UNIVERSITÉ PAUL SABATIER

MATIÈRE

- Contrôle et simulation -
COMMANDE DE SYSTÈMES $(max, +)$ - LINÉAIRES

Auteurs :
Lucien RAKOTOMALALA
David TOCAVEN

Encadrants :
Euriell LE CORRONC
Laurent HOUSSIN

22 décembre 2017

Table des matières

Introduction

Chapitre 1

Étude d'un système de transport

Dans cette première partie, nous allons vous présenter nos travaux sur l'étude d'un système de transport. Dans ce problème, il nous est demandé de mettre en œuvre, avec l'algèbre $(max, +)$, une étude d'un système de GET qui représentera notre système temporisé de transport. Nous commencerons par vous présenter le graphe d'évènements correspondant au procédé étudié, nous effectuerons une analyse de ce système pour enfin essayer d'améliorer le système en modifiant le graphe.

1.1 Graphes d'évènements temporisés

Dans un premier temps, nous avons réalisé le graph des événements temporisés (voir figure 1.1). Ce graph représente les différentes gares, un couple de transition $(x\theta 1, x\theta 2)$ représente la gare θ, la première représente "le train rentre en gare", le t_a de la place suivante le temps d'attente en gare, et la seconde transition $(x\theta 2)$ "le train sort de la gare". Les ta des places entre deux entrées ou sorties de gare représentent le temps de trajet sur une ligne de train. Les places $xB12$ et $xB22$ sont des copies de $xB1$ et $xB2$ afin de représenter les contraintes de synchronisation décritent dans l'énoncé. Afin de permettre une mise en représentation qui suivra, nous avons décomposé la place $p0$ et la transition $xA2$ en deux places $p0$ et $p18$ et deux transitions $xA2$ et $xA22$. Nous avons réparti les jetons sur les deux places. Cela nous permettra de décrire les places suivantes avec un état de plus et mais uniquement par rapport à un événement précédent ("$x(k) = Yx(k-1)$") et non deux ("$x(k) = Yx(k-2)$"). Voici la nouvelle représentation d'état correspondant à cette contrainte, figure 1.2.

1.2 Analyse, matrice d'évolution et valeurs propres

Nous abordons ici, dans l'ordre, la mise en représentation d'état du réseau de transport, l'analyse de réductibilité et le calcul des valeurs et vecteurs propres.

