathbb{R}^+(0,\ldots,0,-1)+=(\mathbb{R}^+)^{p+1}\times\mathbb{R}^-\ .\tag{2.2.b}$$

On introduit l'application

$$\varphi_{\mathcal{C}}:\left\{\begin{array}{ccc}\mathcal{P}(\mathbb{R}^{p+2}) & \longrightarrow & \mathcal{P}(\mathbb{R}^{p+2})\\ x & \longmapsto & x+\mathcal{C}\end{array}\right.\ .\tag{2.2.c}$$

On notera $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^{p+2}]]$ le dioïde quotient de $\mathcal{P}(\mathbb{R}^{p+2})$ par $\varphi_{\mathcal{C}}$, et $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{Z}^{p+2}]]$ le sous dioïde quotient de $\mathcal{P}(\mathbb{Z}^{p+2})$ par la même application.

Comme l'application support est un isomorphisme de $\mathbb{B}[[\gamma_i,\delta]]$ sur $\mathcal{P}(\mathbb{Z}^{p+2})$, et que par ailleurs on a pour toutes séries $s,s'\in\mathbb{B}[[\gamma_i,\delta]]$ $\varphi(s)=\varphi(s')$ ssi $\mathrm{supp}s\ +\mathcal{C}=\mathrm{supp}s'\ +\mathcal{C}$ (cf. (2.1.a)), on a l'isomorphisme induit:

$$\left\{\begin{array}{ccc}\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]] & \simeq & \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{Z}^{p+2}]]\\ s & \mapsto & \mathrm{supp}s\end{array}\right.\ .\tag{2.2.d}$$

Il nous reste à caractériser $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{Z}^{p+2}]]$, et plus généralement, $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^{p+2}]]$.

2.2.3 Lemme *L'application κ définie par:*

$$\left\{\begin{array}{ccc}\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^{p+2}]] & \longrightarrow & \mathcal{P}(\mathbb{R}^{p+2})+\mathcal{C}\\ s & \longmapsto & \kappa(s)=s+\mathcal{C}\end{array}\right.\tag{2.2.e}$$

est un isomorphisme de dioïdes.

Preuve Le seul point non trivial est que $\kappa(s \otimes s') = \kappa(s) \otimes \kappa(s')$, soit $\kappa(s + s') = \kappa(s) + \kappa(s')$, ce qui résulte de $\mathcal{C} + \mathcal{C} = \mathcal{C}$. ∎

Autrement dit, on peut identifier $\mathcal{M}_{\text{ax}}^{\text{in}}[[\mathbb{R}^{p+2}]]$ au dioïde des parties de $\mathbb{R}^{p+2}$ de la forme $A + \mathcal{C}$, muni de l'union et de la somme vectorielle. L'élément neutre de ce dioïde est la partie $\mathcal{C}$. On peut voir $\mathcal{M}_{\text{ax}}^{\text{in}}[[\mathbb{Z}^{p+2}]]$ est le sous dioïde des parties de la forme $A + \mathcal{C}$, où $A \subset \mathbb{Z}$.

On composant (2.2.d) et 2.2.3, on obtient un nouvelle isomorphisme $\imath$ qui nous permet de représenter une série formelle par une partie de $\mathbb{R}^{p+2}$:

$$\imath : \left\{ \begin{array}{ccc} \mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i, \delta]] & \simeq & \mathcal{P}(\mathbb{Z}^{p+2}) + \mathcal{C} \subset \mathcal{P}(\mathbb{R}^{p+2}) \\ s & \mapsto & \imath(s) = \operatorname{supp} s + \mathcal{C} \end{array} \right. \tag{2.2.f}$$

2.2.4 Exemple On a représenté sur la Figure IX.2 la partie $\imath(p)$ de $\mathbb{R}^3$ associée à $p = \gamma_0\delta \oplus \gamma_1^2\delta \oplus \gamma_0^2\gamma_1^2\delta$. On associe à chaque monôme $m = \gamma_0^{n_0}\gamma_1^{n_1}\delta^t$ le cône $\imath(m) = (n_0, n_1, t) + \mathcal{C}$ avec $\mathcal{C} = (\mathbb{R}^+)^2 \times \mathbb{R}^-$, et l'on prend l'union de ces cônes pour obtenir $\imath(p)$.

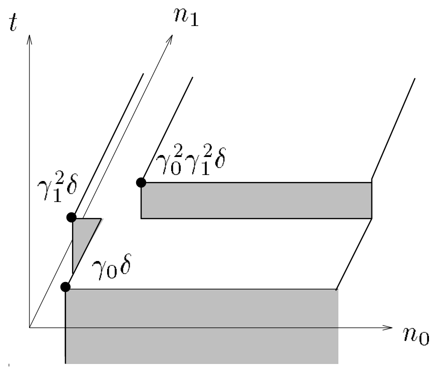

Figure IX.2: Représentation graphique de $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_0, \gamma_1, \delta]$

2.2.5 Remarque L'introduction de $\mathbb{R}^{p+2}$ au lieu de $\mathbb{Z}^{p+2}$ est motivée par les deux points suivants: 1/ dans les applications, on a en général des puissances réelles et non entières de γ et δ, 2/ on donnera dans la suite des règles de simplifications dans $\mathcal{M}_{\text{ax}}^{\text{in}}[[\gamma_i, \delta]]$, qui feront appel à des projections sur certains cônes polyédriques, lesquelles ne s'écrivent que dans $\mathbb{R}^{p+2}$ (ou à la rigueur dans $\mathbb{Q}^{p+2}$).

2.3 Ressources et polynômes

Etant données des quantités fixées de ressources $q_1, \ldots, q_p \geq 0$, on a un morphisme:

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

On pourra donc représenter chaque ressource par une nouvelle indéterminée γ_i. On pourra aussi identifier γ et γ_0 (que l'on peut voir comme indéterminée associée à une ressource constante).

2.3.1 Exemple L'atelier flexible décrit en §1.1 se représente par le système d'équations suivants dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\gamma_1,\gamma_2,\delta]]$:

$$x = Ax \oplus Bu, y = Cx$$

où

$$A = \begin{bmatrix} \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \gamma_1\delta & \varepsilon \\ \delta^9 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma_2\delta^{10} \\ \delta^9 & \varepsilon & \varepsilon & \gamma\delta^7 & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \gamma\delta^{10} \\ \varepsilon & \varepsilon & \varepsilon & \delta^7 & \delta & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}$$

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

2.3.2 Contre exemple ϕ_q n'est plus un morphisme de dioïdes si q n'est pas positif. Soit $q = (q_1)$ avec $q_1 = -2$. On a $e = e \oplus \gamma_1$ dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_0,\gamma_1,\delta]$ mais $\phi_q(e) = e \neq \phi_q(e) \oplus \phi_q(\gamma_1) = e \oplus \gamma^{-2} = \gamma^{-2}$.

D'où la restriction suivante.

2.3.3 Hypothèse *Dans la suite, on supposera que les ressources sont en quantités positives ou nulles, i.e.* $\forall i, q_i \geq 0$.

2.3.4 Pente dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$ On définira la pente $\sigma(s)$ d'une série $s = \bigoplus_i \gamma_0^{n_0^i}\gamma_1^{n_1^i}\dots\gamma_p^{n_p^i}\delta^{t^i} \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$ à partir de la pente σ déjà définie dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$ par

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

De la sorte, il est immédiat que $\sigma(s)$ ne dépend pas du représentant $\bigoplus_i \gamma_0^{n_0^i}\gamma_1^{n_1^i}\dots\gamma_p^{n_p^i}\delta^{t^i}$. Il est clair que les propriétés prouvées en VII,5.4.3 pour la pente dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$ s'étendent à $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$.

2.3.5 Taux de production dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma_i,\delta]]$ On a caractérisé en VII,6.6.1 le taux de production λ d'un système autonome représenté par une matrice $A \in (\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta])^{n\times n}$ par

$$\lambda = \sigma(\bigoplus_c w_A(c)) \tag{2.3.a}$$

où la somme est prise sur l'ensemble des circuits du graphe, ou de manière équivalente, sur l'ensemble des circuits élémentaires. La formule (2.3.a) s'étend à une matrice $A \in (\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta])^{n\times n}$ à condition de définir la pente σ comme en 2.3.4.

2.3.6 Exemple Soit le système décrit par la matrice suivante à coefficients dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\gamma_1,\delta]$:

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

On a en appliquant (2.3.a) l'expression suivante du taux de production en fonction de la quantité q_1:

$$\lambda(q_1) = \min(\frac{1}{1}, \frac{3}{3}, \frac{q_1+1}{10}) , \tag{2.3.b}$$

où les 3 termes du min correspondent respectivement aux circuits $(1), (2)$ et (12).

A posteriori, l'introduction du quotient par $\varphi_{\mathcal{C}}$ (cf. (2.2.c)) est justifiée par le résultat suivant:

2.3.7 Proposition *Soit $\mathcal{R}$ une congruence sur $\mathcal{P}((\mathbb{R}^+)^{p+2})$. Les deux assertions suivantes sont équivalentes:*

(i) $a\mathcal{R}b \;\Rightarrow\; \sigma(a) = \sigma(b)$

(ii) $a\mathcal{R}b \;\Rightarrow\; a + \mathcal{C} = b + \mathcal{C}$

C'est dire que la congruence associée au cône $\mathcal{C}$ est la plus grossière des congruences définie sur le sous dioïde des séries *causales*[1], et compatibles avec la pente σ. En ce sens, le dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^{p+2}]] = \mathcal{P}(\mathbb{R}^{p+2})/\varphi_{\mathcal{C}}$ est "optimal".

Le sens (ii)⇒(i) de 2.3.7 est clair. Le sens (i)⇒(ii) est une conséquence du Lemme suivant:

2.3.8 Lemme *Soient x et y deux parties finies de $(\mathbb{R}^+)^{p+2}$. Si pour toute partie finie u de $(\mathbb{R}^+)^{p+2}$, $\sigma(x \otimes u) \preceq \sigma(y \otimes u)$ alors $x \subset y + \mathcal{C}$.*

En effet, si $a\mathcal{R}b$, on a pour tout u, $au\mathcal{R}bu$, et donc 2.3.7,(i) entraîne $\sigma(au) = \sigma(bu)$, d'où par 2.3.8 $a \subset b + \mathcal{C}$, et on rajoutant $\mathcal{C}$, $a + \mathcal{C} \subset b + \mathcal{C} + \mathcal{C} = b + \mathcal{C}$. Par symétrie, on obtient 2.3.7,(ii).

Preuve de 2.3.8. Nous le montrons dans le cas de $(\mathbb{R}^+)^2$. Le lecteur généralisera aux dimensions supérieures. Par additivité de σ, on peut supposer x réduit à un point, $x = \{(n,t)\}$. Soit $y = \{(n_i,t_i)\}_{i\in I}$. On choisit $u = \{(N,T)\}$. La condition $\forall u \subset (\mathbb{R}^+)^{p+2}$, $\sigma(x \otimes u) \preceq \sigma(y \otimes u)$ se réécrit:

$$\forall N,T \geq 0, \quad \frac{n+N}{t+T} \geq \inf_i \frac{n_i+N}{t_i+T} \quad ,$$

d'où

$$\forall N,T \geq 0, \;\; \exists i, \;\; \frac{n+N}{t+T} \geq \frac{n_i+N}{t_i+T} \quad ,$$

i.e.

$$\forall N,T \geq 0, \;\; \exists i, \;\; nt_i - n_i t \geq N(t_i - t) + T(n - n_i) \;\; .$$

Comme N et T peuvent être arbitrairement grand, cela n'est possible que si $\exists i, \;\; t_i - t \leq 0$ et $n - n_i \leq 0$, i.e. $(n,t) \in (n_i,t_i) + \mathcal{C} \subset b + \mathcal{C}$. ∎

3 Enumération des contraintes

Nous donnons maintenant des algorithmes algébriques pour énumérer les circuits de matrices à coefficients dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$.

3.1 Chemins élémentaires

Dans la suite, on raisonnera dans le dioïde complet commutatif libre engendré par les lettres a_{ij}, i.e. $\mathbb{B}[[a_{ij}]]$. Rappelons quelques définitions: on considère le graphe à n sommets et n^2 arcs (ij), l'arc (ji) étant repéré par la lettre a_{ij}. On a défini en 0,4.2.2 un chemin de longueur k de j à i comme un mot de la forme $p_{ij} = a_{ii_2} \ldots a_{i_{k-1}j}$. Un circuit est un chemin tel que $i = j$. Nous aurons besoin en outre des deux notions suivantes:

[1] i.e. à support dans $(\mathbb{R}^+)^{p+2}$

1/ notion classique de chemin et circuit élémentaire: un chemin $a_{i_1 i_2} \dots a_{i_{k-1} i_k}$ est élémentaire si les i_l sont distincts. Un circuit $a_{i i_2} \dots a_{i_{k-1} i}$ est élémentaire si $i, \dots, i_{k-1}$ sont distincts.
2/ chemin sans circuit: nous dirons que p_{ij} est *sans circuits* s'il ne se factorise pas sous la forme $p_{ij} = p'_{ij} c$, où p'_{ij} est un chemin de j à i et c un circuit non trivial. Le seul chemin sans circuits de i à i est donc le mot vide e. Un chemin sans circuits p_{ij} est élémentaire. La réciproque est vraie si $i \neq j$.

Ces arguties viennent du fait que la définition standard de circuit élémentaire rappelée ci-dessus n'est pas compatible avec celle de chemin élémentaire: un circuit élémentaire passant par i n'est pas un chemin élémentaire de i à i (car $i_1 = i_k$)! D'où l'utilité de la notion 2.

Nous considérons à nouveau la matrice générique d'ordre n, $\mathfrak{A}_n = (a_{ij})$ (cf. 0,§4.2).

3.1.1 Définition (Matrices des chemins élémentaires) *On note $\mathfrak{A}_n^{\mathsf{el}}$ (resp. $\mathfrak{A}_n^{\mathsf{sc}}$) la matrice dont le coefficient ij est égal à la somme des chemins élémentaires de j à i (resp. des chemins sans circuits). On notera $\pi(\mathfrak{A}_n)$ la somme des circuits élémentaires, i.e.*

$$\pi(\mathfrak{A}_n) = \mathrm{tr}(\mathfrak{A}_n^{\mathsf{el}}) \ .$$

On a donc $(\mathfrak{A}_n^{\mathsf{el}})_{ij} = (\mathfrak{A}_n^{\mathsf{sc}})_{ij}$ si $i \neq j$ et $(\mathfrak{A}_n^{\mathsf{sc}})_{ii} = e$. On étendra ces notations à un dioïde quelconque par morphisme: A_{ij}^{el} désigne la somme des *poids* des chemins élémentaires de j à i.

