

## GÉNÉRALISATION MAX-PLUS DES BORNES DE LAGEWEG, LENSTRA ET RINNOOY KAN

CHRISTOPHE LENTÉ<sup>1</sup> ET JEAN-LOUIS BOUQUARD<sup>1</sup>

**Abstract.** The traditional flowshop scheduling problem can be generalised to a matricial optimisation problem in Max-Plus algebra. A family of lower bounds is developed for this new problem and proof is given that these bounds are a generalisation of the lower bounds of Lageweg *et al.*

**Résumé.** Le traditionnel problème d'ordonnement de type flowshop se généralise en un problème d'optimisation matricielle dans l'algèbre Max-Plus. Une famille de bornes inférieures est présentée pour ce nouveau problème et la preuve est apportée que ces bornes généralisent les bornes de Lageweg *et al.*

### INTRODUCTION

Les problèmes d'ordonnement de type flowshop ont été beaucoup étudiés. En premier lieu les problèmes les plus simples ont été abordés puis les chercheurs ont progressivement tenté de prendre en compte de plus en plus de contraintes. Cette évolution est particulièrement sensible en ce qui concerne les problèmes à deux machines.

Il est possible, en utilisant l'algèbre Max-Plus, de modéliser indifféremment des problèmes d'ordonnement de type flowshop avec des contraintes de temps de montage et de démontage indépendants de la séquence, des décalages temporels ou des lots prédéfinis [18]. Si le critère à minimiser est le  $C_{\max}$ , ces problèmes se réduisent tous à l'optimisation d'un produit de matrices triangulaires Max-Plus. On définit ainsi un problème d'optimisation dans l'algèbre Max-Plus qui généralise les problèmes de flowshop listés ci-dessus. Dans le cas de matrices d'ordre deux, qui

---

<sup>1</sup> Laboratoire d'Informatique de l'Université de Tours, Polytech'Tours/DI.

correspond à des flowshops à deux machines, on résoud le problème d'optimisation dans Max-Plus par un algorithme proche de celui de Johnson [16]. Cet algorithme unifie de nombreux résultats éparses de la littérature de ces 50 dernières années.

Dans le cas de matrices d'ordre quelconque, le problème est NP-difficile. Il faut donc développer des heuristiques, des bornes inférieures et supérieures et des méthodes exactes de résolution. Nous nous proposons de montrer dans cet article comment généraliser les bornes de Lageweg, Lenstra et Rinnooy Kan [17]. Ces bornes ont été définies dans le cadre d'un flowshop sans contrainte, mais il est possible de les adapter au problème matriciel en extrayant un produit de matrices d'ordre 2 et en utilisant la notion de résiduation matricielle. Par voie de conséquence, cela signifie qu'il est possible d'étendre les bornes de Lageweg *et al.* [17] à des problèmes de flowshop soumis aux contraintes citées précédemment.

Cet article est divisé en cinq parties. La première partie propose un rapide survol des algèbres tropicales, la deuxième répertorie un certain nombre de travaux antérieurs rapprochant l'ordonnancement et ces algèbres tropicales, la troisième définit précisément le problème de flowshop étudié, la modélisation Max-Plus d'un flowshop est succinctement présentée dans la partie suivante et le calcul des bornes inférieures est détaillé dans la dernière partie.

## 1. ALGÈBRES TROPICALES

Ce qui suit est une présentation succincte de ce que l'on appelle *algèbres tropicales* et plus particulièrement *algèbre Max-Plus*. L'objectif n'est en fait que de présenter les quelques éléments indispensables à la compréhension de l'article. Pour de plus amples informations, on peut par exemple se référer à [1, 9, 12, 13].

### 1.1. GÉNÉRALITÉS

Les trois définitions qui suivent portent sur un ensemble  $E$  muni de deux lois internes notées  $\oplus$  et  $\otimes$ .

**Définition 1.1** (demi-anneau).  $(E, \oplus, \otimes)$  est un demi-anneau ssi  $\oplus$  est commutative, associative et admet un élément neutre ( $\mathbf{0}$ ),  $\otimes$  est associative et admet un élément neutre ( $\mathbf{1}$ ),  $\otimes$  est distributive sur  $\oplus$  à droite et à gauche, et  $\mathbf{0}$  est absorbant pour  $\otimes$ .

L'élément  $\mathbf{0}$  est appelé l'élément nul et  $\mathbf{1}$  est appelé l'élément unité. L'écriture  $a \otimes b$  est souvent simplifiée en  $a.b$  ou en  $ab$ .

**Définition 1.2** (dioïde).  $(E, \oplus, \otimes)$  est un demi-anneau idempotent ou dioïde ssi  $(E, \oplus, \otimes)$  est un demi-anneau et  $\oplus$  est idempotente.

**Définition 1.3** (demi-corps).  $(E, \oplus, \otimes)$  est un demi-corps ssi  $(E, \oplus, \otimes)$  est un demi-anneau et  $(E \setminus \{\mathbf{0}\}, \otimes)$  est un groupe.

L'étude menée dans les paragraphes suivants repose sur le demi-corps idempotent  $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$  également appelé *algèbre Max-Plus*.

L'opérateur  $max$  est noté additivement  $\oplus$  et l'opérateur  $+$  désignant l'addition usuelle est noté multiplicativement  $\otimes$ . L'élément nul est  $-\infty$ , noté  $\mathbf{0}$  et l'élément unité est  $0$ , noté  $\mathbf{1}$ . Lorsque l'on a besoin de faire référence au nombre  $0$  on le note sous forme de chiffre, les termes élément nul ou zéro font forcément référence à  $\mathbf{0}$ . Par convention et par commodité, dans le reste du texte, le signe ' $\equiv$ ' signale la traduction d'une expression de l'algèbre Max-Plus dans l'algèbre usuelle ou inversement. Par exemple :  $(a \oplus b) \otimes c = a \otimes c \oplus b \otimes c \equiv max(a, b) + c = max(a + c, b + c)$ .

De par sa structure, un dioïde définit naturellement un ordre partiel sur ses éléments.

**Définition 1.4** (ordre canonique). Tout dioïde est muni de la relation d'ordre partielle

$$\forall(x, y) \in E^2, x \preceq y \Leftrightarrow x \oplus y = y. \quad (1)$$

Cet ordre partiel est compatible avec les deux lois internes.

Dans  $\mathbb{R}_{\max}$  cet ordre est total et correspond à l'ordre naturel des réels, mais il restera néanmoins noté  $\preceq$  dans un souci de généralité. Les propriétés développées ci-dessous demeurent valables dans tout demi-corps idempotent dont la première loi vérifie  $\forall a, b : a \oplus b \in \{a, b\}$ .