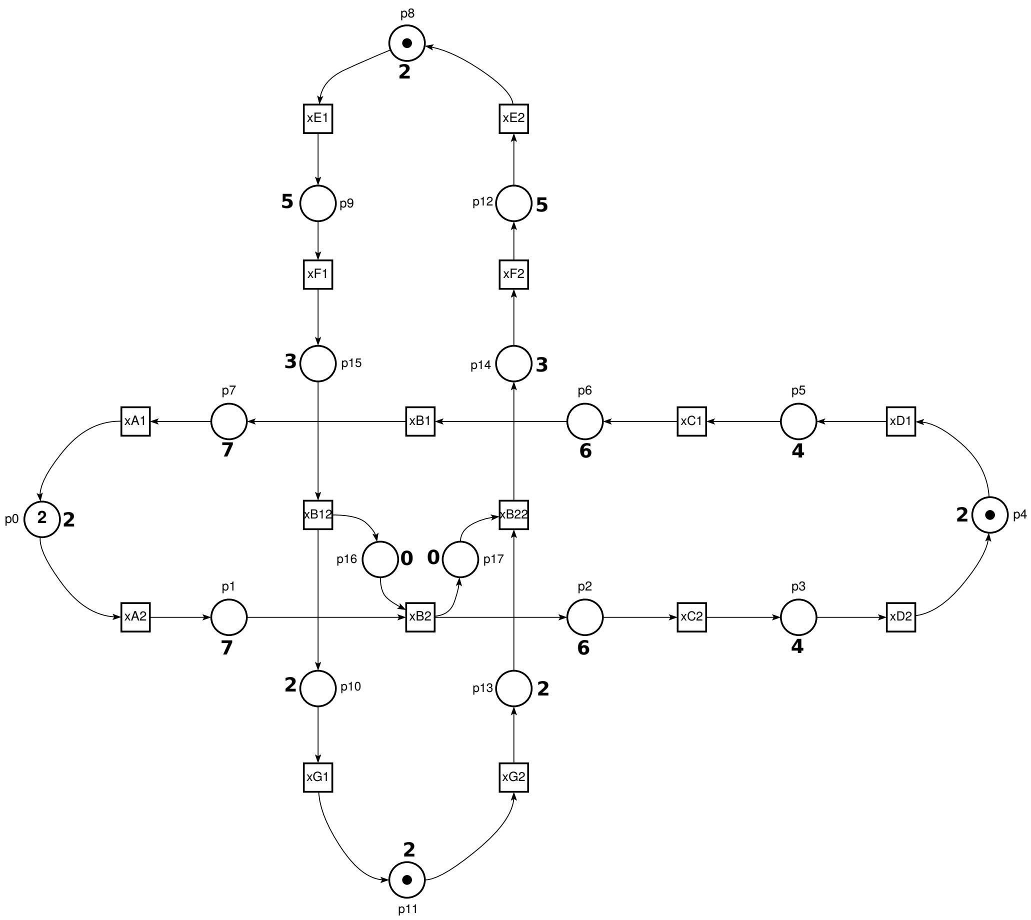

FIGURE 1.1 – Graphe des Événements Temporisé du réseau de transport

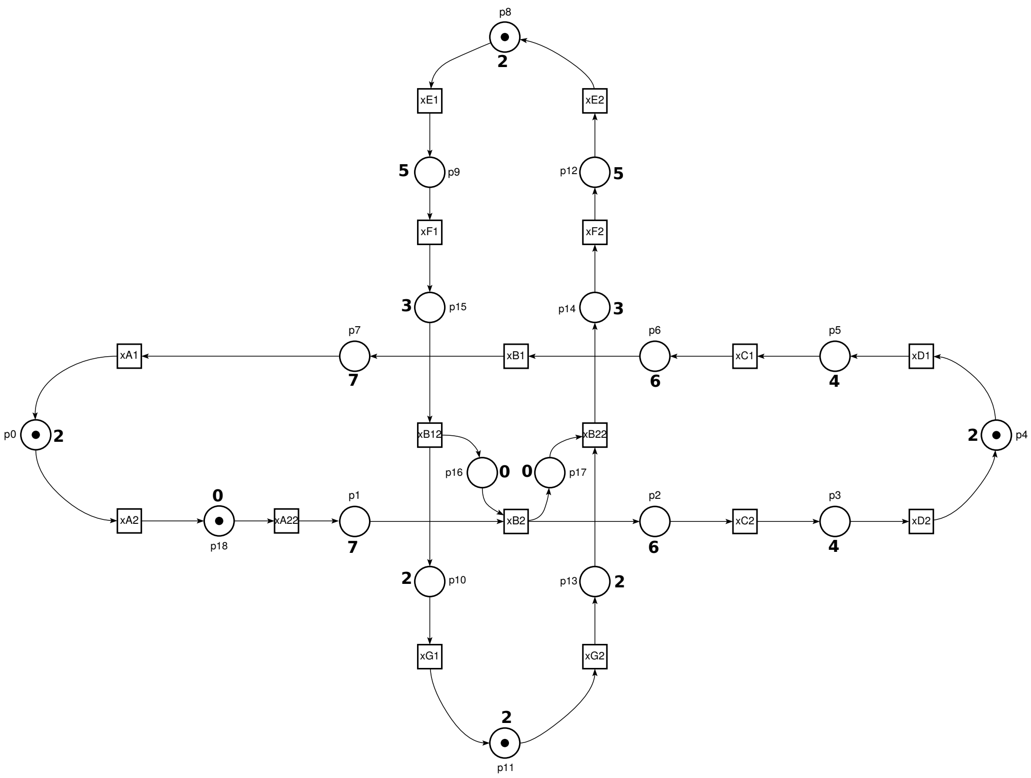

FIGURE 1.2 – Graphe des Événements Temporisé décomposé du réseau de transport

1.2.1 Représentation d'état

Afin de réaliser la mise en représentation d'état, nous avons extrait les égalisés suivantes :