Les propriétés suivantes sont immédiates:

3.1.2 Propriétés *On a*

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

(ii) *Si D est diagonale,* $\pi(D \oplus A') = \mathrm{tr}(D) \oplus \pi(A')$

(iii) $\pi(A^k) \preceq \pi(A)$

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

3.1.3 Exemple On a $\mathfrak{A}_1^{\mathsf{el}} = (a_{11})$, $\mathfrak{A}_1^{\mathsf{sc}} = (e)$,

$$\mathfrak{A}_2^{\mathsf{el}} = \begin{bmatrix} a_{11} \oplus a_{12}a_{21} & a_{12} \\ a_{21} & a_{22} \oplus a_{21}a_{12} \end{bmatrix}, \ \mathfrak{A}_2^{\mathsf{sc}} = \begin{bmatrix} e & a_{12} \\ a_{21} & e \end{bmatrix}, \ \pi(\mathfrak{A}_2) = a_{11} \oplus a_{22} \oplus a_{12}a_{21} \ ,$$

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

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

Dans la pratique, on n'a pas de moyen peu coûteux d'obtenir $\mathfrak{A}^{\mathsf{el}}$ ou $\pi(\mathfrak{A})$. Nous introduisons une notion plus générale de matrice "approchant" $\mathfrak{A}^{\mathsf{el}}$.

3.1.4 Définition (Matrices énumérantes) *On dit que la matrice $\mathfrak{E}$ énumère les chemins si*

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

On dit que la matrice $\mathfrak{E}$ énumère les circuits si

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

On dit que le polynôme $\mathfrak{e}$ énumère les circuits si $\pi(\mathfrak{A}) \preceq \mathfrak{e} \preceq \pi(\mathfrak{A})^+$.

3.1.5 Proposition *Soit un système autonome de matrice $A \in (\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i, \delta])^{n\times n}$. On a, pour tout polynôme $\mathfrak{e}$ énumérant les circuits:*

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

où $\psi_A(\mathfrak{e})$ désigne le polynôme obtenu en donnant à la lettre a_{ij} la valeur A_{ij} dans l'expression de $\mathfrak{e}$.

Preuve On a par monotonie $\psi_A(\pi(\mathfrak{A})) \preceq \psi_A(\mathfrak{e}) \preceq \psi_A(\pi(\mathfrak{A})^*)$, d'où en utilisant la propriété de morphisme de ψ_A, la monotonie de la pente, et le fait que $\sigma(b^*) = \sigma(b)$: $\lambda = \sigma(\pi(A)) \preceq \sigma(\psi_A(\mathfrak{e})) \preceq \sigma((\pi(A))^*) = \lambda$. ∎

Le calcul du taux de production de ramène de la sorte au calcul de polynômes énumérants. Par exemple, les matrices

$$\mathrm{Id} \oplus \mathfrak{A} \oplus \ldots \oplus \mathfrak{A}^n, \quad (\mathrm{Id} \oplus \mathfrak{A}_n)^{2^k} \text{ avec } 2^k \geq n$$

énumèrent trivialement chemins et circuits. Les traces de ces matrices énumèrent donc les circuits. On a l'algorithme suivant, adapté de Gondran et Minoux [47],Chapitre 3, Algorithme 3':

3.1.6 Algorithme *Définissons la suite de matrices $A^{(i)}$ par*

$$A^{(0)} = \mathfrak{A}_n$$
$$A_{ij}^{(k)} = \begin{cases} A_{ij}^{(k-1)} \oplus A_{ik}^{(k-1)} A_{kj}^{(k-1)} & \text{si } i \neq k \text{ et } j \neq k \\ A_{ij}^{(k-1)} & \text{si } i = k \text{ ou } j = k. \end{cases}$$

La matrice $A^{(n)}$ est énumérante.

Preuve Introduisons les règles de simplification suivantes:

$$wc \to \varepsilon \tag{3.1.a}$$

où w est un mot non vide quelconque et c est un circuit quelconque. Il est clair qu'un chemin est irréductible ssi il ne se réduit pas modulo ces règles, et qu'en réduisant un polynôme quelconque, on ne perd pas de chemins irréductibles. On a montré en 0,4.2.1 que la récurrence $A_{ij}^{(k)} = A_{ij}^{(k-1)} \oplus A_{ik}^{(k-1)}(A_{kk}^{(k-1)})^* A_{kj}^{(k-1)}$ énumérait tous les circuits (y compris non élémentaires). En appliquant les règles de simplifications ci-dessus, on a si $i, j \neq k$ $A_{ik}^{(k-1)}(A_{kk}^{(k-1)})^* A_{kj}^{(k-1)} = A_{ik}^{(k-1)} A_{kj}^{(k-1)}$. Si par exemple $i = k$, on a, $A_{kj}^{(k-1)} \oplus A_{kk}^{(k-1)}(A_{kk}^{(k-1)})^* A_{kj}^{(k-1)} = A_{kj}^{(k-1)} \oplus A_{kj}^{(k-1)} = A_{kj}^{(k-1)}$. Cela montre 3.1.6. ∎

Si l'on ne vise que le calcul de $\pi(\mathfrak{A})$, on a un peu mieux. Précisément, on adapte l'élimination par blocs aux matrices énumérantes:

3.1.7 Proposition *Soit une partition $\{1,\dots,n\} = I \cup J$. Pour toute matrice $\mathfrak{E}_{[I|I]}$ énumérant les chemins de $\mathfrak{A}_{[I|I]}$, le polynôme*

$$c = \pi(\mathfrak{A}_{[I|I]}) \oplus \pi(\mathfrak{A}_{[J|J]} \oplus \mathfrak{A}_{[J|I]}\ \mathfrak{E}_{[I|I]}\ \mathfrak{A}_{[I|J]})$$

énumère les circuits.

Preuve L'élément (kl) de $\mathfrak{A}_{[J|J]} \oplus \mathfrak{A}_{[J|I]}\mathfrak{E}_{[I|I]}\mathfrak{A}_{[I|J]}$ est un chemin de l à k. On a ainsi $\mathfrak{A}_{[J|J]} \oplus \mathfrak{A}_{[J|I]}\mathfrak{E}_{[I|I]}\mathfrak{A}_{[I|J]} \preceq (\mathfrak{A}^+)_{[J|J]}$, et donc $c \preceq \mathrm{tr}(\mathfrak{A}^+)$. D'autre part, un circuit élémentaire c_0 est compté dans $\pi(A_{[I|I]})$ s'il ne passe par aucun sommet de J. Sinon, on peut factoriser c_0 sous comme produit de chemins élémentaires de la forme p_{uv} où $u, v \in J$ ou $p_{jk}p_{kl}p_{lr}$ avec $j, r \in J,\ k, l \in I$. Ce produit est compté dans $\pi(\mathfrak{A}_{[J|J]} \oplus \mathfrak{A}_{[J|I]}\mathfrak{E}_{[I|I]}\mathfrak{A}_{[I|J]})$. D'où $\pi(\mathfrak{A}) \preceq c$. ∎

3.1.8 Remarque On reconnaît en 3.1.7 l'analogue de

$$\det A = \det A_{[I|I]}\ \det(A_{[J|J]} - A_{[J|I]}A_{[I|I]}^{-1}A_{[I|J]}) \ .$$

3.1.9 Algorithme *Définissons les matrices $A^{(i)}$ et les scalaires $\pi^{(i)}$ de la manière suivante:*

$$\begin{aligned} A^{(0)} &= \mathfrak{A} \quad \pi^{(0)} = \mathrm{tr}\mathfrak{A} \\ A_{ij}^{(k)} &= A_{ij}^{(k-1)} \oplus A_{ik}^{(k-1)}A_{kj}^{(k-1)} \qquad \text{pour } i \neq j \text{ et } i, j > k \\ A_{ii}^{(k)} &= A_{ik}^{(k-1)}A_{ki}^{(k-1)} \\ \pi^{(k)} &= \pi^{(k-1)} \oplus \mathrm{tr}A_{[k+1\dots n|k+1\dots n]}^{(k)} \end{aligned}$$

Le polynôme $\pi^{(n-1)}$ énumère les circuits.

Si l'on effectue les calculs dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i, \delta]$, on aura donc à l'issue de l'algorithme

$$\lambda = \sigma(\pi^{(n-1)}) \ .$$

On observe que l'algorithme ne modifie que des sous matrices emboîtées de taille décroissante $n-1, \dots 2, 1$. On a ici l'analogue du calcul du déterminant en triangularisant la matrice par l'algorithme de Gauss.

Preuve Résulte d'une part 3.1.2,(ii) en prenant $D = \mathrm{diag}(A)$ et $A'_{ij} = A_{ij}$ en dehors de la diagonale et ε sinon, et d'autre part de l'application répétée l'élimination par bloc donnée en 3.1.7 avec le pivot d'ordre 1 $\mathfrak{E} = e$. ∎

3.1.10 Exemple Pour la matrice A de l'exemple 2.3.6, on obtient

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

$$\pi_1 = \pi_0 \oplus a_{22}^{(1)} = \gamma\delta \oplus \gamma^3\delta^3 \oplus \gamma\delta^{10}\gamma_1 \ , \tag{3.1.b}$$

ce qui donne le même résultat que (2.3.b).

3.1.11 Lien avec le rayon spectral en représentation compteur On a vu en VII,6.6 que le taux de production du système autonome s'écrivant $x(n+1) = A''x(n)$ est donné par $\rho(A'')$. En outre, en identifiant $\delta^t \simeq t$, on passe de la matrice A'' représentant le système dans $\mathbb{R}_{\min}$ à une matrice A représentant le système dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$ en posant $A = \gamma A'$. Comme $\rho(A)$ est le plus grand "coin" du polynôme caractéristique de A'' (cf. Cuninghame Greene, [25]). on s'attend à trouver un lien entre le calcul d'un polynôme énumérant les circuits de A et le calcul du polynôme caractéristique de A''. De manière précise, on a:

3.1.12 Théorème *Le polynôme* $\mathrm{perm}\,(\mathrm{Id} \oplus A)$ *énumère les circuits.*

Comme d'après ce qui précède, $\mathrm{perm}\,(\mathrm{Id} \oplus A) = \mathrm{perm}\,(\mathrm{Id} \oplus \gamma A'')$, l'analogie avec le polynôme caractéristique usuel est complète.

Preuve de 3.1.12. On a pour une permutation σ fixant l'ensemble J et décomposée en produit de cycles non triviaux, soit $\sigma = c_1 \dots c_k$:

$$\bigotimes_i (\mathrm{Id} \oplus A)_{i\sigma(i)} = w_A(c_1) \otimes \dots \otimes w_A(c_k) \otimes \bigotimes_{j \in J} (e \oplus A_{jj}) \ , \tag{3.1.c}$$

et donc $\mathrm{perm}\,(\mathrm{Id} \oplus A) \preceq (\pi(A))^*$. D'autre part, si σ admet le seul cycle non trivial c, on a, d'après (3.1.c), $\bigotimes_i A_{i\sigma(i)} \succeq w_A(c)$, d'où il résulte en sommant sur tous les cycles élémentaires c que $\mathrm{perm}\,(\mathrm{Id} \oplus A) \succeq \pi(A)$. ■

Nous calculons en passant le nombre de termes de $\pi(\mathfrak{A}_n)$, qui servira à estimer la complexité de la méthode.

3.1.13 Dénombrement des circuits élémentaires Le nombre $\mathsf{nc}(n,p)$ de circuits élémentaires commutatifs de longueur p associés à la matrice $\mathfrak{A}_n$ vaut:

$$\mathsf{nc}(n,p) = \frac{n(n-1)\dots(n-p+1)}{p}$$

(un circuit élémentaire de longueur p est donné par une partie à p éléments de $\{1,\dots,n\}$ ainsi qu'un ordre sur $p-1$ de ces éléments, soit $\mathsf{nc}(n,p) = \mathsf{C}_n^p \times (p-1)!$). Le nombre $\mathsf{nc}(n)$ de circuits élémentaires vaut donc:

$$\begin{aligned}\mathsf{nc}(n) = \sum_{p=1}^{n} \mathsf{nc}(n,p) &= n + \frac{n(n-1)}{2} + \frac{n(n-1)(n-2)}{3} + \dots \\ &+ \frac{n(n-1)\times\dots\times 2}{n-1} + \frac{n(n-1)\times\dots\times 2\times 1}{n} \\ &= (\dots + \frac{n}{(n-2)2!} + \frac{n}{(n-1)1!} + \frac{1}{0!})(n-1)!\end{aligned}$$

soit

$$\mathsf{nc}(n) \simeq e.(n-1)! \tag{3.1.d}$$

($e = 2.7182818\dots$). On les premières valeurs suivantes:

n	1	2	3	4	5	6	7	8	9	10
$\mathsf{nc}(n)$	1	3	8	24	89	415	2372	16072	125673	1112083

3.1.14 Application En appliquant l'algorithme 3.1.9 à la matrice A représentant l'atelier flexible décrit en 2.3.1, on trouve le polynôme suivant énumérant les circuits:

$$c=\delta^{11}\gamma\oplus\delta^{12}\gamma_1\oplus\delta^{19}\gamma_2\oplus\delta^{29}\gamma\gamma_1\oplus\delta^{30}\gamma\gamma_2\oplus\delta^{33}\gamma^2\gamma_2\oplus\delta^{40}\gamma^2\gamma_1\oplus\delta^{42}\gamma\gamma_1\gamma_2\oplus\delta^{53}\gamma^2\gamma_1\gamma_2\oplus\delta^{62}\gamma^3\gamma_1\gamma_2 \ ,$$

soit une somme de 10 termes. On obtient donc le taux de production comme min des 10 formes affines suivantes:
$\lambda(q)=\min(1/11,\frac{q_1}{12},\frac{q_2}{19},1/29+\frac{q_1}{29},1/30+\frac{q_2}{30},\frac{2}{33}+\frac{q_2}{33},1/20+\frac{q_1}{40},1/42+\frac{q_1}{42}+\frac{q_2}{42},\frac{2}{53}+\frac{q_1}{53}+\frac{q_2}{53},\frac{3}{62}+\frac{q_1}{62}+\frac{q_2}{62})$

On observe que l'écriture ci-dessus n'est pas minimale, i.e. que l'on peut retirer certaines formes affines sans changer la valeur de $\lambda(q)$. Par exemple, considérons les deux sous expressions:

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

On vérifie aisément que pour toute valeur positive ou nulle de q_1, on a $v\geq u$. On aurait ainsi pu retirer v de l'expression de λ.