## 1.2. MATRICES

On est amené dans la suite à définir de nombreuses matrices. La plupart dépendent d'un paramètre, ce sont plus exactement des fonctions matricielles d'un ensemble de travaux ou de séquences sur un dioïde, d'autres sont des transformées de ces matrices.

### 1.2.1. Convention de notations

Afin de préserver une certaine cohérence et pour éviter de devoir préciser les notations à chaque nouvelle matrice, on convient de désigner les matrices par des lettres capitales et leurs composantes par les lettres minuscules correspondantes. Ainsi  $a_{i,j}$  est une composante de la matrice  $A$ .

Mais pour des raisons qui apparaissent au paragraphe 4, on choisit également de désigner les composantes d'abord par indice de colonne puis par indice de ligne, les termes diagonaux pouvant ne comporter qu'un unique indice. Pour aider le lecteur à garder en mémoire cette convention, les initiales  $c$  et  $\ell$  sont aussi souvent que possible utilisées pour indiquer les composantes.

**Exemple 1.5.**  $A = \begin{pmatrix} a_1 & \mathbf{0} & \mathbf{0} \\ a_{1,2} & a_2 & \mathbf{0} \\ a_{1,3} & a_{2,3} & a_3 \end{pmatrix}, \tilde{M}(\sigma) = \begin{pmatrix} \tilde{m}_1(\sigma) & \mathbf{0} & \mathbf{0} \\ \tilde{m}_{1,2}(\sigma) & \tilde{m}_2(\sigma) & \mathbf{0} \\ \tilde{m}_{1,3}(\sigma) & \tilde{m}_{2,3}(\sigma) & \tilde{m}_3(\sigma) \end{pmatrix},$

la matrice  $R^{(u,v)}(k)$  est composée des éléments  $r_{c,\ell}^{(u,v)}(k)$ .

La notation employée en particulier dans [12] est aussi utilisée, la composante à l'intersection de la ligne  $\ell$  et de la colonne  $c$  de la matrice  $A$  est notée  $[A]_{\ell,c}$ , ainsi  $r_{c,\ell}^{(u,v)}(k) = [R^{(u,v)}(k)]_{\ell,c}$ .

### 1.2.2. Algèbres de matrices

Les structures de demi-anneau et de dioïde peuvent se transmettre à des ensembles de matrices, il suffit pour cela d'étendre correctement les lois internes. Soient  $A$  et  $B$  deux matrices  $n \times m$ , on définit  $C = A \oplus B$  par

$$\forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}, [C]_{i,j} = [A]_{i,j} \oplus [B]_{i,j}. \quad (2)$$

Soient  $A$  une matrice  $n \times m$  et  $B$  une matrice  $m \times p$ , on définit  $C = A \otimes B$  par

$$\forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}, [C]_{i,j} = \bigoplus_{k=1}^m [A]_{i,k} \otimes [B]_{k,j}. \quad (3)$$

**Théorème 1.1.** *Soit  $\mathcal{M}_n(E)$  l'ensemble des matrices carrées d'ordre  $n$  sur  $E$ . Si  $(E, \oplus, \otimes)$  est un demi-anneau alors  $(\mathcal{M}_n(E), \oplus, \otimes)$  en est un également. Si  $(E, \oplus, \otimes)$  est un dioïde alors  $(\mathcal{M}_n(E), \oplus, \otimes)$  en est un également.*

### 1.3. RÉSIDUATION

La résiduation est une théorie vaste [4], on ne retiendra ici que la possibilité de définir une sorte de division dans les dioïdes complets et ainsi de pallier la non inversibilité des matrices. Cette opération n'est pas toujours définissable mais elle existe dans l'algèbre Max-Plus, on en rappelle les propriétés principales.

**Définition 1.6.** Un ensemble partiellement ordonné est complet si et seulement si chacun de ses sous-ensembles admet une borne supérieure.

**Définition 1.7.** Un dioïde est complet si et seulement si il est complet en tant qu'ensemble ordonné et il vérifie les propriétés de distributivité infinie :

$$\forall X \subset E, \forall y \in E, \left( \bigoplus_{x \in X} x \right) y = \bigoplus_{x \in X} (xy) \text{ et } y \left( \bigoplus_{x \in X} x \right) = \bigoplus_{x \in X} (yx).$$

Étant données les équations  $ax = b$  et  $xa = b$  :

**Définition 1.8.** On appelle résidué à gauche le nombre  $a \setminus b = \bigoplus \{x \mid ax \preceq b\}$  et résidué à droite le nombre  $b / a = \bigoplus \{x \mid xa \preceq b\}$ .

Ainsi  $a(a \setminus b) \preceq b$  et  $(b/a)a \preceq b$ . Il faut noter que si  $a$  est inversible alors  $a \setminus b = a^{-1}b$  et  $b/a = ba^{-1}$ .

En particulier, dans le dioïde complet  $\overline{\mathbb{R}}_{max} = (\mathbb{R} \cup \{-\infty, +\infty\}, max, +)$  :

$$\forall a, b \in \mathbb{R}, a \setminus b = b/a = \frac{b}{a} \equiv b - a$$

$$\forall a \in \overline{\mathbb{R}}_{max}, a \setminus \infty = \infty/a = \infty$$

$$\forall b \in \mathbb{R}_{max}, \infty \setminus b = b/\infty = \mathbf{0}$$

$$\forall b \in \overline{\mathbb{R}}_{max}, \mathbf{0} \setminus b = b/\mathbf{0} = \infty$$

$$\forall a \in \mathbb{R} \cup \{\infty\}, a \setminus \mathbf{0} = \mathbf{0}/a = \mathbf{0}.$$

Un théorème général du à Blyth [3] permet de calculer la résiduée de deux matrices. On utilise la notation traditionnelle ( $\wedge$ ) pour représenter l'opérateur *inf*.

**Théorème 1.2.** Soient  $A \in \mathcal{M}_n(\overline{\mathbb{R}}_{max})$  et  $B \in \mathcal{M}_n(\overline{\mathbb{R}}_{max})$

$$[A \setminus B]_{\ell,c} = \bigwedge_{k=1}^n [A]_{k,\ell} \setminus [B]_{k,c} \quad \text{et} \quad [B/A]_{\ell,c} = \bigwedge_{k=1}^n [B]_{\ell,k} / [A]_{c,k}.$$

**Exemple 1.9.** Soient les deux matrices

$$A = \begin{pmatrix} 2 & \mathbf{0} \\ \infty & 3 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} \mathbf{0} & \infty \\ 1 & 4 \end{pmatrix}.$$

On a

$$A \setminus B = \begin{pmatrix} 2 \setminus \mathbf{0} \wedge \infty \setminus 1 & 2 \setminus \infty \wedge \infty \setminus 4 \\ \mathbf{0} \setminus \mathbf{0} \wedge 3 \setminus 1 & \mathbf{0} \setminus \infty \wedge 3 \setminus 4 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \wedge \mathbf{0} & \infty \wedge \mathbf{0} \\ \infty \wedge -2 & \infty \wedge 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -2 & 1 \end{pmatrix}$$

et cette matrice est la plus grande matrice  $X$  telle que  $AX \preceq B$ .