$$\left\{\begin{array}{lcl}
x_{A1}(k) & = & 7x_{B1}(k) \\
x_{B1}(k) & = & 6x_{C1}(k) \\
x_{D1}(k) & = & 2x_{D2}(k-1) \\
x_{D2}(k) & = & 4x_{C2}(k) \\
x_{C2}(k) & = & 6x_{B1}(k) \\
x_{B2}(k) & = & 7x_{A2}(k) \oplus x_{B12} \\
x_{A2}(k) & = & 2x_{A1}(k-2) \\
x_{A22}(k) & = & x_{A1}(k-1) \\
x_{E1}(k) & = & 2x_{E2}(k-1) \\
x_{F1}(k) & = & 5x_{E1}(k) \\
x_{B12}(k) & = & 3x_{F1}(k) \\
x_{G1}(k) & = & 2x_{B12}(k) \\
x_{G2}(k) & = & 2x_{G1}(k-1) \\
x_{B22}(k) & = & 2x_{G2}(k) \oplus x_{B2}(k) \\
x_{F2}(k) & = & 3x_{B22}(k) \\
x_{E1}(k) & = & 5x_{F2}(k)
\end{array}\right.$$

On peut remarquer que notre système est autonome, c'est-à-dire qu'il ne présente pas d'entrée ou de sortie dans l'espace d'état suivant, les matrices B et C ne sont donc pas nécessaires.

$$\left\{\begin{array}{lcl}
x(k) & = & A_0 \cdot x(k) + A_1 \cdot x(k-1) + Bu(k) \\
y(k) & = & C \cdot x(k)
\end{array}\right. \Leftrightarrow \left\{\begin{array}{lcl} x(k) & = & A \cdot x(k-1) \end{array}\right. \tag{1.1}$$

Ces égalités nous ont permis de créer la matrice dynamique suivante :

$$A = \begin{pmatrix}
. & . & . & . & 19 & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & 12 & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & 6 & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & 2 & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & 19 & . & . & . & . & . & . & . & 20 \\
. & . & . & . & . & . & . & . & 15 & . & . & . & . & . & . & . & 16 \\
. & . & . & . & . & . & . & . & 9 & . & . & . & . & . & . & . & 10 \\
. & . & . & . & . & . & . & . & 2 & . & . & . & . & . & . & . & . \\
0 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 2 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 7 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 10 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 12 \\
. & . & . & . & . & . & . & . & . & . & . & . & 2 & . & . & . & . \\
. & . & . & . & . & . & . & . & 9 & . & . & . & 4 & . & . & . & 8 \\
. & . & . & . & . & . & . & . & 12 & . & . & . & 7 & . & . & . & 11 \\
. & . & . & . & . & . & . & . & 17 & . & . & . & 12 & . & . & . & 18
\end{pmatrix} \tag{1.2}$$

(. signifie ϵ)

1.2.2 Irréductibilité

Maintenant que nous avons une représentation d'état de notre procédé, nous allons étudier s'il est irréductible ou non, si c'est le cas, cela signifie qu'il a une unique valeur propre (c'est une implication et non une équivalence). Une des méthodes de vérification de la propriété d'irréductibilité d'une matrice est de vérifier si elle comporte une ligne ou une colonnes d'éléments neutres (ϵ). Nous pouvons voir ici que la matrice A comporte au moins 12 colonnes d'éléments neutres, elle est donc réductible. Une des autres façon est de voir si le graphe des marquages accessible est fortement connexe ou non. Si il n'est, elle est irréductible. Figure 1.3, on peut voir que ce n'est pas le cas les états 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15 et 16 sont des états puits.

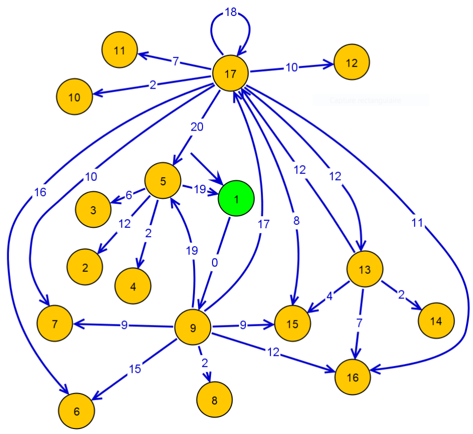

FIGURE 1.3 – Graphe des Marquages accessibles de A

1.2.3 Valeur propre

Nous avons calculé la valeur propre à l'aide de ScicosLab, elle est unique est vaux 18. Cela veut dire que le cycle interne du système est de 18 événements une fois le régime permanent établi.