On a une difficulté analogue en calculant le permanent comme en 3.1.12. On obtient alors le polynôme:
$\mathrm{perm}\,(\mathrm{Id}\oplus A)=e\oplus\delta^4\gamma_3\oplus\delta^{12}\gamma_1\oplus\delta^{12}\gamma\oplus\delta^{16}\gamma_1\gamma_3\oplus\delta^{19}\gamma_2\oplus\delta^{23}\gamma_2\gamma_3\oplus\delta^{24}\gamma^2\oplus\delta^{30}\gamma_1\gamma\oplus\delta^{30}\gamma\gamma_3\oplus$ $\delta^{31}\gamma_1\gamma_2\oplus\delta^{31}\gamma\gamma_2\oplus\delta^{35}\gamma_1\gamma_2\gamma_3\oplus\delta^{35}\gamma\gamma_2\gamma_3\oplus\delta^{35}\gamma_1\gamma_2\gamma\oplus\delta^{35}\gamma\gamma_1\gamma_3\oplus\delta^{35}\gamma^3\oplus\delta^{35}\gamma^2\gamma_2\oplus\delta^{35}\gamma^2\gamma_3\oplus\delta^{35}\gamma^2\gamma_1$

Ici encore, certains monômes sont redondants. Le fait que les algorithmes d'énumération 3.1.9 et 3.1.12 produisent des termes redondants conduisent à un accroissement “combinatoire” de la taille des données, ce qui rend ces algorithmes ineffectifs pour des systèmes de taille plus importante. Il est donc naturel de chercher les règles permettant de simplifier les calculs intermédiaires. Nous montrons dans la section suivante que l'introduction de certaines congruences permet d'obtenir l'expression minimale de $\lambda(q)$.

3.2 Congruences polyédriques en dimension 2

Nous commençons par donner les règles de simplification valables “en dimension 2”, i.e. dans le dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$, lorsqu'il s'agit de calculer le taux de production d'un système sans ressources inconnues. On verra que ces règles se généralisent aisément au cas d'un nombre quelconque de ressources.

On montre sur un exemple comment le fait d'avoir énuméré certains circuits permet de simplifier les calculs. On donnera ensuite les règles de simplifications générales dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$.

3.2.1 Exemple On considère une matrice A à coefficients dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$. On suppose qu'à une certaine étape k de l'algorithme de Gauss 3.1.9, on a un certain coefficient de $A^{(k)}$, $A_{ij}^{(k)}=e\oplus\gamma^2\delta^2$ et que par ailleurs, on a énuméré le circuit $c=\gamma\delta^3$. Le polynôme C donnant les circuits énumérés à l'issue de l'algorithme pourra s'écrire

$$C=c\oplus uA_{ij}^{(k)}\oplus v \tag{3.2.a}$$

où u et v sont des polynômes. On a

$$\sigma(C)=\min(\sigma(c),\sigma(uA_{ij}^{(k)}),\sigma(v)) \ . \tag{3.2.b}$$

Supposons, pour fixer les idées, que u est un monôme, soit $u=\gamma^n\delta^t$. On a alors

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

De deux choses l'une: soit $\sigma(\gamma^n\delta^t) < \sigma(c)$, auquel cas $\sigma(\gamma^n\delta^t\gamma^2\delta^2) > \sigma(\gamma^n\delta^t)$ (cf. VII,5.4.3), soit $\sigma(\gamma^n\delta^t) \geq \sigma(c)$, auquel cas le terme $\sigma(uA_{ij}^{(k)})$ ne modifie pas le min à droite de 3.2.b. Dans les deux cas, on aurait pu remplacer $A_{ij}^{(k)} = e \oplus \gamma^2\delta^2$ par $A_{ij}^{(k)} = e$ sans modifier la valeur finale de C.

Nous généralisons maintenant cette remarque.

Soit $c = \{c_1, \ldots, c_k\}$ un ensemble de k vecteurs de $(\mathbb{R}^+)^2$. On pose

$$\mathcal{C}_c = \mathbb{R}^+c_1 + \ldots + \mathbb{R}^+c_k \ .$$

3.2.2 Définition *On appelle* congruence polyédrique *modulo c la relation $\mathcal{R}_c$ définie comme suit dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^2]]$:*

$$x\mathcal{R}_c y \ \Leftrightarrow \ x + \mathcal{C}_c = y + \mathcal{C}_c \ .$$

On étend la définition de $\mathcal{R}_c$ à $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$ par isomorphisme. En posant

$$\varphi_c(x) = x + \mathcal{C}_c \ , \tag{3.2.c}$$

on observe que φ_c vérifie les propriétés B,1.2.8. La relation $\mathcal{R}_c$ est donc une *bonne congruence*. Ici encore, on peut donner une interprétation géométrique simple des classes d'équivalence modulo $\mathcal{C}_c$.

3.2.3 Lemme *Soient x, y deux ensembles de points de $(\mathbb{R}^+)^2$. Les assertions suivantes sont équivalentes:*

(i) $\overline{x}\mathcal{R}_c\overline{y}$ *dans* $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^2]]$

(ii) $x + \mathcal{C} + \mathcal{C}_c = y + \mathcal{C} + \mathcal{C}_c$ *dans* $\mathcal{P}(\mathbb{R}^2)$.

$\mathcal{C}$ est ici le cône servant à définir $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$, donné en 2.2.2.

Preuve Résulte de ce que $x + \mathcal{C}$ est un représentant de $\overline{x}$ et $y + \mathcal{C}$ est un représentant de $\overline{y}$ dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^2]]$. ∎

On introduisant $\Psi_c(x) = x + \mathcal{C} + \mathcal{C}_c$, on a ainsi le diagramme commutatif suivant, où les flèches sont des projections canoniques:

$$\begin{tikzcd}
\mathcal{P}(\mathbb{R}^2) \arrow[r] \arrow[d] & \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^2]] \arrow[d] \\
\mathcal{P}(\mathbb{R}^2)/\Psi_c \arrow[r, phantom, "\simeq"] & \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\mathbb{R}^2]]/\varphi_c
\end{tikzcd}$$

On en déduit alors l'analogue du Lemme 2.2.3.

3.2.4 Lemme *L'application $\imath_c$ définie par*

$$\begin{cases} \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]/\mathcal{R}_c & \longrightarrow \quad \mathcal{P}(\mathbb{Z}^2) + \mathcal{C} + \mathcal{C}_c \\ \quad s & \longmapsto \quad \imath(s) = \mathrm{supp}\, s + \mathcal{C} + \mathcal{C}_c \end{cases}$$

est un isomorphisme de dioïdes.

3.2.5 Exemple On a représenté sur la Figure IX.3 le polynôme $p = e \oplus \gamma\delta^2 \oplus \gamma^3\delta^3 \oplus \gamma^4\delta^4 \oplus \gamma^7\delta^5$ (cercles et carrés noirs), et son représentant minimal modulo $\mathcal{R}_{\gamma^2\delta}$, $p_0 = e \oplus \gamma\delta^2 \oplus \gamma^4\delta^4$ (cercles noirs exclusivement). La partie $\imath_c(p)$ est représentée en grisé. Le représentant minimal p_0 est formé des point extrémaux de cette partie.

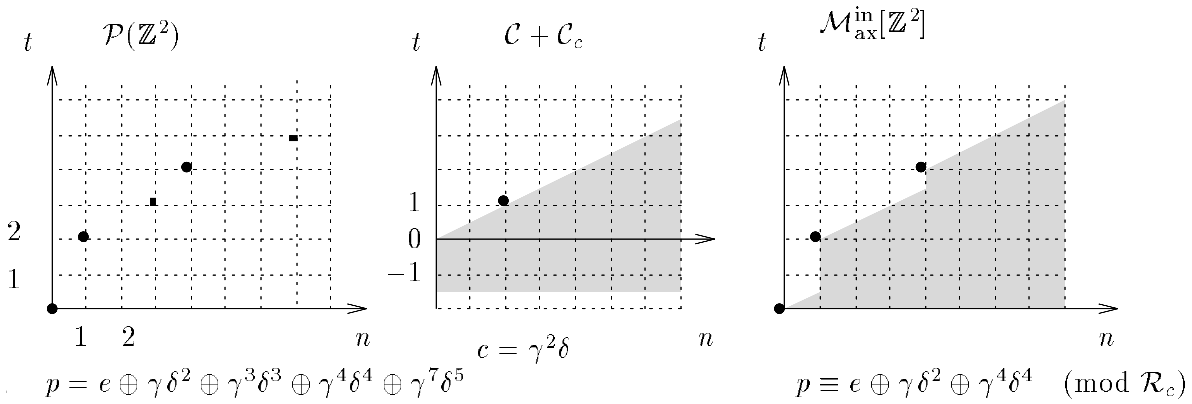

Figure IX.3: Congruence polyédrique en dimension 2

Via B,§1.2 le calcul effectif dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\mathbb{R}^2]$ se réduit à l'examen de la relation de redondance suivante pour des points de $\mathbb{R}^2$:

$$x \leq_{\Phi_c} y \Leftrightarrow x \in y + \mathcal{C} + \mathcal{C}_c \ . \tag{3.2.d}$$

3.2.6 Lemme *On a*

$$(t', n') \leq_{\Phi_c} (t, n) \quad \Leftrightarrow \quad n' \geq n \quad \text{et} \quad \frac{n' - n}{t' - t} \geq \sigma(c) = \min_j \frac{\nu_j}{\tau_j} \ . \tag{3.2.e}$$

Preuve de 3.2.6. On vérifie par un calcul immédiat que

$$\begin{aligned} \mathcal{C}_c + \mathcal{C} &= (\tau_1, \nu_1)\mathbb{R}^+ + \ldots + (\tau_r, \nu_r)\mathbb{R}^+ + (-1, 0)\mathbb{R}^+ + (0, 1)\mathbb{R}^+ \\ &= (1, \sigma(c)) + (-1, 0)\mathbb{R}^+ + (0, 1)\mathbb{R}^+ \ . \end{aligned} \tag{3.2.f}$$

Ainsi, via 3.2.3,(ii) la congruence associée à c coïncide avec la congruence associée au seul monôme $(\sigma(c), 1)$. Supposons $\sigma(c) > 0$ (sinon le résultat est trivial). On a

$$(0, 1) = \sigma(c)^{-1}(-1, 0) + \sigma(c)^{-1}(1, \sigma(c)) \ ,$$

et donc:

$$\mathcal{C}_c + \mathcal{C} = (1, \sigma(c))\mathbb{R}^+ + (-1, 0)\mathbb{R}^+ \ .$$

Ainsi, on a $(t', n') \leq_{\Phi_c} (t', n')$ ssi $\{(t, n')\} \subset \{(t, n)\} + \mathcal{C} + \mathcal{C}_c = \{(t, n)\} + (1, \sigma(c))\mathbb{R}^+ + (-1, 0)\mathbb{R}^+$. La condition (3.2.e) en résulte. ∎

On a comme conséquence immédiate de 3.2.6 une caractérisation des formes minimales modulo $\mathcal{R}_c$.

3.2.7 Corollaire *Le polynôme p écrit sous forme canonique dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma, \delta]$, $p = \gamma^{n_1}\delta^{t_1} \oplus \ldots \oplus \gamma^{n_k}\delta^{t_k}$ est minimal modulo $\mathcal{R}_c$ ssi*

$$\forall j > i, \quad \sigma(\gamma^{n_j}\delta^{t_j}\gamma^{-n_i}\delta^{-t_i}) < \sigma(c) \ .$$

Autrement dit, les pentes entre deux monômes quelconques de p doivent être inférieures à la pente de c.

On notera dans la suite $\oplus$ le min des pentes.

3.2.8 Lemme *Supposons $\pi(A) = \pi(A) \oplus c$. Pour toute matrice A' congruente*[2] *à A modulo $\mathcal{R}_c$, on a $\sigma(\pi(A')) \oplus \sigma(c) = \sigma(\pi(A))$.*

Le résultat central de ce chapitre en découle immédiatement:

3.2.9 Théorème *Chaque pas de l'algorithme de Gauss peut s'effectuer modulo la congruence polyédrique associée aux circuits énumérés auparavant.*

Par exemple, on pourra effectuer l'algorithme 3.1.9 dans la chaîne suivante de dioïdes:

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

Preuve du lemme 3.2.8. La remarque suivante immédiate est la clé du résultat:

3.2.10 Lemme *Pour tous points a et b, pour tous $s, r \geq 0$,*

$$\sigma(a^s b^r) \preceq \sigma(a) \oplus \sigma(b) \ .$$

Avec les notations usuelles, c'est dire que $\sigma(s.a + r.b) \geq \min(\sigma(a), \sigma(b))$.