On a également

$$B/A = \begin{pmatrix} \mathbf{0}/2 \wedge \infty/\mathbf{0} & \mathbf{0}/\infty \wedge \infty/3 \\ 1/2 \wedge 4/\mathbf{0} & 1/\infty \wedge 4/3 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \wedge \infty & \mathbf{0} \wedge \infty \\ -1 \wedge \infty & \mathbf{0} \wedge 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -1 & \mathbf{0} \end{pmatrix}$$

c'est la plus grande matrice  $X$  telle que  $XA \preceq B$ .

## 2. ALGÈBRES TROPICALES ET ORDONNANCEMENT

Plusieurs auteurs se sont appuyés sur les théories tropicales pour étudier des problèmes d'ordonnancement. Giffler [11] a développé deux algèbres originales pour travailler sur des ordonnancements modélisés par des graphes. Il montre comment une modification mineure des données ou du graphe peut être prise en compte par un simple calcul matriciel, mais ses travaux ne semblent pas avoir été poursuivis.

Une grande partie des travaux porte sur les problèmes d'ordonnancement cycliques, qui peuvent être abordés au moyen de la théorie spectrale des algèbres. Dans [14], Hanen et Munier étudient des problèmes d'ordonnancement cyclique à machines parallèles avec contraintes de précédences, dans [6] les auteurs abordent un problème cyclique de type flowshop, repris ensuite par Gaubert [9] pour calculer des taux de productions et la répartition optimale de certaines ressources afin de saturer au mieux les machines. Gaubert et Mairesse [10] calculent, dans un problème de jobshop généralisé, les dates de fin des opérations pour une séquence donnée. Ils produisent une très belle modélisation basée sur la construction d'un tas de pièces, ressemblant à un jeu de Tetris. Ils sont malheureusement obligés de manipuler des matrices dont l'ordre est égal à la somme du nombre d'opérations et du nombre de machines.

### 3. FLOWSHOP

Le problème étudié est bien défini depuis longtemps, il s'agit d'un flowshop de permutation avec temps de montage et de démontage indépendants de la séquence, décalages temporels et traitement par lots ( $Fm/prmu, s_{nsd}, r_{nsd}, a_{i,j}, batch/C_{max}$ ). L'atelier est composé de  $m$  machines, numérotées de 1 à  $m$ , chaque travail  $x$  se décompose en  $m$  opérations dont les durées opératoires sont notées  $p_{x,k}$ . Tous les travaux suivent la même et unique gamme linéaire. Un temps de montage noté  $s_{x,k}$  symbolise le temps de préparation de la  $k^{\text{ème}}$  machine nécessaire à l'exécution du travail  $x$  et un temps de démontage  $r_{x,k}$  symbolise le temps de remise en état de la machine après l'exécution du travail. Durant ces deux périodes, la machine est indisponible pour toute autre activité. Le décalage temporel  $a_{x,k}$  indique le temps minimal qui doit s'écouler entre la fin d'un travail sur une machine  $M_{k-1}$  et le début de ce travail sur la machine suivante. Les lots sont des groupes de travaux qui doivent s'exécuter consécutivement, on suppose que l'ordre des travaux à l'intérieur des lots est connu à l'avance, le problème est alors de déterminer l'ordre optimal de passage des lots. Il peut exister des temps de montage et de démontage majeurs avant et après chaque lot en plus de ceux qui encadrent les opérations. L'objectif est de terminer au plus tôt l'exécution des  $n$  travaux, autrement dit de minimiser le  $C_{max}$ .

Ce problème est NP-difficile dès qu'il comporte trois machines ou plus [8], le problème à deux machines se résout par contre en temps polynomial.

Dans un flowshop de permutation, les travaux passent dans le même ordre sur toutes les machines, donc dans le problème étudié, un ordonnancement est entièrement défini par une séquence de travaux. En effet, le  $C_{max}$  est un critère régulier donc, à séquence donnée, il suffit d'exécuter les opérations au plus tôt tout en respectant les contraintes du problème pour obtenir un ordonnancement potentiellement optimal. On note  $C_{i,k}(\sigma)$  la date de fin du  $i^{\text{ème}}$  travail de la séquence  $\sigma$  sur la machine  $M_k$  (temps de démontage inclus), c'est aussi la date de fin de la  $k^{\text{ème}}$  opération de ce travail dans l'ordonnancement défini par  $\sigma$ . Cette date est en fait mal définie car elle dépend de la date de début des opérations, sans précision

contraire, on supposera que la première opération de la séquence a débuté à la date 0 et que rien ne vient perturber l'exécution de la séquence. Les notations précédentes sont étendues, ainsi  $p_{i,k}(\sigma)$  est la durée d'exécution du  $i^{\text{ème}}$  travail de la séquence  $\sigma$  sur la machine  $M_k$  et  $a_{i,k}(\sigma)$ ,  $s_{i,k}(\sigma)$  et  $r_{i,k}(\sigma)$  sont définis de façon similaire.

On distingue ce qui fait directement référence à un travail et qui est une donnée de ce qui fait référence à une position dans une séquence et qui est donc une fonction définie sur l'ensemble des séquences.

#### 4. MODÉLISATION MAX-PLUS D'UN FLOWSHOP

Dans cette partie, la modélisation Max-Plus d'un flowshop [18] est rapidement décrite, suivie des principales propriétés qui en découlent.

##### 4.1. MATRICES ASSOCIÉES À UN TRAVAIL

**Définition 4.1.** Les matrices suivantes sont associées à tout travail  $x$ .

Une matrice diagonale de temps de montage  $S_x = S(x) = \text{diag}[s_{x,1}, s_{x,2}, \dots, s_{x,m}]$ .

Une matrice diagonale de temps de démontage  $R_x = R(x) = \text{diag}[r_{x,1}, r_{x,2}, \dots, r_{x,m}]$ .

Une matrice de durées d'exécution

$$P_x = P(x) = \begin{pmatrix} p_{x,1} & \mathbf{0} & \dots & \mathbf{0} \\ p_{x,1}a_{x,2}p_{x,2} & p_{x,2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ p_{x,1}a_{x,2}p_{x,2}a_{x,3} \dots a_{x,m}p_{x,m} & p_{x,2}a_{x,3} \dots a_{x,m}p_{x,m} & \dots & p_{x,m} \end{pmatrix}$$

ou plus formellement

$$\forall (c, \ell) \in \{1, \dots, m\}^2, \begin{cases} \text{si } \ell < c \text{ alors } [P_x]_{\ell,c} = \mathbf{0} \\ \text{si } \ell \geq c \text{ alors } [P_x]_{\ell,c} = p_{x,c} \bigotimes_{k=c+1}^{\ell} (a_{x,k}p_{x,k}). \end{cases}$$

Et la matrice  $T_x = T(x) = R(x)P(x)S(x)$ .

Les résultats à venir présupposent que les opérations du travail  $x$  s'exécutent dès que possible sans interférence extérieure. Si ce n'est pas le cas, on n'obtient que les dates de fin au plus tôt des opérations.

**Propriété 4.2.** Soit  $\vec{t}$  le vecteur des dates de disponibilité au plus tôt des machines, les dates de fin des opérations d'un travail  $x$  sont données par la formule

$$\vec{C}_x = T(x)\vec{t}. \quad (4)$$

**Corollaire 4.3.** *Si toutes les machines sont libres à la date 0, les dates de fin des opérations d'un travail  $x$  sont données par la formule*

$$\vec{C}_x = T(x) \begin{pmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{pmatrix}. \quad (5)$$

En effet, dans ce cas le vecteur des dates de disponibilité des machines est composé uniquement de  $\mathbf{1}$  ( $\mathbf{1} \equiv 0$ ).

#### 4.2. MATRICES ASSOCIÉES À UNE SÉQUENCE

**Définition 4.4.** Étant donnée une séquence  $\sigma$  :

$T_i(\sigma)$  est la matrice associée au  $i^{\text{ème}}$  travail de  $\sigma$  ( $T_i(\sigma) = T(\sigma(i))$ ). Le même principe de notation est appliqué aux deux autres matrices (montage et démontage);  $\vec{C}_{i..}(\sigma) = (C_{i,1}(\sigma), \dots, C_{i,m}(\sigma))^t$  est le vecteur des dates de fin des opérations du  $i^{\text{ème}}$  travail de  $\sigma$  et  $\vec{C}_\sigma$  le vecteur des dates de fin des opérations du dernier travail de  $\sigma$  ( $\vec{C}_\sigma = \vec{C}_{|\sigma|..}(\sigma)$ ).

À toute séquence peut être associée une matrice.

**Définition 4.5.**  $\forall \sigma$  séquence,  $T(\sigma) = \bigotimes_{i=|\sigma|}^1 T(\sigma(i)) = \bigotimes_{i=|\sigma|}^1 T_i(\sigma)$ .

Si la séquence est de longueur 1, sa matrice est la matrice associée au travail qui la compose. Du point de vue matriciel il n'y a donc pas de différence sensible entre séquences et travaux. Une séquence peut être vue comme l'extension de la notion de travail ou un travail comme une séquence de longueur 1.

Du point de vu matriciel, il n'existe pas de différence entre des lots (*cf.* Sect. 3) et des travaux, c'est ce qui nous permet de traiter la contrainte "batch".

**Propriété 4.6.** *Soit  $\sigma$  la concaténation de deux séquences  $\sigma_1$  et  $\sigma_2$  :*

$$T(\sigma) = T(\sigma_1\sigma_2) = T(\sigma_2)T(\sigma_1).$$

Cette propriété est une conséquence de l'associativité du produit matriciel.

**Propriété 4.7.** *Pour toute séquence  $\sigma$ ,*

$$\forall i \in \{1, \dots, |\sigma|\}, \vec{C}_{i..}(\sigma) = T_i(\sigma) \vec{C}_{i-1..}(\sigma). \quad (6)$$

*Démonstration.* Le  $i^{\text{ème}}$  travail de la séquence ne peut commencer au plus tôt qu'aux dates  $\vec{t} = \vec{C}_{i-1..}(\sigma)$ , il ne reste qu'à appliquer la propriété 4.2.  $\square$

**Propriété 4.8.** Soit  $\vec{t}$  le vecteur des dates de disponibilité au plus tôt des machines et  $\sigma$  une séquence à exécuter.

$$\vec{C}_\sigma = T(\sigma) \vec{t}. \quad (7)$$

Cette propriété n'est qu'un corollaire de la propriété précédente.

**Corollaire 4.9.** Si toutes les machines sont libres à la date 0, les dates de fin des opérations du dernier travail d'une séquence  $\sigma$  sont

$$\vec{C}_\sigma = T(\sigma) \begin{pmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{pmatrix}. \quad (8)$$

**Corollaire 4.10.** Pour toute séquence  $\sigma$ ,

$$C_{max}(\sigma) = \left( \mathbf{1} \quad \dots \quad \mathbf{1} \right) T(\sigma) \begin{pmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{pmatrix}. \quad (9)$$

*Démonstration.*  $C_{max}(\sigma) = \bigoplus_{k=1}^m C_{|\sigma|,k}$ , c'est exactement ce que calcule la formule (9). □

On constate que la résolution d'un problème de flowshop revient à l'optimisation d'un produit matriciel. La modélisation présentée permet d'aborder indifféremment des problèmes avec ou sans contraintes, elle offre donc une démarche unificatrice en gommant les différences induites par la présence de telle ou telle contrainte.

## 5. LE PROBLÈME MATRICIEL

Le problème étudié se résume ainsi : on dispose d'une famille  $\mathcal{M}$  de  $n$  matrices Max-Plus triangulaires inférieures ( $\mathcal{M} = (M(k))_{1 \leq k \leq n} \in [(\mathcal{M}_m(\mathbb{R}_{\max}))]^n$ ), à toute séquence  $\sigma$  sont associés la matrice  $M(\sigma) = \bigotimes_{k=n}^1 M(\sigma(k))$  et le critère

$$\Gamma(\sigma) = \left( \mathbf{1} \quad \dots \quad \mathbf{1} \right) M(\sigma) \begin{pmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{pmatrix} \text{ et on cherche une séquence minimisant ce critère.}$$

Ce problème est polynomial lorsque les matrices sont d'ordre 2 et il est clairement NP-difficile au delà puisque le problème de flowshop est NP-difficile pour trois machines et plus. Une minoration, ou borne inférieure, du critère  $\Gamma$  est proposée qui

généralise les bornes inférieures proposées par Lageweg, Lenstra et Rinnooy Kan pour le flowshop sans contrainte [17].

### 5.1. MATRICES D'ORDRE 2

La recherche d'une séquence optimale passe par celle d'une règle d'échange locale [19, 20].