Preuve Analogue à VII,5.4.3, compte tenu de $\sigma(a^s) = \sigma(a)$ et $\sigma(b^t) = \sigma(b)$. ■

Nous allons montrer que (i): $A' \preceq \varphi_c(A) \Rightarrow \sigma(w(A')) \oplus \sigma(c) \preceq \sigma(w(A))$ et (ii): $A \preceq \varphi_c(A') \Rightarrow \sigma(w(A)) \preceq \sigma(w(A')) \oplus \sigma(c)$. La conclusion en résultera. Avec les notations du dioïde, on a si $c = \{c_1, \ldots, c_k\}$:

$$\varphi_c(x) = x + \mathcal{C} = x \otimes c = x c_1^{\mathbb{R}^+} \ldots c_k^{\mathbb{R}^+} \ . \tag{3.2.g}$$

(i): Supposons $A' \preceq \varphi_c(A)$. Soit $\alpha = (i_1, \ldots, i_m)$ un circuit (on convient que $i_{m+1} = i_1$). On a donc $A'_{i_v i_{v+1}} \preceq \varphi_c(A_{i_v i_{v+1}})$ d'où en prenant le produit pour $v = 1 \ldots m$ et en utilisant la propriété de morphisme de φ_c: $w_{A'}(\alpha) \preceq \varphi_c(w_A(\alpha))$. Nous supposons que $w_{A'}(\alpha)$ est un singleton, le cas général s'en déduisant par additivité. On a donc $w_{A'}(\alpha) \preceq w_A(\alpha)\, c_1^{s_1} \ldots c_\ell^{s_\ell}$ pour un certain $s \in (\mathbb{R}^+)^k$. D'où

$$\sigma(w_{A'}(\alpha)) \preceq \sigma(w_A(\alpha)\, c_1^{s_1} \ldots c_\ell^{s_\ell}) \ .$$

En rajoutant $\sigma(c)$ de part et d'autre et en utilisant 3.2.10,

$$\sigma(w_{A'}(\alpha)) \oplus \sigma(c) \preceq \sigma(w_A(\alpha)) \oplus \sigma(c) \ .$$

En sommant sur tous les circuits α, on obtient une première inégalité:

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

(ii): Supposons au rebours $A \preceq \varphi_c(A')$. On écrit au terme d'un argument analogue

$$\sigma(w_A(\alpha)) \preceq \sigma(w_{A'}(\alpha)) \oplus \sigma(c)$$

qui en sommant donne immédiatement $\sigma(\pi(A)) \preceq \sigma(\pi(A')) \oplus \sigma(c)$. Cela montre l'autre inégalité. Si $A\,\mathcal{R}_c\,A'$, on a $A \preceq \varphi_c(A')$ et $A' \preceq \varphi_c(A)$. Cela achève la preuve de 3.2.8. ■

[2] coefficient par coefficient

3.2.11 Exemple Soit à calculer le taux de production associé à la matrice

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

On énumère trivialement les circuits diagonaux, soit

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

Modulo la congruence associée à c, on a $\gamma \oplus \gamma^4\delta \equiv \gamma$ (car $\gamma^4\delta \preceq \gamma(\gamma\delta)$), et de même $\gamma\delta^{10} \oplus \gamma^2\delta^{11} \equiv \gamma\delta^{10}$. Il résulte que le calcul du taux de production associé à A est le même pour la matrice suivante:

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

Ainsi, le taux de production est:

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

3.2.12 Analogie avec les fonctions support Une idée naturelle de simplification découle de l'observation suivante.
(i) D'ordinaire, on associe à une partie X de $\mathbb{R}^n$ sa fonction support :

$$\delta_X^\star : \quad \delta_X^\star(p) = \sup_{x\in X} \langle p, x\rangle .$$

Il est bien connu que $\delta_X^\star$ ne dépend que de l'enveloppe convexe de X.
(ii) Ici, on a l'application:

$$\sigma : \quad [\sigma(P)](q) = \inf_{(n,t)\in P} \frac{\langle q, n\rangle}{t} ,$$

qu'on peut voir comme une variante homogène de la fonction support. On peut montrer que $\sigma(P)$ ne dépend que du cône engendré par P (i.e. de l'ensemble des combinaisons linéaires positives ou nulles de vecteurs de P). Au lieu de la congruence ci-dessus, on pourrait penser introduire la règle suivante: $P\ \mathcal{R}'\ P'$ ssi P et P' engendrent le même cône. Il faut bien voir que cette règle n'est pas compatible avec le calcul du taux de production. Par exemple, $\gamma\delta$ et $\gamma^2\delta^2$ définissent le même cône de $\mathbb{R}^2$. Cependant, les deux matrices

$$\begin{bmatrix} \varepsilon & \gamma\delta \\ \delta & \varepsilon \end{bmatrix}, \quad \begin{bmatrix} \varepsilon & \gamma^2\delta^2 \\ \delta & \varepsilon \end{bmatrix}$$

ont des taux de production différents.

On peut montrer plus précisément que la congruence introduite ci-dessus est optimale.

3.2.13 Proposition *Soit $\mathcal{R}$ une congruence sur $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$. Les deux assertions suivantes sont équivalentes:*

(i) $\forall a, b, u, c \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$, $a\mathcal{R}b \Rightarrow \sigma(au) \oplus \sigma(c) = \sigma(bu) \oplus \sigma(c)$,

(ii) $a\mathcal{R}b \Rightarrow a + \mathcal{C} + \mathcal{C}_c = b + \mathcal{C} + \mathcal{C}_c$.

De manière analogue à la Proposition 2.3.7, il suffit de montrer le Lemme suivant.

3.2.14 Lemme *Si pour tout u, $\sigma(au) \preceq \sigma(bu) \oplus \sigma(c)$, alors $a \subset b + \mathcal{C} + \mathcal{C}_c$.*

Preuve du Lemme. On suppose $a = \gamma^n \delta^t$ et $b = \bigoplus_i \gamma^{n_i} \delta^{t_i}$. On pose $u = \gamma^N \delta^T$, avec

$$T = \frac{1}{\sigma(c)} N + T' \ .$$

Pour T' assez grand, il est clair que $\sigma(bu) \succ \sigma(c)$. L'inégalité $\sigma(au) \preceq \sigma(bu) \oplus \sigma(c)$ se réécrit alors:

$$\frac{n+N}{t+T} \geq \inf_i \frac{n_i + N}{t_i + T} \ .$$

Il existe donc un indice i (dépendant a priori de N et T) tel que

$$nt_i - tn_i \geq N(t - t_i) + T(n_i - n) = N(t - t_i + \frac{n_i - n}{\sigma(c)}) + T'(n_i - n) \ .$$

Comme N et T' peuvent être arbitrairement grands, cela n'est possible que s'il existe i (indépendant de N et T) tel que

$$n_i - n \leq 0, \quad t - t_i + \frac{n_i - n}{\sigma(c)} \leq 0 \ ,$$

ce qui montre le Lemme. ■

3.3 Congruences polyédriques en dimension quelconque

Soit $c \in \mathcal{M}_{\rm ax}^{\rm in}[\mathbb{R}^{p+2}]$, causal. De manière analogue au cas précédent, on introduit le cône

$$\mathcal{C}_c = \mathbb{R}^+ c_1 + \ldots + \mathbb{R}^+ c_k \ ,$$

où $\{c_1, \ldots, c_k\}$ est un ensemble de k vecteurs de $\mathbb{R}^{p+2}$ représentant c. On définit la congruence polyédrique $\mathcal{R}_c$ associée à c de manière analogue à 3.2.2.

3.3.1 Exemple Soit $p = \gamma_0 \delta \oplus \gamma_0 \gamma_1 \delta^2 \in \mathcal{M}_{\rm ax}^{\rm in}[\gamma_0, \gamma_1, \delta]$, et $c = \gamma_1^2 \delta^6$. On a $p \, \mathcal{R}_c \, \gamma_0 \delta$ car $\gamma_0 \gamma_1 \delta^2 \preceq \gamma_0 \delta (c)^{\frac{1}{2}}$, ou dans $\mathbb{R}^3$,

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

Cependant, pour la congruence modulo c^* (cf. B,2.7.1) on n'a pas $p \equiv \gamma_0 \delta$.

Nous noterons seulement ceci.

3.3.2 Observation Le Théorème 3.2.9 s'étend à $\mathcal{M}_{\rm ax}^{\rm in}[\mathbb{R}^{p+2}]$.

3.3.3 Exemple Pour la matrice A de 2.3.6,3.1.10 on peut en particulier simplifier π_1 modulo la congruence associée à π_0. Comme $\gamma^3 \delta^3 \preceq \gamma \delta (\gamma \delta)^2$, ou dans $\mathbb{R}^2$, $(3,3) \in (1,1)\mathbb{R}^+$, le monôme $\gamma^3 \delta^3$ est $\leq_{\Phi_c}$-dominé par $\gamma \delta$ (cf. 3.2.d), et l'on peut écrire

$$\lambda = \sigma(\gamma \delta \oplus \gamma_1 \gamma \delta^{10}) = \min(1, \frac{1 + q_1}{10}) \ ,$$

ce qui était évident à partir de (2.3.b).

3.4 Exemple

On considère l'atelier flexible de type "flowshop" représenté sur la Figure IX.1 et décrit par la matrice suivante:

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

Nous appliquons l'algorithme d'énumération 3.1.9. On a en écrivant à chaque étape de l'algorithme que la partie utile de la matrice $a^{(k)}$:

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

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

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

Nous détaillons le calcul de $a^{(2)}_{35}$. On a $a^{(2)}_{35} = a^{(1)}_{35} \oplus a^{(1)}_{32} a^{(1)}_{25} = \gamma_1\delta^{10} \oplus \gamma\delta^{11}\gamma_1\delta^{10} \equiv \gamma_1\delta^{10} \pmod{\mathcal{R}_{\pi^2}}$, car $\gamma\delta^{11}\gamma_1\delta^{10} \preceq \gamma_1\delta^{10}\pi^2$. On continue de même:

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

(on a mis les termes inessentiels entre parenthèses).

$$a^{(4)} = \begin{bmatrix} \gamma_1\delta^{12}(\oplus\gamma\gamma_1\delta^{21}) & \gamma\delta^{10} \oplus \gamma\gamma_2\delta^{23}(\oplus\gamma\gamma_2\delta^{21}) \\ \delta \oplus \gamma_1\delta^{19} & \gamma_2\delta^{19} \end{bmatrix} \qquad \pi^{(4)} = \gamma\delta^{11} \oplus \gamma_1\delta^{12} \oplus \gamma^2\delta^{19}$$

$a^{(5)} = (\delta \oplus \gamma_1\delta^{19})(\gamma\delta^{10} \oplus \gamma\gamma_2\delta^{23}) = \gamma\delta^{11} \oplus \gamma\gamma_1\delta^{22} \oplus (\gamma\gamma_2\delta^{24}) \oplus \gamma\gamma_2\delta^{42}$, $\pi^{(5)} = \gamma\delta^{11} \oplus \gamma_1\delta^{12} \oplus \gamma_2\delta^{19} \oplus \gamma\gamma_1\delta^{29}$. Soit l'ensemble de contraintes

$$\lambda = \sigma(\pi^{(5)}) = \min(\frac{1}{11}, \frac{q_1}{12}, \frac{1+q_1}{29}, \frac{q_2}{19}) \ . \tag{3.4.a}$$

Nous avons à l'aide de `MAX` calculé les traces des matrices a^* pour les valeurs suivantes de q_1 et q_2:

(q_1, q_2)	$\mathrm{tr}(a^*)$	$\sigma(\mathrm{tr}(a^*))$
$(1,1)$	$(\gamma\delta^{19})^*$	$\frac{1}{19}$
$(1,2)$	$(e \oplus \gamma\delta^{12})\,(\gamma^2\delta^{29})^*$	$\frac{2}{29}$
$(2,1)$	$(\gamma\delta^{19})^*$	$\frac{1}{19}$
$(2,2)$	$(\gamma\delta^{11})^*$	$\frac{1}{11}$

Le taux de production $\sigma(\mathrm{tr}(a^*))$ coïncide bien avec $\lambda(q)$ donné par la formule (3.4.a).

3.5 Autre exemple

Pour l'atelier étudié en VIII,§4, on trouve l'expression suivante:

$$\text{circuits} = \delta^4\gamma_3 \oplus \delta^{12}\gamma \oplus \delta^{12}\gamma_1 \oplus \delta^{19}\gamma_2 \oplus \delta^{30}\gamma\gamma_3 \oplus \delta^{30}\gamma\gamma_1$$

$$\lambda(q) = \min(\frac{q_3}{4}, 1/12, \frac{q_1}{12}, \frac{q_2}{19}, 1/30 + \frac{q_3}{30}, 1/30 + \frac{q_1}{30})$$

Les calculs sont laissés au lecteur. On pourra aussi utiliser la macro `enum` écrite dans le package MAX, décrite plus bas.

3.6 Calcul modulo une congruence polyédrique

Nous étudions maintenant le calcul effectif modulo $\mathcal{R}_c$ en dimension quelconque. Ici encore, via 3.2.3 et le fait que $\mathcal{R}_c$ est une bonne congruence, il suffit de savoir déterminer si, étant donné deux points a et b de $\mathbb{R}^{p+2}$, on a

$$a \in b + \mathcal{C} + \mathcal{C}_c \ . \tag{3.6.a}$$

Autrement dit, il suffit de savoir déterminer si le point a appartient au cône polyédrique convexe $b+\mathcal{C}+\mathcal{C}_c$, ce qui peut se faire en projetant a. Précisément, soit U la matrice formée en concaténant la matrice J des $p+2$ vecteurs de base du cône $\mathcal{C}$, i.e.

$$J = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & & 0 & 0 \\ \vdots & 1 & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{bmatrix}$$

et les ℓ vecteurs de base $c_1, \dots . c_\ell$ du cône $\mathcal{C}_c = c_1\mathbb{R}^+ + \dots c_\ell\mathbb{R}^+$ (i.e. $U = [J, c_1, \dots, c_\ell]$). Etant donné deux points a et b de $\mathbb{R}^{p+2}$, on a $a \in b + \mathcal{C}_c$ si et seulement si:

$$\exists x \in (\mathbb{R}^+)^{p+2+\ell}, \quad \|a - b - Ux\|^2 = 0,$$

où $\|y\|$ désigne la norme euclidienne de y. On a employé pour vérifier cela les méthodes de programmation quadratique avec gestion de contraintes actives implémentées dans BASILE. On peut aussi, et c'est plus simple, déterminer à l'aide d'algorithmes de programmation linéaire si l'ensemble de contraintes suivant admet un point $y \in (\mathbb{R}^+)^l$ admissible:

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

3.7 Complexité en dimension 2

Nous passons maintenant à l'étude de la complexité des algorithmes gaussiens décrits plus haut dans le cas systèmes "constants" ($p = 0$). La matrice représentant le système aura alors ses coefficients dans $\mathcal{M}_{\rm ax}^{\rm in}[\gamma, \delta]$.

Le point important est que grâce à 3.2.6, la complexité des simplifications modulo φ_c ne dépend que de la pente de c et pas du nombre de termes de c. Nous appellerons *comparaison élémentaire* un test de redondance modulo φ_c entre deux monômes dans $\mathcal{M}_{\rm ax}^{\rm in}[\gamma, \delta]$.