**Théorème 5.1.** *Soient  $A$  et  $B$  deux matrices de la famille  $\mathcal{M}$ .*

$$BA \preceq AB \Leftrightarrow \frac{a_1}{a_{1,2}} \oplus \frac{b_2}{b_{1,2}} \preceq \frac{b_1}{b_{1,2}} \oplus \frac{a_2}{a_{1,2}}. \quad (10)$$

**Théorème 5.2.** *Il existe une permutation  $J$  telle que*

$$\forall i < j : \frac{m_1(J(i))}{m_{1,2}(J(i))} \oplus \frac{m_2(J(j))}{m_{1,2}(J(j))} \preceq \frac{m_1(J(j))}{m_{1,2}(J(j))} \oplus \frac{m_2(J(i))}{m_{1,2}(J(i))}.$$

*Et cette permutation est telle que  $M(J) = \min_{\sigma \in S_n} \{M(\sigma)\}$ . Où  $S_n$  est l'ensemble de toutes les permutations (séquences) de  $n$  éléments.*

Ce théorème mène à une généralisation de l'algorithme de Johnson [16] au problème matriciel et il unifie de nombreux résultats éparses de la littérature de ces 50 dernières années [2, 5, 7, 15, 23–26]. Pour cela on appellera  $J$  : *séquence de Johnson*. Il est à noter que dans ce cas, on minimise non seulement le critère  $\Gamma$  mais également le produit matriciel.

### 5.2. CAS GÉNÉRAL

#### 5.2.1. Notations

L'idée est d'extraire des sous-problèmes à deux machines du problème général. Pour cela, à toute matrice  $A$  sont associées les matrices  $A^{(u,v)}$  ( $1 \leq u < v \leq m$ ) où toutes les composantes de la matrice  $A$  sont mises à  $\mathbf{0}$  exceptées les quatre (trois en pratique) dont les indices de ligne ou de colonne sont  $u$  ou  $v$ .

$$A^{(u,v)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a_{u,u} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a_{u,v} & \mathbf{0} & a_{v,v} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (11)$$

ou plus formellement

$$\forall (\ell, c) \in \{1, \dots, m\}^2 \begin{cases} \text{si } (\ell, c) \in \{u, v\}^2 \text{ alors } [A^{(u,v)}]_{\ell,c} = [A]_{\ell,c} \\ \text{sinon } [A^{(u,v)}]_{\ell,c} = \mathbf{0}. \end{cases} \quad (11.a)$$

Pour toute matrice  $M(k)$  de la famille  $\mathcal{M}$ , on note  $M^{(u,v)}(k)$  la matrice  $(M(k))^{(u,v)}$  et étant donnée une séquence  $\sigma$ ,  $M^{(u,v)}(\sigma)$  représente par définition le produit  $\bigotimes_{i=n}^1 M^{(u,v)}(\sigma(i))$  (il est à noter qu'a priori  $M^{(u,v)}(\sigma) \neq (M(\sigma))^{(u,v)}$ ).

5.2.2. Résultats centraux

**Lemme 5.1.**  $\forall \sigma$  séquence,  $M(\sigma) \succeq M^{(u,v)}(\sigma)$ .

*Démonstration.* Toute matrice  $M(k)$  est supérieure ou égale à la matrice extraite  $M^{(u,v)}(k)$  et le produit matriciel est compatible avec la relation d'ordre.  $\square$

**Définition 5.2.** Étant donnée une séquence  $\sigma$ , on note  $J_{(u,v)}$  une séquence de Johnson associée à la famille de matrices  $\mathcal{M}^{(u,v)} = (M^{(u,v)}(k))_{(1 \leq k \leq n)}$ .

Cette séquence de Johnson existe bien car le produit des matrices de  $\mathcal{M}^{(u,v)}$  est équivalent à un produit de matrices carrées triangulaires inférieures d'ordre 2, s'il en existe plusieurs, on en prend une au hasard.

**Propriété 5.3** (borne inférieure).  $\forall \sigma \in \mathcal{S}_n$ ,  $M(\sigma) \succeq M^{(u,v)}(J_{(u,v)})$ .

*Démonstration.*  $J_{(u,v)}$  est la séquence qui minimise le produit des matrices de  $\mathcal{M}^{(u,v)}$ , ce qui implique que  $M^{(u,v)}(J_{(u,v)}) \preceq M^{(u,v)}(\sigma) \preceq M(\sigma)$ .  $\square$

Cette approche privilégie fortement les deux machines  $u$  et  $v$ , elle prend légèrement en compte les machines de  $u + 1$  à  $v - 1$  au travers des termes  $m_{u,v}(k)$  qui traduisent des temps de traitements de la machine  $u$  à la machine  $v$  mais elle ignore entièrement les machines de  $1$  à  $u - 1$  et de  $v + 1$  à  $m$ . Il est partiellement possible de corriger ce travers.

**Définition 5.4.** Soient les matrices :

- $Q^{(u,v)}(k) = M(k)/M^{(u,v)}(k)$  (la matrice résiduelle à droite) ;
- $R^{(u,v)}(k) = M^{(u,v)}(k) \setminus M(k)$  (la matrice résiduelle à gauche) ;
- $\hat{Q}^{(u,v)} = \bigwedge_{1 \leq k \leq n} Q^{(u,v)}(k)$  et
- $\hat{R}^{(u,v)} = \bigwedge_{1 \leq k \leq n} R^{(u,v)}(k)$ .

Le minimum est utilisé ici comme une loi interne, au même titre que la loi  $\oplus$ .  $\hat{Q}^{(u,v)}$  n'est donc pas la plus petite des matrices  $Q^{(u,v)}(k)$ , il n'y a d'ailleurs aucune assurance qu'une telle matrice existe, mais la matrice dont la composante  $\hat{q}_{c,\ell}^{(u,v)}(k)$  est égale à la plus petite des composantes  $q_{c,\ell}^{(u,v)}(k)$ .

**Théorème 5.3** (généralisation des bornes de Lageweg et al.).

$$\forall \sigma \in \mathcal{S}_n, M(\sigma) \succeq \hat{Q}^{(u,v)} M^{(u,v)}(J_{(u,v)}) \hat{R}^{(u,v)}.$$

*Démonstration.*

$$M(\sigma) = \bigotimes_{k=n}^1 M(\sigma(k))$$

or  $\forall k \in \{1 \dots n\}$ ,  $M(\sigma(k)) \succeq M^{(u,v)}(\sigma(k))$  et, par définition de la résiduation,  $M(\sigma(1)) \succeq M^{(u,v)}(\sigma(1))R^{(u,v)}(\sigma(1))$  et  $M(\sigma(n)) \succeq Q^{(u,v)}(\sigma(n))M^{(u,v)}(\sigma(n))$ , donc

$$M(\sigma) \succeq Q^{(u,v)}(\sigma(n)) \left( \bigotimes_{k=n}^1 M^{(u,v)}(\sigma(k)) \right) R^{(u,v)}(\sigma(1))$$

mais par définition,  $Q^{(u,v)}(\sigma(n)) \succeq \hat{Q}^{(u,v)}$ ,  $R^{(u,v)}(\sigma(1)) \succeq \hat{R}^{(u,v)}$  et le produit entre parenthèses est égal à  $M^{(u,v)}(\sigma)$ , donc

$$M(\sigma) \succeq \hat{Q}^{(u,v)} M^{(u,v)}(\sigma) \hat{R}^{(u,v)}.$$

Enfin, par définition d'une séquence de Johnson,  $M^{(u,v)}(\sigma) \succeq M^{(u,v)}(J_{(u,v)})$ , d'où la minoration

$$M(\sigma) \succeq \hat{Q}^{(u,v)} M^{(u,v)}(J_{(u,v)}) \hat{R}^{(u,v)}. \quad \square$$

### 5.2.3. Mise en application

Les formules de calcul des deux matrices résiduées sont données par les deux propriétés suivantes.