3.7.1 Corollaire *Soient p, q, c des polynômes causaux dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$. Le calcul de $p \oplus q$ modulo $\mathcal{R}_c$ s'effectue en $O((\max(\deg_\gamma p, \deg_\gamma q))^2)$ comparaisons élémentaires. Le calcul de pq s'effectue en $O((\deg_\gamma p + \deg_\gamma q)^2)$ comparaisons élémentaires.*

Preuve Le polynôme $p \oplus q$ dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$ a au plus $n = \max(\deg_\gamma p, \deg_\gamma q)$ termes. En utilisant l'algorithme B,1.2.22, sa mise sous forme minimale requiert $O(n^2)$ comparaisons élémentaires. Argument analogue pour pq. ∎

3.7.2 Corollaire *Soit $A \in (\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta])^{n\times n}$. Notons $\deg_\gamma A = \max_{ij} \deg_\gamma A_{ij}$. Les algorithmes 3.1.6 et 3.1.9, effectués modulo 3.2.9, coûtent $O(n^5(\deg_\gamma A)^2)$ comparaisons élémentaires.*

Preuve D'après l'interprétation combinatoire de l'algorithme de Gauss donnée en 0,4.2.3, le coefficient ij de $A_{ij}^{(k)}$ comme une somme de termes de la forme $p_{ij}c_1 \ldots c_l$, où p_{ij} est un chemin élémentaire de j à i passant éventuellement par $1,..,k$, et $c_1, \ldots, c_\ell$ sont des circuits contenus dans $\{1, \ldots, k\}$. Si l'on simplifie $A^{(k)}$ modulo la congruence polyédrique associée aux circuits contenus dans $\{1, \ldots, k\}$ (et donc précédemment énumérés), on ne retiendra qu'un sous ensemble des chemins élémentaires. Ainsi, on aura $\deg_\gamma A_{ij}^{(k)} \leq (k+1)\deg_\gamma A \leq n \deg_\gamma A$. D'après 3.7.1 le coût d'un produit (et a fortiori d'une somme) lors de l'élimination de Gauss sera $O((n \deg_\gamma A)^2)$. Comme il y a $O(n^3)$ de telles opérations, on obtient la borne 3.7.2. ∎

3.7.3 Remarque On a en effectuant 3.1.9, modulo 3.2.9 un nouvel algorithme pour calculer le taux de production d'un système sans ressources inconnues. De manière générale, soit un graphe donc chaque arc (ij) est valué par deux quantités positives ou nulles N_{ij} et T_{ij}. On a de la sorte un algorithme pour calculer la moyenne minimale suivante:

$$\lambda = \min_{k\geq 1} \min_{i_1,\ldots,i_k} \frac{N_{i_1 i_2} + \ldots + N_{i_k i_1}}{T_{i_1 i_2} + \ldots + T_{i_k i_1}} \quad .$$

Cet algorithme serait à comparer aux méthodes usuelles de détermination de ratio minimum (cf. Gondran et Minoux, Annexe V de [48]),

Nous donnerons un résultat de complexité en dimension quelconque après avoir décrit un procédé d'agrégation.

4 Quelques raffinements

4.1 Agrégation

Soit une partition $\{1, \ldots, n\} = I \cup J$ de l'ensemble des sommets du graphe. On décompose la matrice associés a un système autonome sous la forme:

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

On a via 3.1.7

$$\pi(A) = \pi(A_{II}) \oplus \pi\left(A_{JJ} \oplus A_{JI}A'_{II}A_{IJ}\right) \tag{4.1.a}$$

pour toute matrice A'_{II} énumérant les chemins de A_{II}. On peut voir cela comme un résultat d'agrégation. Si la matrice A_{II} est "constante" (i.e. ne contient que les indéterminées γ et δ), la matrice énumérante A'_{II} sera facile à déterminer. En particulier, les simplifications modulo les congruences polyédriques s'effectueront aisément. La complexité du calcul sera donc reliée à la taille de la matrice résiduelle $A_{JJ} \oplus A_{JI}A'_{II}A_{IJ}$, qui correspond au nombre de ressources. Autrement dit, on pourra traiter de la sorte des graphes d'événements d'assez grande taille, pourvu que le nombre de ressources soit petit.

Nous illustrons cette approche par un exemple spectaculaire où l'on obtient une formule fermée donnant le taux de production d'un atelier à 2 pièces et n machines identiques, en fonction non seulement des ressources, mais également des temps de fabrication.

4.2 Taux de production d'un flowshop à 2 types de pièces et n étages identiques

On considère l'atelier de type flowshop analogue à celui traité en §3.4, avec ici 2 types de pièces et n machines identiques (ou prenant des temps identiques pour traiter les pièces), représenté à gauche de la Figure IX.4. On suppose donc que la machine i traite pendant un temps a la pièce 1 et ensuite pendant un temps b la pièce 2, a et b ne dépendant pas de i. Initialement, q_1 palettes de type 1 et q_2 palettes de type 2 se trouvent en amont de la machine 1.

4.2.1 Théorème *Le taux de production du Flowshop à n machines identiques et 2 types de pièces, donné sur la Figure IX.4, est égal à:*

$$\lambda(q_1, q_2) = \min\left(\frac{1}{a+b}, \frac{q_1}{na}, \frac{q_2}{nb}, \frac{1+\min(q_1,q_2)}{2a+2b+(n-2)\max(a,b)}\right) . \tag{4.2.a}$$

Preuve On peut écrire avec les notations de la Figure IX.4:

$$\begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} = \begin{bmatrix} \varepsilon & \gamma\delta^b \\ \delta^a & \varepsilon \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} \oplus \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \oplus \begin{bmatrix} \gamma_1\delta^a & \varepsilon \\ \varepsilon & \gamma_2\delta^b \end{bmatrix} \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} , \tag{4.2.b}$$

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

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

Plutôt que d'appliquer formellement la réduction (4.1.a), nous préférons éliminer à la main les variables x_{1i}, x_{2i} associées aux étages $i = 2 \ldots n$, ce qui est en fait équivalent. Il suffit donc de calculer le transfert H de u à y du système non rebouclé à droite de la Figure IX.4. Le transfert H' du système bouclé s'obtiendra en résolvant

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

soit

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

Pour simplifier les calculs, on notera les durées comme des scalaires, i.e. $\gamma^k\delta^a \simeq a\gamma^k$. On a alors par un calcul d'étoile immédiat le transfert suivant entre l'étage $i-1$ et l'étage i:

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

On a donc énuméré un premier circuit

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

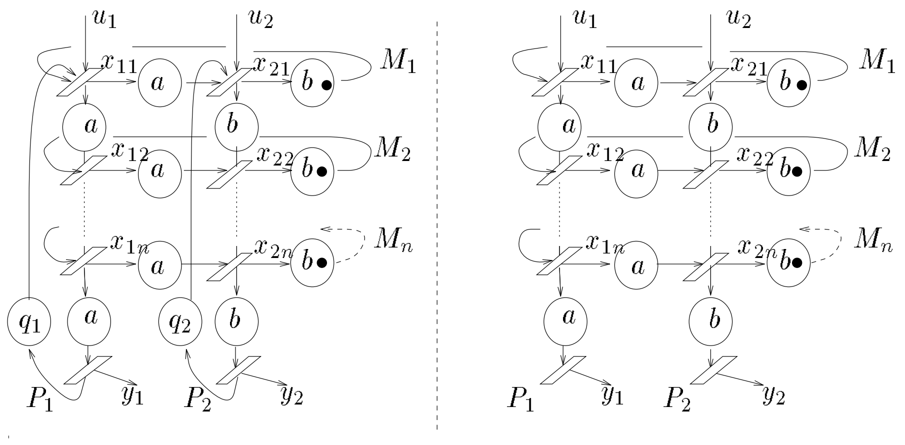

Figure IX.4: Atelier flexible à n étages identiques

Modulo la congruence associée à π_1, on a $(ab\gamma)^* = e$, et l'on peut écrire

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

Ainsi,

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

On calcule aisément

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

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

Modulo la congruence polyédrique associée à π_1, on a

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

car $a^3b^3\gamma \preceq ab(a \oplus b)(ab\gamma)$. Ainsi:

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

et via une récurrence immédiate:

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

Le taux de production est donc égal au min des poids des circuits de la matrice $D\mathcal{H}^n$ ainsi que du circuit π_1 déjà énuméré, soit

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

On a donc $\lambda = \sigma(\pi_1) \oplus \sigma(\pi)$ où π est donné par 3.1.3:

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

Supposons par exemple $a \succeq b$. On a $\gamma_1 a^n \otimes a^2 b^2 (a \oplus b)^{n-2} \gamma\gamma_2 \succeq a^2 b^2 (a \oplus b)^{2n-2}$ d'où il résulte que le dernier terme de π est inessentiel. De même si $a \preceq b$. Ainsi,

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

et donc en revenant à l'algèbre habituelle, on a la formule (4.2.a). ■

4.3 Complexité

Pour un grand nombre de ressources indéterminées, la simplification modulo les congruences polyédriques sera ineffective. L'algorithme 3.2.9 sera pertinent lorsque le nombre p de ressources indéterminées sera petit, et lorsque par ailleurs le graphe d'événements temporisés sera grand. La section précédente représente un cas limite, où l'on a pu traiter une graphe d'événements de taille arbitrairement grande avec 2 ressources. Plus généralement, nous considérons un système sous la forme standard donnée sur la Figure IX.5. $\mathrm{diag}(\gamma_i)$ est une matrice diagonale de taille p dont les coef-

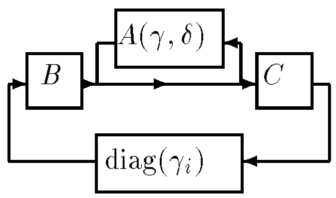

Figure IX.5: Forme standard d'un système à p ressources indéterminées

ficients diagonaux sont les γ_i, $i = 1, \ldots, p$. A est une matrice de taille $n \times n$ à coefficients dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma, \delta]$, qui représente la partie “constante” du système. B et C sont des matrices à coefficients booléiens. Nous laissons le lecteur se convaincre que l'on peut toujours se ramener à une telle forme, au besoin en rajoutant des transitions en amont et en aval des places où les p ressources inconnues sont situées. Via (4.1.a), on se ramène à énumérer les circuits de $A'' = D \oplus CA'B$, où A' est une matrice énumérant les chemins de A. D'après 3.7.2, l'obtention de A', et donc le calcul de $D \oplus CA'B$ requiert de l'ordre de $O(n^5 (\deg_\gamma A)^2)$ opérations. Le nombre de termes de chaque coefficient de A'' est alors borné par $n^2 \deg_\gamma A$. Il reste à énumérer les circuits de A''. Nous supposons ici que cette énumération s'effectue sans utiliser les congruences. Comme A'' résulte d'une agrégation, A'' sera en général pleine, on a donc d'après 3.1.13 $O((p-1)!)$ circuits élémentaires, chacun somme au plus de $(n^2 \deg_\gamma A)^p$ monômes, soit un nombre final de termes $O((p-1)!(n^2 \deg_\gamma A)^p)$. L'intérêt de cette borne, sans doute assez grossière, est de montrer que pour un nombre de ressources p fixé, la complexité est pseudo-polynomiale en la taille du système n (i.e. dépend polynomialement de n, mais également de la taille des données représentée par $\deg_\gamma A$).

5 Application

5.1 Implémentation courante

La manipulation des polynômes et matrices dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$ a été provisoirement développée en MAPLE pour être compatible avec MAX. Par contre, la simplification modulo les congruences polyédriques exige de savoir projeter sur un cône polyédrique. Cela relève du calcul numérique, et les programmes standard sont évidemment écrits en FORTRAN. Notre propos n'était pas ici d'écrire le code général et performant pour l'optimisation des ressources, mais de tester les algorithmes décrits plus haut sur des exemples raisonnables. Nous nous sommes résolus pour cela au compromis (provisoire) suivant: 1/ on simplifie en MAPLE modulo certaines règles "approchant" la congruence polyédrique modulo c (voir ci-dessous). 2/ on effectue l'énumération des chemins en MAPLE (élimination de Gauss ou énumération exhaustive selon la taille du système). 3/ on transmet de manière interactive les données à BASILE, par exemple à l'aide de l'interface BASILE-MAPLE. On effectue ensuite les simplifications exactes. On récupère le résultat en MAPLE.
Cela sera illustré sur un exemple réel dans la section suivante.

Une implémentation définitive devrait inclure la projection sur des cônes polyédriques à coordonnées rationnelles, du moins si l'on prétend faire du calcul formel. Si l'on admet des erreurs d'arrondi, on pourrait écrire assez rapidement un programme BASILE gérant et simplifiant les matrices polynomiales, et utilisant les codes existants de programmation quadratique.

5.1.1 Approximation de la congruence polyédrique modulo c Soit $c=\bigoplus_i c_i$ un polynôme. La congruence polyédrique modulo c est plus grossière que chacune des congruences associée à chaque monôme c_i. Soit $c_i=\delta^t\gamma_0^{n_0}\dots\gamma_p^{n_p}$. Il est immédiat de déterminer si

$$\delta^{t'}\gamma_0^{n'_0}\dots\gamma_p^{n'_p}\subset\mathcal{C}_{\delta^t\gamma_0^{n_0}\dots\gamma_p^{n_p}}$$

Cela revient à vérifier si x tel que $t'=xt$ satisfait

$$(n'_0,\dots,n'_p)\geq x(n_0,\dots,n_p)\ .$$

Ainsi, la simplification modulo la congruence associée à un seul monôme c_i est immédiate. En pourra donc effectuer l'approximation suivante de $\mathcal{R}_c$, qui consiste à simplifier successivement modulo chacun des monômes c_i. Les représentants obtenus de la sorte ne seront évidemment plus minimaux.

Soient par exemple $c=\gamma_0\delta\oplus\gamma_1\delta^3=c_1\oplus c_2$, $a=e\oplus\gamma_2\delta^5$ et $a'=e\oplus\gamma_0\gamma_1^2\delta^6$. Modulo $\mathcal{R}_c$, on a $a\equiv e$ car $\gamma_1^2\delta^5\preceq e\otimes c_2^2$. Modulo $\mathcal{R}_{c_1}$ et $\mathcal{R}_{c_2}$, a' est minimal. Cependant, $\gamma_0\gamma_1^2\delta^6\preceq c_1c_2^2$ ce qui montre que $a'\equiv e\pmod{\mathcal{R}_c}$. Ainsi, l'approximation précédente ne simplifie pas a'.

5.1.2 Définition des objets de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$ en MAX Les polynômes de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$ sont définis comme suit:
`polynôme` = ⟨liste de monômes de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$⟩ | `eps`.

Le monôme $\gamma_0^{n_0}\gamma_1^{n_1}\dots\gamma_p^{n_p}\delta^t$ est représenté par la liste $[t,n_0,n_1,\dots,n_p]$. Les listes de monômes sont ordonnées par t croissants. La fonction

`p2l` ⟨polynôme usuel⟩,⟨dimension⟩ → ⟨ polynôme dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i,\delta]$⟩
convertit un polynôme usuel en les indéterminées $g_0,\dots,g_p$ en un polynôme dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_0,\dots,\gamma_p,\delta]$, où p est la dimension. L'indéterminée g est un synonyme de g_0. On a par exemple:

```
MAX>u:=g*g3*d^2+d+g3*d^2;
                                          2            2
                              u := g g3 d  + d + g3 d

MAX> w:=p2l(u,3);
                      w := [[1, 0, 0, 0, 0], [2, 0, 0, 0, 1]]
MAX> l2p(w);
                                              2
                                      d + g3 d
MAX> w1:=p2l(u,4);
                   w1 := [[1, 0, 0, 0, 0, 0], [2, 0, 0, 0, 1, 0]]
MAX> l2p(w1);
                                              2
                                      d + g3 d
MAX> u:=p2l(g*g1*g2+d*g1,2):l2p(u);
                                         d g1
```

Les polynômes ainsi convertis sont sous forme minimale, comme on l'observe sur le dernier exemple.

Les opérateurs &+et &*de `MAX` reconnaissent les polynômes de $\mathcal{M}_{\text{ax}}^{\text{in}}[\gamma_i, \delta]$. Par exemple:

```
MAX> u:=p2l(g*g1+d*g2,2);
                         u := [[0, 1, 1, 0], [1, 0, 0, 1]]
MAX> l2p(u);
                                     g g1 + d g2

MAX> v:=p2l(d*g2*g1+d^3*g1^2,2);
                         v := [[1, 0, 1, 1], [3, 0, 2, 0]]
MAX> u&+ v;
                    [[0, 1, 1, 0], [1, 0, 0, 1], [3, 0, 2, 0]]
MAX> l2p(");
                                                3   2
                                g g1 + d g2 + d   g1
MAX> l2p(u&* v);
                         2      2    3      3    4   2
                        d  g1 g2  + d  g g1  + d  g1  g2
```

5.1.3 La routine `enum` de MAX La fonction

`enum` : ⟨ matrice⟩ → ⟨ polynôme⟩
retourne un polynôme énumérant les circuits de la matrice, simplifié modulo l'approximation décrite plus haut. Par exemple

```
MAX> a:=p2l(array([[g0*d ,g1*d^3],[g^2, g^2*g1^2*d^2]]),2);

                           [ [[1, 1, 0, 0]]  [[3, 0, 1, 0]] ]
                      a := [                                ]
                           [ [[0, 2, 0, 0]]  [[2, 2, 2, 0]] ]
MAX> l2p(a);
```

$$\begin{bmatrix} \gamma\delta & \gamma_1\delta^3 \\ \gamma^2 & \gamma^2{\gamma_1}^2\delta^2 \end{bmatrix}$$

```
MAX> c:=enum(a);
                              c := [[1, 1, 0, 0], [3, 2, 1, 0]]
MAX> l2p(c);
```

$$\gamma\delta \oplus \delta^3\gamma^2\gamma_1$$

`enum` admet également la syntaxe `enum(<matrice>,<polynome>)`, auquel cas les circuits énumérés sont également simplifiés modulo le polynôme passé ne second paramètre. Par exemple, avec la matrice A ci-dessus, on a:

```
MAX> c:=enum(a,p2l(g1*d^2,2));
          c := [[1, 1, 0, 0], [2, 0, 1, 0]]
MAX> l2p(c);
```

$$\gamma\delta \oplus \gamma_1\delta^2$$

En effet, on a $\delta^3\gamma^2\gamma_1 \preceq (\gamma\delta)(\gamma_1\delta^2)$, et ce monôme redondant modulo la congruence associée à $\gamma_1\delta^2$ a bien été éliminé.

5.1.4 La routine `array2flex`

L'instruction `array2flex`(A) génère la matrice $a \in \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma_i, \delta]$ représentant un atelier flexible de type flowshop à n machines et p pièces, spécifié par une matrice A de taille $n \times p$. A_{ij} désigne le temps de passage (réel positif ou `eps`) de la pièce j sur la machine i. Si ce temps est égal à `eps`, on suppose que la pièce j ne passe pas sur la machine i et passe à la machine suivante. Les indéterminées $\gamma_1, \ldots, \gamma_p$ correspondent respectivement aux nombres de palettes de type $1, \ldots, p$. Par exemple, la matrice a de l'exemple 2.3.1 à été produite par

```
MAX> A:=array([[9,2],[2,7],[1,10]]):

MAX> a:=array2flex(A);
            a := [eps, [[2, 1, 0, 0]], eps, eps, [[1, 0, 1, 0]], eps]
                 [[[9, 0, 0, 0]], eps, eps, eps, eps, [[10, 0, 0, 1]]]
                 [[[9, 0, 0, 0]], eps, eps, [[7, 1, 0, 0]], eps, eps]
                 [eps, [[2, 0, 0, 0]], [[2, 0, 0, 0]], eps, eps, eps]
                 [eps, eps, [[2, 0, 0, 0]], eps, eps, [[10, 1, 0, 0]]]
                 [eps, eps, eps, [[7, 0, 0, 0]], [[1, 0, 0, 0]], eps]

MAX> l2p(a);
```

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

On obtient le polynôme $\pi^{(5)}$ énumérant les circuits de a donné en §3.4 de la manière suivante:

```
MAX> c:=enum(a);
```

$$c := [[11, 1, 0, 0], [12, 0, 1, 0], [19, 0, 0, 1], [29, 1, 1, 0]]$$

```
MAX> l2p(c);
```

$$\delta^{11}\gamma \oplus \delta^{12}\gamma_1 \oplus \delta^{19}\gamma_2 \oplus \delta^{29}\gamma\gamma_1 \ .$$

5.2 Un exemple réel

Nous traitons ici l'atelier flexible étudié par Cohen, Dubois, Quadrat et Viot en 83 [17]. On considère l'atelier flexible de type flowshop à 8 machines et 6 types de pièces, avec les temps de passage des pièces sur les machines décrits sur le tableau IX.1. On a pris les ordres de passage fixes et définis

temps de passage	pièce 1	pièce 2	pièce 3	pièce 4	pièce 5	pièce 6	charge totale
machine 1	2	3.9	0.95	1.1	0.7	1.4	10.05
machine 2	-	-	2	1.2	-	1.7	4.9
machine 3	3.7	-	2.2	-	6.4	-	12.3
machine 4	-	-	2	-	1	1	4
machine 5	1.7	3.1	3	-	1.3	-	9.1
machine 6	0.5	3.2	4.3	1.9	1.6	0.4	11.9
machine 7	1	1	1	1	1	1	6
machine 8	1.5	1.5	1.5	1.2	1.2	1.2	8.1
durée circuit pièce	10.4	12.7	16.95	6.4	13.2	6.7	

Table IX.1:

sur la table IX.2. La machine critique (de charge maximale 12.3) est la machine 3. On voit qu'une

machine 1	séquence	1 2 3 4 5 6
machine 2	séquence	3 4 6
machine 3	séquence	1 3 5
machine 4	séquence	3 5 6
machine 5	séquence	1 2 3 5
machine 6	séquence	1 2 3 4 5 6
machine 7	séquence	1 2 3 4 5 6
machine 8	séquence	1 2 3 4 5 6

Table IX.2:

seule palette par type de pièce est insuffisante pour saturer la machine 3. La durée du cycle de la pièce 3 est en effet de 16.95, et l'on a sait donc a priori que le taux de production avec 1 palette par type de pièce est inférieur à $(16.95)^{-1}$.

La matrice représentant le système est de taille 48×48 (on a $48 = 8 \times 6$ transitions). Une première approche consiste à appliquer l'algorithme d'énumération avec l'approximation 5.1.1. On obtient alors en utilisant la macro `enum` décrite ci-dessus au bout d'1h40 de MAPLE les 156 circuits donnés dans la Table IX.3