**Propriété 5.5** (composantes de la matrice R).

$$\begin{aligned} \text{Si } \ell \notin \{u, v\} \quad & r_{c,\ell}^{(u,v)}(k) = \infty. \\ \text{Si } \ell = u \quad & r_{c,u}^{(u,v)}(k) = \frac{m_{c,u}(k)}{m_{u,u}(k)} \wedge \frac{m_{c,v}(k)}{m_{u,v}(k)} \\ & \text{en particulier } r_{c,u}^{(u,v)}(k) = \mathbf{0} \text{ si } c > u \text{ et } r_{u,u}^{(u,v)}(k) = \mathbf{1}. \\ \text{Si } \ell = v \quad & r_{c,v}^{(u,v)}(k) = \frac{m_{c,v}(k)}{m_{v,v}(k)} \\ & \text{en particulier } r_{c,v}^{(u,v)}(k) = \mathbf{0} \text{ si } c > v \text{ et } r_{v,v}^{(u,v)}(k) = \mathbf{1}. \end{aligned}$$

*Démonstration.* D'après le théorème 1.2 :

$$\left[ R^{(u,v)}(k) \right]_{\ell,c} = \bigwedge_{t=1}^m \left[ M^{(u,v)}(k) \right]_{t,\ell} \setminus [M(k)]_{t,c} = \bigwedge_{t=1}^m m_{\ell,t}^{(u,v)}(k) \setminus m_{c,t}(k).$$

Or si  $\ell, t \notin \{u, v\}$  alors  $m_{\ell,t}^{(u,v)}(k) = \mathbf{0}$  donc  $m_{\ell,t}^{(u,v)}(k) \setminus m_{c,t}(k) = \infty$ . Donc

$$\begin{aligned}
\ell \notin \{u, v\} &\Rightarrow r_{c,\ell}^{(u,v)}(k) = \bigwedge_{t=1}^m \mathbf{0} \setminus m_{c,t}(k) = \infty \\
\ell \in \{u, v\} &\Rightarrow r_{c,\ell}^{(u,v)}(k) = \bigwedge_{t \in \{u,v\}} m_{\ell,t}^{(u,v)}(k) \setminus m_{c,t}(k) \wedge \infty \\
&= m_{\ell,u}^{(u,v)}(k) \setminus m_{c,u}(k) \wedge m_{\ell,v}^{(u,v)}(k) \setminus m_{c,v}(k) \\
&= m_{\ell,u}(k) \setminus m_{c,u}(k) \wedge m_{\ell,v}(k) \setminus m_{c,v}(k) \\
\ell = u &\Rightarrow r_{c,\ell}^{(u,v)}(k) = m_{u,u}(k) \setminus m_{c,u}(k) \wedge m_{u,v}(k) \setminus m_{c,v}(k) \\
&= \frac{m_{c,u}(k)}{m_{u,u}(k)} \wedge \frac{m_{c,v}(k)}{m_{u,v}(k)} \\
\ell = v &\Rightarrow r_{c,\ell}^{(u,v)}(k) = m_{v,u}(k) \setminus m_{c,u}(k) \wedge m_{v,v}(k) \setminus m_{c,v}(k) \\
&= \infty \wedge m_{v,v}(k) \setminus m_{c,v}(k) \\
&= \frac{m_{c,v}(k)}{m_{v,v}(k)}, \tag{12}
\end{aligned}$$

ceci en supposant que les termes aux dénominateurs sont non nuls ( $\neq \mathbf{0}$ ), ce qui est légitime.  $\square$

**Propriété 5.6** (composantes de la matrice Q).

$$\begin{aligned}
\text{Si } c \notin \{u, v\} & \quad q_{c,\ell}^{(u,v)}(k) = \infty. \\
\text{Si } c=u & \quad q_{u,\ell}^{(u,v)}(k) = \frac{m_{u,\ell}(k)}{m_{u,u}(k)} \\
& \quad \text{en particulier } q_{u,\ell}^{(u,v)}(k) = \mathbf{0} \text{ si } \ell < u \text{ et } q_{u,u}^{(u,v)}(k) = \mathbf{1}. \\
\text{Si } c=v & \quad q_{v,\ell}^{(u,v)}(k) = \frac{m_{u,\ell}(k)}{m_{u,v}(k)} \wedge \frac{m_{v,\ell}(k)}{m_{v,v}(k)} \\
& \quad \text{en particulier } q_{v,\ell}^{(u,v)}(k) = \mathbf{0} \text{ si } \ell < v \text{ et } q_{v,v}^{(u,v)}(k) = \mathbf{1}.
\end{aligned}$$

*Démonstration.* En appliquant de nouveau le théorème 1.2 :

$$\left[ Q^{(u,v)}(k) \right]_{\ell,c} = \bigwedge_{t=1}^m [M(k)]_{\ell,t} / \left[ M^{(u,v)}(k) \right]_{c,t} = \bigwedge_{t=1}^m m_{t,\ell}(k) / m_{t,c}^{(u,v)}(k).$$