$$
\begin{array}{llllll}
c_1 = \delta^{6.4}\gamma_4 & \oplus & \delta^{6.7}\gamma_6 & \oplus & \delta^{10.4}\gamma_1 & \oplus \\
\delta^{12.3}\gamma & \oplus & \delta^{12.7}\gamma_2 & \oplus & \delta^{13.2}\gamma_5 & \oplus \\
\delta^{16.95}\gamma_3 & \oplus & \delta^{28.3}\gamma\gamma_4 & \oplus & \delta^{28.75}\gamma\gamma_1 & \oplus \\
\delta^{28.8}\gamma\gamma_5 & \oplus & \delta^{28.65}\gamma\gamma_6 & \oplus & \delta^{29.65}\gamma\gamma_2 & \oplus \\
\delta^{30.05}\gamma\gamma_3 & \oplus & \delta^{40.75}\gamma^2\gamma_4 & \oplus & \delta^{41.25}\gamma^2\gamma_5 & \oplus \\
\delta^{41.35}\gamma\gamma_4\gamma_6 & \oplus & \delta^{41.45}\gamma\gamma_1\gamma_4 & \oplus & \delta^{41.75}\gamma^2\gamma_6 & \oplus \\
\delta^{41.85}\gamma^2\gamma_1 & \oplus & \delta^{42.85}\gamma\gamma_2\gamma_6 & \oplus & \delta^{42.95}\gamma\gamma_1\gamma_2 & \oplus \\
\delta^{44.85}\gamma\gamma_2\gamma_4 & \oplus & \delta^{45.35}\gamma\gamma_2\gamma_5 & \oplus & \delta^{55.55}\gamma\gamma_2\gamma_4\gamma_6 & \oplus \\
\delta^{55.65}\gamma\gamma_1\gamma_2\gamma_4 & \oplus & \delta^{58.50}\gamma\gamma_1\gamma_3\gamma_5 & \oplus & \delta^{70.0}\gamma^2\gamma_1\gamma_5\gamma_6 & \oplus \\
\delta^{71.20}\gamma^3\gamma_2\gamma_3 & \oplus & \delta^{74.20}\gamma^2\gamma_3\gamma_5\gamma_6 & \oplus & \delta^{74.30}\gamma^2\gamma_1\gamma_3\gamma_5 & \oplus \\
\delta^{79.50}\gamma^4\gamma_3\gamma_4 & \oplus & \delta^{79.85}\gamma^2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{81.90}\gamma^4\gamma_2\gamma_4 & \oplus \\
\delta^{82.25}\gamma^2\gamma_2\gamma_4{}^2\gamma_5 & \oplus & \delta^{82.65}\gamma^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{82.70}\gamma^2\gamma_1\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{82.75}\gamma^3\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{82.75}\gamma^2\gamma_1\gamma_4{}^2\gamma_5 & \oplus & \delta^{82.80}\gamma^2\gamma_1{}^2\gamma_4\gamma_5 & \oplus \\
\delta^{84.10}\gamma^2\gamma_3{}^2\gamma_4\gamma_5 & \oplus & \delta^{84.20}\gamma^2\gamma_1\gamma_2\gamma_5\gamma_6 & \oplus & \delta^{84.30}\gamma^2\gamma_1{}^2\gamma_2\gamma_5 & \oplus \\
\delta^{85.55}\gamma^3\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{85.60}\gamma^3\gamma_1\gamma_5\gamma_6 & \oplus & \delta^{85.65}\gamma^3\gamma_1\gamma_4\gamma_5 & \oplus \\
\delta^{86.50}\gamma^2\gamma_2\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{86.65}\gamma^2\gamma_1\gamma_3\gamma_4\gamma_6 & \oplus & \delta^{86.75}\gamma^2\gamma_1{}^2\gamma_3\gamma_4 & \oplus \\
\delta^{86.90}\gamma^3\gamma_2\gamma_3\gamma_5 & \oplus & \delta^{86.90}\gamma^2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{87.0}\gamma^2\gamma_1\gamma_3\gamma_4\gamma_5 & \oplus \\
\delta^{87.30}\gamma^3\gamma_3\gamma_5\gamma_6 & \oplus & \delta^{87.40}\gamma^3\gamma_1\gamma_3\gamma_5 & \oplus & \delta^{98.35}\gamma^3\gamma_1\gamma_4{}^2\gamma_5 & \oplus \\
\delta^{88.40}\gamma^2\gamma_2\gamma_3\gamma_5\gamma_6 & \oplus & \delta^{88.50}\gamma^2\gamma_1\gamma_2\gamma_3\gamma_5 & \oplus & \delta^{90.60}\gamma^5\gamma_1\gamma_5 & \oplus \\
\delta^{95.40}\gamma^2\gamma_1\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{95.45}\gamma^3\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{95.85}\gamma^4\gamma_3\gamma_4\gamma_5 & \oplus \\
\delta^{96.90}\gamma^2\gamma_1\gamma_2\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{97.0}\gamma^2\gamma_1{}^2\gamma_2\gamma_4\gamma_5 & \oplus & \delta^{98.25}\gamma^4\gamma_2\gamma_4\gamma_5 & \oplus \\
\delta^{98.25}\gamma^3\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{98.65}\gamma^4\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{98.70}\gamma^4\gamma_1\gamma_5\gamma_6 & \oplus \\
\delta^{98.75}\gamma^4\gamma_1\gamma_4\gamma_5 & \oplus & \delta^{98.80}\gamma^4\gamma_1{}^2\gamma_5 & \oplus & \delta^{99.20}\gamma^3\gamma_2{}^2\gamma_4\gamma_5 & \oplus \\
\delta^{99.20}\gamma^2\gamma_2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{99.30}\gamma^2\gamma_1\gamma_2\gamma_4{}^2\gamma_5 & \oplus & \delta^{99.75}\gamma^3\gamma_2\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{99.75}\gamma^3\gamma_1\gamma_3\gamma_4\gamma_6 & \oplus & \delta^{99.80}\gamma^3\gamma_1\gamma_2\gamma_5\gamma_6 & \oplus & \delta^{99.85}\gamma^3\gamma_1\gamma_2\gamma_4\gamma_5 & \oplus \\
\delta^{99.85}\gamma^3\gamma_1{}^2\gamma_3\gamma_4 & \oplus & \delta^{100.0}\gamma^4\gamma_2\gamma_5\gamma_6 & \oplus & \delta^{100.10}\gamma^4\gamma_1\gamma_2\gamma_5 & \oplus \\
\delta^{100.40}\gamma^4\gamma_3\gamma_5\gamma_6 & \oplus & \delta^{100.50}\gamma^4\gamma_1\gamma_3\gamma_5 & \oplus & \delta^{100.85}\gamma^2\gamma_1\gamma_2\gamma_3\gamma_4\gamma_6 & \oplus \\
\delta^{100.95}\gamma^2\gamma_1{}^2\gamma_2\gamma_3\gamma_4 & \oplus & \delta^{101.10}\gamma^3\gamma_2{}^2\gamma_5\gamma_6 & \oplus & \delta^{101.10}\gamma^2\gamma_2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{101.20}\gamma^3\gamma_1\gamma_2{}^2\gamma_5 & \oplus & \delta^{110.05}\gamma^5\gamma_3{}^2\gamma_5 & \oplus & \delta^{111.10}\gamma^5\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{111.20}\gamma^5\gamma_1\gamma_4\gamma_5 & \oplus & \delta^{111.80}\gamma^5\gamma_1\gamma_5\gamma_6 & \oplus & \delta^{112.05}\gamma^5\gamma_2{}^2\gamma_5 & \oplus \\
\delta^{112.45}\gamma^5\gamma_2\gamma_5\gamma_6 & \oplus & \delta^{112.45}\gamma^3\gamma_2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{112.55}\gamma^5\gamma_1\gamma_2\gamma_5 & \oplus \\
\delta^{112.55}\gamma^3\gamma_1\gamma_2\gamma_4{}^2\gamma_5 & \oplus & \delta^{112.85}\gamma^5\gamma_3\gamma_5\gamma_6 & \oplus & \delta^{112.95}\gamma^5\gamma_1\gamma_3\gamma_5 & \oplus \\
\delta^{113.40}\gamma^2\gamma_2{}^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{113.50}\gamma^2\gamma_1\gamma_2{}^2\gamma_4{}^2\gamma_5 & \oplus & \delta^{113.55}\gamma^4\gamma_2{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{113.65}\gamma^4\gamma_1\gamma_2{}^2\gamma_5 & \oplus & \delta^{113.80}\gamma^3\gamma_2{}^2\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{113.90}\gamma^3\gamma_1\gamma_2{}^2\gamma_4\gamma_5 & \oplus \\
\delta^{123.55}\gamma^6\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{123.65}\gamma^6\gamma_1\gamma_4\gamma_5 & \oplus & \delta^{124.25}\gamma^6\gamma_1\gamma_5\gamma_6 & \oplus \\
\delta^{128.0}\gamma^3\gamma_1{}^2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{128.10}\gamma^3\gamma_1{}^3\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{131.05}\gamma^7\gamma_4{}^2\gamma_5 & \oplus \\
\delta^{132.20}\gamma^3\gamma_1\gamma_3{}^2\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{132.30}\gamma^3\gamma_1{}^2\gamma_3{}^2\gamma_4\gamma_5 & \oplus & \delta^{140.70}\gamma^3\gamma_1{}^2\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{142.20}\gamma^3\gamma_1{}^2\gamma_2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{142.30}\gamma^3\gamma_1{}^3\gamma_2\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{143.55}\gamma^4\gamma_1\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{143.65}\gamma^4\gamma_1{}^2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{144.50}\gamma^3\gamma_1\gamma_2\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{144.60}\gamma^3\gamma_1{}^2\gamma_2\gamma_3\gamma_4{}^2\gamma_5 & \oplus \\
\delta^{145.30}\gamma^4\gamma_1\gamma_3{}^2\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{145.40}\gamma^4\gamma_1{}^2\gamma_3{}^2\gamma_4\gamma_5 & \oplus & \delta^{146.40}\gamma^3\gamma_1\gamma_2\gamma_3{}^2\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{146.50}\gamma^3\gamma_1{}^2\gamma_2\gamma_3{}^2\gamma_4\gamma_5 & \oplus & \delta^{156.65}\gamma^5\gamma_1\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{156.70}\gamma^5\gamma_1{}^2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{156.75}\gamma^5\gamma_1{}^2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{156.80}\gamma^5\gamma_1{}^3\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{157.75}\gamma^4\gamma_1\gamma_2\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{157.85}\gamma^4\gamma_1{}^2\gamma_2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{158.0}\gamma^5\gamma_1\gamma_2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{158.10}\gamma^5\gamma_1{}^2\gamma_2\gamma_3\gamma_4\gamma_5 & \oplus \\
\delta^{158.40}\gamma^5\gamma_1\gamma_3{}^2\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{158.50}\gamma^5\gamma_1{}^2\gamma_3{}^2\gamma_4\gamma_5 & \oplus & \delta^{158.70}\gamma^3\gamma_1\gamma_2{}^2\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{158.80}\gamma^3\gamma_1{}^2\gamma_2{}^2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{159.10}\gamma^4\gamma_1\gamma_2{}^2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{159.20}\gamma^4\gamma_1{}^2\gamma_2{}^2\gamma_3\gamma_4\gamma_5 & \oplus \\
\delta^{169.10}\gamma^6\gamma_1\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{169.20}\gamma^6\gamma_1{}^2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{170.45}\gamma^6\gamma_1\gamma_2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{170.55}\gamma^6\gamma_1{}^2\gamma_2\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{170.85}\gamma^6\gamma_1\gamma_3{}^2\gamma_4\gamma_5\gamma_6 & \oplus & \delta^{170.90}\gamma^5\gamma_1{}^2\gamma_2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{171.65}\gamma^5\gamma_1{}^2\gamma_2{}^2\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{170.95}\gamma^6\gamma_1{}^2\gamma_3{}^2\gamma_4\gamma_5 & \oplus & \delta^{171.55}\gamma^5\gamma_1\gamma_2{}^2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{181.55}\gamma^7\gamma_1\gamma_3\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{181.65}\gamma^7\gamma_1{}^2\gamma_3\gamma_4{}^2\gamma_5 & \oplus & \delta^{182.25}\gamma^7\gamma_1{}^2\gamma_3\gamma_4\gamma_5\gamma_6 & \oplus \\
\delta^{182.35}\gamma^7\gamma_1{}^3\gamma_3\gamma_4\gamma_5 & \oplus & \delta^{186.0}\gamma^4\gamma_1{}^3\gamma_3{}^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{189.80}\gamma^4\gamma_1{}^2\gamma_2\gamma_3{}^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{189.90}\gamma^4\gamma_1{}^3\gamma_2\gamma_3{}^2\gamma_4{}^2\gamma_5 & \oplus & \delta^{199.10}\gamma^5\gamma_1{}^3\gamma_3{}^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{202.90}\gamma^5\gamma_1{}^2\gamma_2\gamma_3{}^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus \\
\delta^{203.0}\gamma^5\gamma_1{}^3\gamma_2\gamma_3{}^2\gamma_4{}^2\gamma_5 & \oplus & \delta^{204.0}\gamma^4\gamma_1{}^2\gamma_2{}^2\gamma_3{}^2\gamma_4{}^2\gamma_5\gamma_6 & \oplus & \delta^{204.10}\gamma^4\gamma_1{}^3\gamma_2{}^2\gamma_3{}^2\gamma_4{}^2\gamma_5 &
\end{array}
$$

Table IX.3: Circuits avant simplification

On passe ensuite ces 156 circuits en BASILE, et l'on obtient après simplification exacte un ensemble minimal de 25 circuits.

ensemble simplifié de circuits=$c = \delta^{6.4}\gamma_4 \oplus \delta^{6.7}\gamma_6 \oplus \delta^{10.4}\gamma_1 \oplus \delta^{12.3}\gamma \oplus \delta^{12.7}\gamma_2 \oplus \delta^{13.2}\gamma_5 \oplus \delta^{16.95}\gamma_3 \oplus \delta^{28.3}\gamma\gamma_4 \oplus \delta^{28.65}\gamma\gamma_6 \oplus \delta^{28.75}\gamma\gamma_1 \oplus \delta^{28.8}\gamma\gamma_5 \oplus \delta^{29.65}\gamma\gamma_2 \oplus \delta^{30.05}\gamma\gamma_3 \oplus \delta^{40.75}\gamma^2\gamma_4 \oplus \delta^{41.25}\gamma^2\gamma_5 \oplus \delta^{41.35}\gamma\gamma_4\gamma_6 \oplus \delta^{41.45}\gamma\gamma_1\gamma_4 \oplus \delta^{41.75}\gamma^2\gamma_6 \oplus \delta^{41.85}\gamma^2\gamma_1 \oplus \delta^{42.85}\gamma\gamma_2\gamma_6 \oplus \delta^{42.95}\gamma\gamma_1\gamma_2 \oplus \delta^{44.85}\gamma\gamma_2\gamma_4 \oplus \delta^{45.35}\gamma\gamma_2\gamma_5 \oplus \delta^{55.55}\gamma\gamma_2\gamma_4\gamma_6 \oplus \delta^{55.65}\gamma\gamma_1\gamma_2\gamma_4$

Une autre approche consiste à appliquer l'algorithme de Gauss seulement jusqu'à l'itération 42, et à énumérer brutalement (i.e. sans simplifications) les 415 circuits de la matrice résiduelle $A^{(42)}$ de taille 6×6. Les coefficients de $A^{(42)}$ sont les suivants:

$$
\begin{aligned}
A_{11}^{(42)} &= \delta^{28.35}\gamma\gamma_1 \oplus \delta^{41.45}\gamma^2\gamma_1\\
A_{12}^{(42)} &= \delta^{26.35}\gamma\gamma_2 \oplus \delta^{39.45}\gamma^2\gamma_2\\
A_{13}^{(42)} &= \delta^{22.45}\gamma\gamma_3 \oplus \delta^{35.55}\gamma^2\gamma_3\\
A_{14}^{(42)} &= \delta^{16.3}\gamma\gamma_4 \oplus \delta^{31.9}\gamma^2\gamma_4 \oplus \delta^{44.35}\gamma^3\gamma_4\\
A_{15}^{(42)} &= \delta^{15.2}\gamma\gamma_5 \oplus \delta^{30.8}\gamma^2\gamma_5 \oplus \delta^{43.25}\gamma^3\gamma_5\\
A_{16}^{(42)} &= \delta^{1.2}\gamma \oplus \delta^{11.5}\gamma\gamma_6 \oplus \delta^{29.45}\gamma^2\gamma_6 \oplus \delta^{42.55}\gamma^3\gamma_6
\end{aligned}
$$

$$
\begin{aligned}
A_{21}^{(42)} &= \delta^{1.5} \oplus \delta^{16.2}\gamma_1 \oplus \delta^{31.65}\gamma\gamma_1\\
A_{22}^{(42)} &= \delta^{27.35}\gamma\gamma_2 \oplus \delta^{40.45}\gamma^2\gamma_2\\
A_{23}^{(42)} &= \delta^{25.75}\gamma\gamma_3\\
A_{24}^{(42)} &= \delta^{22.1}\gamma\gamma_4 \oplus \delta^{34.55}\gamma^2\gamma_4\\
A_{25}^{(42)} &= \delta^{21.0}\gamma\gamma_5 \oplus \delta^{33.45}\gamma^2\gamma_5\\
A_{26}^{(42)} &= \delta^{17.3}\gamma\gamma_6 \oplus \delta^{32.75}\gamma^2\gamma_6
\end{aligned}
$$

$$
\begin{aligned}
A_{31}^{(42)} &= \delta^{22.85}\gamma_1 \oplus \delta^{35.95}\gamma\gamma_1\\
A_{32}^{(42)} &= \delta^{1.5} \oplus \delta^{20.85}\gamma_2 \oplus \delta^{33.95}\gamma\gamma_2\\
A_{33}^{(42)} &= \delta^{24.45}\gamma\gamma_3 \oplus \delta^{37.55}\gamma^2\gamma_3\\
A_{34}^{(42)} &= \delta^{26.4}\gamma\gamma_4 \oplus \delta^{38.85}\gamma^2\gamma_4\\
A_{35}^{(42)} &= \delta^{25.3}\gamma\gamma_5 \oplus \delta^{37.75}\gamma^2\gamma_5\\
A_{36}^{(42)} &= \delta^{23.95}\gamma\gamma_6 \oplus \delta^{37.05}\gamma^2\gamma_6
\end{aligned}
$$

$$
\begin{aligned}
A_{41}^{(42)} &= \delta^{24.75}\gamma_1 \oplus \delta^{37.85}\gamma\gamma_1\\
A_{42}^{(42)} &= \delta^{22.75}\gamma_2 \oplus \delta^{35.85}\gamma\gamma_2\\
A_{43}^{(42)} &= \delta^{1.5} \oplus \delta^{18.85}\gamma_3 \oplus \delta^{31.95}\gamma\gamma_3\\
A_{44}^{(42)} &= \delta^{19.3}\gamma\gamma_4 \oplus \delta^{34.9}\gamma^2\gamma_4 \oplus \delta^{47.35}\gamma^3\gamma_4\\
A_{45}^{(42)} &= \delta^{27.2}\gamma\gamma_5 \oplus \delta^{39.65}\gamma^2\gamma_5\\
A_{46}^{(42)} &= \delta^{25.85}\gamma\gamma_6 \oplus \delta^{38.95}\gamma^2\gamma_6
\end{aligned}
$$

$$
\begin{aligned}
A_{51}^{(42)} &= \delta^{26.35}\gamma_1 \oplus \delta^{39.45}\gamma\gamma_1\\
A_{52}^{(42)} &= \delta^{24.35}\gamma_2 \oplus \delta^{37.45}\gamma\gamma_2\\
A_{53}^{(42)} &= \delta^{20.45}\gamma_3 \oplus \delta^{33.55}\gamma\gamma_3\\
A_{54}^{(42)} &= \delta^{1.2} \oplus \delta^{14.3}\gamma_4 \oplus \delta^{29.9}\gamma\gamma_4 \oplus \delta^{42.35}\gamma^2\gamma_4\\
A_{55}^{(42)} &= \delta^{19.2}\gamma\gamma_5 \oplus \delta^{34.8}\gamma^2\gamma_5 \oplus \delta^{47.25}\gamma^3\gamma_5\\
A_{56}^{(42)} &= \delta^{27.45}\gamma\gamma_6 \oplus \delta^{40.55}\gamma^2\gamma_6
\end{aligned}
$$

$$\begin{aligned}
A_{61}^{(42)} &= \delta^{27.35}\gamma_1 \oplus \delta^{40.45}\gamma\gamma_1 \\
A_{62}^{(42)} &= \delta^{25.35}\gamma_2 \oplus \delta^{38.45}\gamma\gamma_2 \\
A_{63}^{(42)} &= \delta^{21.45}\gamma_3 \oplus \delta^{34.55}\gamma\gamma_3 \\
A_{64}^{(42)} &= \delta^{15.3}\gamma_4 \oplus \delta^{30.9}\gamma\gamma_4 \oplus \delta^{43.35}\gamma^2\gamma_4 \\
A_{65}^{(42)} &= \delta^{1.2} \oplus \delta^{14.2}\gamma_5 \oplus \delta^{29.8}\gamma\gamma_5 \oplus \delta^{42.25}\gamma^2\gamma_5 \\
A_{66}^{(42)} &= \delta^{6.7}\gamma_6 \oplus \delta^{28.45}\gamma\gamma_6 \oplus \delta^{41.55}\gamma^2\gamma_6
\end{aligned}$$

On pourrait vérifier que le dernier coefficient $A_{66}^{(42)}$ représente effectivement des circuits passant par la transition de sortie de la pièce 6 et par les transitions internes. En particulier, on a le monôme $\delta^{6.7}\gamma_6$ qui correspond au circuit physique la pièce 6, ce qui est conforme avec la 6 ième colonne du tableau IX.1. On obtient alors en 20 mn à l'issue de cette énumération sans simplifications 338 circuits qui se réduisent via BASILE aux mêmes 25 circuits. Cette méthode est donc provisoirement plus rapide.

En traduisant l'ensemble minimal de 25 circuits obtenu plus haut, on obtient l'expression formelle suivante du taux de production comme min de 25 fonctions affines:

$\lambda(q) = \min(0.1562500000q_4, 0.1492537313q_6, 0.09615384615q_1, 0.08130081301,$
$0.07874015748q_2, 0.07575757576q_5,$
$0.05899705015q_3,$
$0.03533568905 + 0.03533568905q_4,$
$0.03490401396 + 0.03490401396q_6,$
$0.03478260870 + 0.03478260870q_1,$
$0.03472222222 + 0.03472222222q_5,$
$0.03372681282 + 0.03372681282q_2,$
$0.03327787022 + 0.03327787022q_3,$
$0.04907975460 + 0.02453987730q_4,$
$0.04848484848 + 0.02424242424q_5,$
$0.02418379686 + 0.02418379686q_4 + 0.02418379686q_6,$
$0.02412545235 + 0.02412545235q_1 + 0.02412545235q_4,$
$0.04790419162 + 0.02395209581q_6,$
$0.04778972520 + 0.02389486260q_1,$
$0.02333722287 + 0.02333722287q_2 + 0.02333722287q_6,$
$0.02328288708 + 0.02328288708q_1 + 0.02328288708q_2,$
$0.02229654404 + 0.02229654404q_2 + 0.02229654404q_4,$
$0.02205071665 + 0.02205071665q_2 + 0.02205071665q_5,$
$0.01800180018 + 0.01800180018q_2 + 0.01800180018q_4 + 0.01800180018q_6,$
$0.01796945193 + 0.01796945193q_1 + 0.01796945193q_2 + 0.01796945193q_4)$

Cohen, Dubois, Quadrat et Viot ont appliqué l'algorithme suivant pour saturer la machine menante:

5.2.1 Algorithme

1/ commencer avec 1 palette par type de pièce
2/ calculer le taux de production
3/ si la machine menante n'est pas saturée, rajouter une palette sur un circuit critique, et recommencer en 2/.

Cohen, Dubois, Quadrat et Viot calculaient le taux de production à chaque étape par l'algorithme de Karp (cf. VIII,2.2.2). Nous calculons ici ce taux de manière immédiate par évaluation de l'expression formelle de λ ci-dessus. Nous reprenons la suite de répartitions des palettes obtenue dans [23] par application de 5.2.1, et donnons à chaque étape le taux de production. Les résultats coïncident avec ceux de [23].

répartition des palettes $(q_1, q_2, q_3, q_4, q_5, q_6)$	λ (unité de temps)
$(1,1,1,1,1,1)$	16.95
$(1,1,2,1,1,1)$	15.11666667
$(1,1,2,1,2,1)$	14.95
$(1,2,2,1,2,1)$	14.375
$(2,2,2,1,2,1)$	14.325
$(2,2,2,1,2,2)$	14.15
$(2,2,2,2,2,2)$	12.3

Ainsi, on trouve que la machine menante est saturée pour la dernière répartition (deux palettes pour chaque type de pièce). Cependant, par l'algorithme itératif 5.2.1, on ne peut savoir s'il y a d'autres répartitions minimales saturantes. Nous montrons ce résultat à l'aide de l'expression formelle de λ.

5.2.2 Proposition *L'unique répartition minimale des palettes saturant la machine menante est*

$$q = (2,2,2,2,2,2) \ .$$

Preuve La machine menante est saturée ssi $\sigma(c)$ est égale à la pente correspondante à la partie constante de c, i.e.

$$\sigma(\gamma\delta^{12.3}) = \frac{1}{12.3} \ ,$$

ce qui est réalisé ssi pour tout monôme m de c, on a $\sigma(m) \geq \frac{1}{12.3}$. Nous inspectons successivement les monômes, et prenons la plus petite valeur entière de q satisfaisant les inégalités.

$$\begin{array}{rcll}
\sigma(\delta^{6.4}\gamma_4) &=& \frac{q_4}{6.6} \geq \frac{1}{12.3} & \Rightarrow \quad q_4 \geq 1 \\
\sigma(\delta^{6.7}\gamma_6) &=& \frac{q_6}{6.7} \geq \frac{1}{12.3} & \Rightarrow \quad q_6 \geq 1 \\
\sigma(\delta^{10.4}\gamma_1) &=& \frac{q_1}{10.4} \geq \frac{1}{12.3} & \Rightarrow \quad q_1 \geq 1 \\
\sigma(\delta^{12.7}\gamma_2) &=& \frac{q_2}{10.4} \geq \frac{1}{12.3} & \Rightarrow \quad q_2 \geq 2 \\
\sigma(\delta^{13.2}\gamma_5) &=& \frac{q_5}{13.2} \geq \frac{1}{12.3} & \Rightarrow \quad q_5 \geq 2 \\
\sigma(\delta^{16.95}\gamma_3) &=& \frac{q_3}{16.95} \geq \frac{1}{12.3} & \Rightarrow \quad q_3 \geq 2 \\
\sigma(\delta^{28.3}\gamma\gamma_4) &=& \frac{q_4+1}{28.3} \geq \frac{1}{12.3} & \Rightarrow \quad q_4 \geq 2 \\
\sigma(\delta^{28.65}\gamma\gamma_6) &=& \frac{q_6+1}{28.65} \geq \frac{1}{12.3} & \Rightarrow \quad q_6 \geq 2 \\
\sigma(\delta^{28.75}\gamma\gamma_1) &=& \frac{q_1+1}{28.75} \geq \frac{1}{12.3} & \Rightarrow \quad q_1 \geq 2
\end{array}$$

Ainsi, une répartition saturante vérifie nécessairement $q \geq (2,2,2,2,2,2)$. On a vu plus haut que cette répartition saturait effectivement la machine menante, d'où le résultat. ∎

5.2.3 Remarque En général, on n'a pas unicité de la répartition minimale des ressources. Soit par exemple l'ensemble de circuits

$$c' = \gamma\delta^5 \oplus \delta^{10}\gamma_1\gamma_2 \ .$$

On trouve en raisonnant comme ci dessus les trois répartitions minimales suivante

$$q = (q_1, q_2) = (2,0), \quad (1,1), \quad (0,2) \ .$$

6 Formulation en termes de vecteurs propres

Classiquement, on a au moins deux points de vue pour les problèmes de valeurs propres: a/ passer par le polynôme caractéristique, et b/ passer par une étude des vecteurs propres. Les développements qui précédent sont l'analogue de a/ (cf. en particulier 3.1.12). Nous allons voir que l'approche b/ conduit ici à un nouveau problème de programmation linéaire.

6.1 Equivalence à la recherche d'un sous-vecteur propre

6.1.1 Position du problème Dans la suite, nous considérons un graphe d'événements temporisé, représenté de la manière suivante: T_{ij} représente le temps de séjour dans la place[3] (ij). N_{ij} représente le nombre de jetons du marquage initial de cette place. Si ce marquage est fixé, N_{ij} est une constante. Si la place (ij) correspond à une ressource l, on aura $N_{ij} = q_l$, où q_l est l'indéterminée associée à cette ressource. On notera $N(q)$ au lieu de N pour rappeler cette dépendance. On suppose que les transitions sont temporisées à 0, ce qui ne restreint pas la généralité (cf. [23]). Quitte à écrire autant de problèmes d'optimisation des ressources qu'il y a de composantes connexes dans le graphe, on pourra supposer le graphe connexe. On notera $\lambda(q)$ le taux de production en fonction des ressources, et l'on se donne λ_0 un taux de production souhaité.

Le résultat central est le suivant:

6.1.2 Théorème *Pour le système décrit en 6.1.1, les assertions suivantes sont équivalentes:*

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

(ii) $\rho(C(q)) \preceq e$ *dans* $\mathbb{R}_{\min}$*, où la matrice* $C(q)$ *est donnée par* $C_{ij}(q) = N_{ij}(q) - \lambda_0 T_{ij}$.

(iii) *il existe un vecteur* u *à coefficients dans* $\mathbb{R}_{\min}$ *tel que* $C(q) \otimes u \preceq u$.

On a noté $\rho(C)$ le rayon spectral de la matrice C dans $\mathbb{R}_{\min}$ (cf. IV,1.3.1), i.e. dans l'algèbre habituelle:

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

La condition (iii) se réécrit dans l'algèbre habituelle[4]

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

soit

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

En remplaçant $N_{ij}(q)$ par l'inconnue q_l s'il y a lieu (cf. 6.1.1), on constate qu'il s'agit là d'un ensemble de contraintes linéaires en les variables u_i et q_l. On a donc comme conséquence immédiate de 6.1.2:

6.1.3 Corollaire *Le problème d'optimisation des ressources §1.3 est équivalent au problème de programmation linéaire suivant: "minimiser* $J(q)$*, sous l'ensemble de contraintes (6.1.b), où* $\forall i,\ u_i \in \mathbb{R}$ *et* $\forall l, q_l \in \mathbb{N}$*".*

[3] i.e. la place en aval de la transition j et en amont de la transition i

[4] On rappelle que $\preceq$ est l'ordre dual de $\leq$

Preuve du Théorème 6.1.2. (i)$\Leftrightarrow$(ii). Etant donné une matrice B, on définit le poids $w_B(\alpha)$ du circuit $\alpha = (i_1, \ldots, i_k)$ par $w_B(\alpha) = B_{i_1 i_2} + \ldots + B_{i_k i_1}$ (cf. IV,§1.3,(1.3.d)). D'après VII,§6.6,(6.6.a) on a

$$\lambda(q) = \min_{\alpha \text{ circuit}} \frac{w_N(\alpha)}{w_T(\alpha)} \ . \tag{6.1.c}$$

Les assertions suivantes sont équivalentes:

	$\lambda(q) \geq \lambda_0$	
Pour tout circuit α,	$\frac{w_N(\alpha)}{w_T(\alpha)} \geq \lambda_0$	(par (6.1.c))
Pour tout circuit α,	$w_N(\alpha) \geq \lambda_0 w_T(\alpha)$	
Pour tout circuit α,	$w_N(\alpha) - \lambda_0 w_T(\alpha) \geq 0$	
Pour tout circuit α,	$w_C(\alpha) \geq 0$	
	$\rho(C) \geq 0.$	(d'après (6.1.a)).

On a montré (i)$\Leftrightarrow$(ii). L'équivalence de (ii) et de (iii) résulte du Lemme 1.3.8 du Chapitre IV. ■

6.2 Lien avec l'algorithme de Proth et Xie

Proth et Xie [83] ont proposé un algorithme réduisant le problème de l'optimisation des ressources à un problème de programmation linéaire mixte. Notre corollaire 6.1.3 constitue une généralisation de ce résultat. L'algorithme optimal que Proth et Xie donnent dans [83] n'est en effet valable que moyennant des restrictions fortes: 1/ il faut se restreindre à une sous classe de graphes d'événements temporisés [5] 2/ il faut supposer que *tous* les marquages initiaux sont indéterminés, ce qui revient à prendre autant de ressources inconnues que de places[6] 3/ il faut supposer que le critère est invariant par tir des transitions[7]. Ces restrictions sont rendues inutiles par le corollaire 6.1.3. En outre, la preuve algébrique de 6.1.3 nous paraît plus simple.

Nous conclurons ce chapitre en comparant les algorithmes d'énumération de circuits et la généralisation 6.1.3 de l'algorithme de Proth et Xie. Il est clair que lorsque le nombre de ressources inconnues est de l'ordre du nombre de places, l'algorithme 6.1.3 sera préférable. La simplification modulo les congruences polyédriques devient en effet rédhibitoire dès que la dimension (i.e. le nombre de ressources) est grande. Par contre, lorsque le nombre de ressources est petit devant la taille du système (i.e. le nombre de places) on pourra via les technique d'agrégation décrites en §4.1, se ramener à un problème de programmation linéaire de type (1.3.a) dont le nombre d'inconnues est égal au nombre de ressources, et non pas à la taille du système. L'algorithme donné en §4.1 sera alors préférable. On aura en outre, via l'expression formelle $\lambda(q)$, une information plus précise sur le système.

[5] cf. [83], p. 797 "dans la suite, nous supposerons que le début d'un franchissement n'est possible que si la transition considérée n'est pas active". Une telle restriction interdit par exemple de modéliser le problème suivant: "sachant qu'une machine peut fabriquer une pièce par heure et que plusieurs machines peuvent travailler en parallèle, combien faut il de machines pour produire 10 pièces par heure?"

[6] Une telle restriction est gênante si une partie du système est "fixée" et non indéterminée. Laftit, Proth et Xie doivent alors recourir à des techniques de pénalisation pour traiter ces cas (cf. [59]).

[7] Moyennant la restriction 2/, l'allocation des ressources q coïncide avec le marquage initial. On suppose que si q' est un marquage accessible depuis q, alors $J(q') = J(q)$. Cette restriction pose, entre autres, des problèmes d'initialisation du système. Cela signifie par exemple que l'on prend comme initial un état où les palettes peuvent être à un endroit quelconque de leur cycle, et portent donc des pièces semi-finies.