Or si  $c, t \notin \{u, v\}$  alors  $m_{t,c}^{(u,v)}(k) = \mathbf{0}$  donc  $m_{t,\ell}(k)/m_{t,c}^{(u,v)}(k) = \infty$ . Donc

$$\begin{aligned} c \notin \{u, v\} &\Rightarrow q_{c,\ell}^{(u,v)}(k) = \bigwedge_{t=1}^m m_{t,\ell}(k)/\mathbf{0} = \infty \\ c \in \{u, v\} &\Rightarrow q_{c,\ell}^{(u,v)}(k) = \bigwedge_{t \in \{u,v\}} m_{t,\ell}(k)/m_{t,c}^{(u,v)}(k) \wedge \infty \\ &= m_{u,\ell}(k)/m_{u,c}^{(u,v)}(k) \wedge m_{v,\ell}(k)/m_{v,c}^{(u,v)}(k) \\ &= m_{u,\ell}(k)/m_{u,c}(k) \wedge m_{v,\ell}(k)/m_{v,c}(k) \\ c = u &\Rightarrow q_{c,\ell}^{(u,v)}(k) = m_{u,\ell}(k)/m_{u,u}^{(u,v)}(k) \wedge m_{v,\ell}(k)/\mathbf{0} \\ &= m_{u,\ell}(k)/m_{u,u}(k) = \frac{m_{u,\ell}(k)}{m_{u,u}(k)} \\ c = v &\Rightarrow q_{c,\ell}^{(u,v)}(k) = m_{u,\ell}(k)/m_{u,v}(k) \wedge m_{v,\ell}(k)/m_{v,v}(k) \\ &= \frac{m_{u,\ell}(k)}{m_{u,v}(k)} \wedge \frac{m_{v,\ell}(k)}{m_{v,v}(k)}, \end{aligned}$$

en supposant de nouveau que les termes aux dénominateurs sont non nuls ( $\neq \mathbf{0}$ ). □

En développant le produit matriciel on obtient les relations suivantes.

**Corollaire 5.7.**

$$\begin{aligned} \forall \sigma \text{ séquence, } \forall 1 \leq c \leq \ell \leq m, \quad m_{c,\ell}(\sigma) &\succeq \hat{r}_{c,u}^{(u,v)} m_{u,v}^{(u,v)}(J_{(u,v)}) \hat{q}_{v,\ell}^{(u,v)} . \\ \forall \sigma \text{ séquence, } \Gamma(\sigma) &\succeq \bigoplus_{1 \leq c \leq \ell \leq m} \bigoplus_{1 \leq u \leq v \leq m} \hat{r}_{c,u}^{(u,v)} m_{u,v}^{(u,v)}(J_{(u,v)}) \hat{q}_{v,\ell}^{(u,v)} . \end{aligned}$$

Ce dernier corollaire fournit une minoration du critère.

5.3. APPLICATION AU FLOWSHOP DE PERMUTATION

Dans un flowshop composé de  $m$  machines, à chaque travail  $x$  de l'ensemble des travaux  $\mathcal{T}$  est associée une matrice  $T(x)$ , on lui associe de plus les matrices résiduées  $R^{(u,v)}(x)$  et  $Q^{(u,v)}(x)$  et on définit  $\hat{R}^{(u,v)} = \bigwedge_{x \in \mathcal{T}} R^{(u,v)}(x)$  et  $\hat{Q}^{(u,v)} = \bigwedge_{x \in \mathcal{T}} Q^{(u,v)}(x)$ . De même, à chaque séquence  $\sigma$  sont associées, en plus de la matrice  $T(\sigma)$  définie au paragraphe 4.2, les matrices extraites  $T^{(u,v)}(\sigma)$ . La seconde minoration du corollaire 5.7 se réécrit

$$\forall \sigma \text{ séquence, } C_{max}(\sigma) \succeq \bigoplus_{1 \leq c \leq \ell \leq m} \bigoplus_{1 \leq u \leq v \leq m} \hat{r}_{c,u}^{(u,v)} t_{u,v}^{(u,v)}(J_{(u,v)}) \hat{q}_{v,\ell}^{(u,v)} . \quad (13)$$

Dans le cas d'un flowshop sans contrainte, on montre que  $C_{max}(\sigma) = t_{1,m}(\sigma)$ , ce qui permet de simplifier l'expression (13) ainsi :

$$\forall \sigma \text{ séquence, } C_{max}(\sigma) \succeq \bigoplus_{1 \leq u \leq v \leq m} \hat{r}_{1,u}^{(u,v)} t_{u,v}^{(u,v)}(J_{(u,v)}) \hat{q}_{v,m}^{(u,v)}. \tag{14}$$

On vérifie que

$$r_{1,u}^{(u,v)}(x) = \frac{t_{1,u}(x)}{t_{u,u}(x)} \wedge \frac{t_{1,v}(x)}{t_{u,v}(x)} = \bigotimes_{t=1}^{u-1} p_{x,t} \text{ et que } \hat{r}_{1,u}^{(u,v)}(x) = \bigwedge_{x \in \mathcal{T}} \bigotimes_{t=1}^{u-1} p_{x,t}.$$

Ce dernier terme est noté  $r_u$  dans l'article de Lageweg, Lenstra et Rinnooy Kan [17]. On vérifie également que

$$q_{v,m}^{(u,v)}(x) = \frac{t_{u,m}(x)}{t_{u,v}(x)} \wedge \frac{t_{v,m}(x)}{t_{v,v}(x)} = \bigotimes_{t=v+1}^m p_{x,t} \text{ et que } \hat{q}_{v,m}^{(u,v)} = \bigwedge_{x \in \mathcal{T}} \bigotimes_{t=v+1}^m p_{x,t}.$$

On retrouve cette fois ce qui est noté  $q_v$  dans [17]. Enfin, extraire les matrices  $T^{(u,v)}$  revient à construire un flowshop à deux machines  $M_u$  et  $M_v$  et à interpréter la somme des durées opératoires sur les machines intermédiaires comme un décalage ou, ce qui est équivalent, à considérer ces machines intermédiaires comme non goulet [18]. La composante  $t_{u,v}^{(u,v)}(J_{(u,v)})$  représente alors le  $C_{max}$  optimal de ce flowshop à deux machines. La minoration (14) correspond dans ce cas aux bornes inférieures développées par Lageweg *et al.*

5.3.1. *Propriétés complémentaires*

Les propriétés suivantes ont été séparées du reste car elles nécessitent de préciser, et donc d'alourdir, encore les notations. Elles s'appuient sur le fait que le théorème 5.3 est valable pour toute séquence  $\sigma$  de matrices et pas uniquement pour une séquence complète. Dans ce cas,  $J_{(u,v)}$ , qu'il faudrait noter  $J_{(u,v)}(\sigma)$ , est la séquence de Johnson des matrices de la sous-famille  $(M^{(u,v)}(\sigma(k)))_{(1 \leq k \leq |\sigma|)}$ , et

$\hat{R}^{(u,v)}$  et  $\hat{Q}^{(u,v)}$  doivent être redéfinies ainsi :

- $\hat{Q}^{(u,v)}(\sigma) = \bigwedge_{1 \leq k \leq |\sigma|} Q^{(u,v)}(\sigma(k))$  et
- $\hat{R}^{(u,v)}(\sigma) = \bigwedge_{1 \leq k \leq |\sigma|} R^{(u,v)}(\sigma(k)).$

Le théorème 5.3 devient alors

**Théorème 5.4** (généralisation des bornes de Lageweg *et al.*).

$$\forall \sigma \text{ séquence, } M(\sigma) \succeq \hat{Q}^{(u,v)}(\sigma) M^{(u,v)}(J_{(u,v)}(\sigma)) \hat{R}^{(u,v)}(\sigma).$$

**Corollaire 5.8.**  $\forall \sigma_1, \sigma_2, \sigma_3$  séquences,

$$M(\sigma_1 \sigma_2 \sigma_3) \succeq M(\sigma_3) \hat{Q}^{(u,v)} (\sigma_2) M^{(u,v)}(J_{(u,v)}(\sigma_2)) \hat{R}^{(u,v)} (\sigma_2) M(\sigma_1).$$

Si on connaît le début ou la fin d'une séquence, ce corollaire sert à estimer son  $C_{\max}$ .

### CONCLUSION

Nous avons obtenu au travers de l'algèbre Max-Plus une modélisation matricielle de problèmes d'ordonnancement de type flowshop. Cela a mené à définir un nouveau problème d'ordonnancement purement algébrique. Une famille de bornes inférieures a été construite et étudiée qui généralise la famille de bornes proposée par Lageweg, Lenstra et Rinnooy Kan pour le flowshop sans contrainte au problème matriciel et donc en particulier au problème  $Fm/prmu, a_{ik}, s_{nsd}, r_{nsd}, batch/C_{\max}$ . Les contraintes sont prises en compte de façon transparente dans le processus : elles apparaissent dans les données, mais pas dans les formules de calcul. Ceci constitue un avantage important de cette approche car cela permet d'unifier l'étude de plusieurs problèmes de flowshop différents.

Le processus d'unification peut se poursuivre, il est possible de définir des bornes supérieures et des méthodes de résolution pour le problème matriciel et donc communes aux divers problèmes de flowshop [21] et d'inclure ces bornes dans une procédure par séparation et évaluation [18, 22]. La modélisation matricielle perdure si l'on introduit des dates de début au plus tôt et des durées de latence et une modélisation plus complète se met en place si l'on considère des décalages interopération maximaux en sus des décalages minimaux. Des travaux futurs consisteront à définir d'autres bornes dans ce dernier cas, ou à étendre des bornes déjà existantes.

### RÉFÉRENCES

- [1] F. Baccelli, G. Cohen, G. Olsder and J.P. Quadrat, *Synchronisation and Linearity : An Algebra for Discrete Event Systems*. John Wiley and Sons, New York (1992).
- [2] K.R. Baker, Scheduling groups of jobs in the two machine flow shop. *Math. Comput. Modeling* **13** (1990) 29-36.
- [3] T.S. Blyth, Matrices over ordered algebraic structures. *J. Lond. Math. Soc.* **39** (1964) 427-432.
- [4] T.S. Blyth and M.F. Janowitz, *Residuation Theory*. Pergamon Press (1972).
- [5] F.C. Cetinkaya, Lot streaming in a two stage flow shop with setup, processing and removal times separated. *J. Oper. Res. Soc.* **45** (1994) 1445-1455.
- [6] G. Cohen, D. Dubois, J.P. Quadrat and M. Viot, A linear system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Trans. Autom. Control* **30** (1985) 210-220.
- [7] Y. Dar-Li and C. Maw-Sheng, Two-machine flowshop group scheduling problem. *Comput. Oper. Res.* **27** (2000) 75-985.

- [8] M.R. Garey, D.S. Johnson and R. Sethi, The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* **1** (1976) 117-129.
- [9] S. Gaubert, *Théorie des systèmes linéaires dans les dioïdes*. Thèse de doctorat, École des mines de Paris (1992).
- [10] S. Gaubert and J. Mairesse, Modeling and analysis of timed petri nets using heaps of pieces. *IEEE Trans. Autom. Control* **44** (1999) 683-698.
- [11] B. Giffler, Schedule algebras and their use in formulating general systems simulations, in *Industrial scheduling*. Muth and Thompson, Prentice Hall, New Jersey, USA (1963).
- [12] M. Gondran and M. Minoux, *Graphes et algorithmes*. Eyrolles, Paris, 3<sup>e</sup> edn. (1995).
- [13] J. Gunawardena, *Idempotency*. Publications of the Newton Institute, Cambridge University Press (1998).
- [14] C. Hanen and A. Munier, Cyclic scheduling on parallel processors : An overview, in *Scheduling Theory and its Applications*, edited by P. Chretienne, E. Coffman, J. Lenstra and Z. Liu. John Wiley (1995) 193-226.
- [15] J.R. Jackson, An extension of johnson's results on job-lot scheduling. *Naval Res. Log. Quart.* **3** (1956) 201-203.
- [16] S.M. Johnson, Optimal two- and three-stage production schedules with setup times included. *Naval Res. Log. Quart.* **1** (1954) 61-68.
- [17] B.J. Lageweg, J.K. Lenstra and A.H.G. Rinnooy Kan, A general bounding scheme for the flow-shop problem. *Oper. Res.* **26** (1978) 53-67.
- [18] C. Lenté, *Analyse Max-Plus de problèmes d'ordonnancement de type Flowshop*. Thèse de doctorat, Université François Rabelais, Tours (novembre 2001).
- [19] C. Lenté and J.C. Billaut, Une application des algèbres tropicales aux problèmes d'ordonnancement de type flowshop, in *MOSIM'99*, Annecy, France (octobre 1999) 177-182.
- [20] C. Lenté, J.C. Billaut and J.L. Bouquard, Modélisation unifiée de flowshops à deux machines, in *conférence francophone de modélisation et simulation, (MOSIM'01)*, Troyes, France (avril 2001) 599-603.
- [21] C. Lenté, J.C. Billaut and J.L. Bouquard, Max-plus generalization of a flowshop problem lower and upper bounds, in *(INCOM'01)*, Vienne, Autriche (sept. 2001).
- [22] C. Lenté, F. Chevalyere and N. Neel, Flowshops et minimisation de produits matriciels max+, in *3<sup>e</sup> journées francophone de recherche opérationnelle, (FRANCORO'01), l'aide à la décision pour l'amélioration de la performance*, Québec, Canada (mai 2001) 599-603.
- [23] L.G. Mitten, Sequencing n jobs on two machines with arbitrary time lags. *Manage. Sci.* **5** (1959) 293-298.
- [24] I. Nabeshima and S. Maruyama, A note on the two-machine flow shop scheduling problem with separated setup and cleanup times, time lags and transportation times. *Rep. Univ. Electro-com* **34** (1983).
- [25] D.R. Sule, Sequencing n jobs on two machines with setup, processing and removal times separated. *Naval Res. Log. Quart.* **29** (1982) 517-519.
- [26] T. Yoshida and K. Hitomi, Optimal two-stage production scheduling with setup times separated. *AIIE Transactions* **11** (1979) 261-263.