1.2.4 Vecteur propre

Comme précédemment, nous avons utilité ScicosLab pour calculer le vecteur propre de A.

$$vecteur_{propre} = \begin{bmatrix} 3 & -4 & -10 & -14 & 2 & -2 & -8 & -31 & -15 & -16 & -11 & -8 & -6 & -22 & -10 & -7 & 0 \end{bmatrix}^T \tag{1.3}$$

1.3 Commande des conditions initiales

Chapitre 2

Optimisation d'une ressource dans un problème de jobshop

Cette fois ci, nous allons nous concentrer sur la modélisation d'un problème à base de *jobshop* dans lequel nous étudierons le travail entre deux machines M_1 et M_2 sur deux pièces A et B. Dans un premier temps, nous modéliserons comme dans le chapitre précédent, le GET associé à notre système. Puis nous étudierons chaque marquage initial de notre graphe, selon les ordonnancement possible, afin déterminer le cycle de temps associé de chacun.

2.1 Étude du problème

Le problème de cet exercice peut être décomposé en deux sous problèmes équivalents. Le premier concerne l'ordre dans lequel les pièces A et B peuvent être traités par les machines : A ne peut pas passer sur M_1 tant qu'elle n'est pas d'abord passer sur M_2. Pour la pièce B, elle doit d'abord passer par M_1 puis elle va sur M_2, comme cela est expliqué dans l'ennoncé.

Le deuxième problème qui apparait n'en ai pas vraiment un, il s'agit de choisir l'ordonnancement des usinages de pièces. Très naturellement, pour 2 machines et 2 pièces il vient 2×2 possibilité d'ordonnancement qui sont :

— **Ordonnancement 1** $M_2 \rightarrow A$ puis $M_1 \rightarrow A \rightarrow B$ et $M_2 \rightarrow B$

— **Ordonnancement 2** $M_2 \rightarrow B \rightarrow A$ parallèle à $M1 \rightarrow A \rightarrow B$

— **Ordonnancement 3** $M_2 \rightarrow A \rightarrow B$ parallèle à $M1 \rightarrow B \rightarrow A$

— **Ordonnancement 4** $M_1 \rightarrow B$ puis $M_2 \rightarrow A \rightarrow B$ et $M_1 \rightarrow B$

Pour modéliser notre problème sous forme de GET, nous aurons besoin des évènements qui lient les machines et les pièces entre elles. Nous pouvons obtenir ces informations à partir du tableau donné dans le sujet de TP. Ces temps d'attentes seront utilisés dans les temps d'attentes des places du GET pour modéliser le temps que met une pièce x à être usiné par une machine y.

2.2 Modélisation des GET et représentation d'état et analyse du cycle associé

FIGURE 2.1 – Graphe des Événements Temporisé du *jobshop*

Dans un premier temps, nous avons réalisé le graph des événements temporisés général(voir figure 2.4). Ce graph a 4 transitions (en noir). Elles correspondent, en partant de celle en haut à gauche et dans le sens horaire, à *A est usiné dans M1*, *B est usiné dans M1*, *B est usiné dans M2* et à *A est usiné dans M2*. Les annotations *A*, *B*, *M1* et *M2* ne font pas partis du graph, elles font offices de légendes pour la couleurs des éléments du graph.

2.2.1 Ordonnancement 1

FIGURE 2.2 – Graphe des Événements Temporisé de l'ordonnancement 1 (2.1)

$$\begin{cases} x_{AM_1}(k) &=& 10x_{AM_1}(k-1) \oplus 9x_{BM_2}(k-1) \oplus 7x_{BM_1}(k-1) \\ x_{AM_2}(k) &=& 4x_{AM_1}(k-1) \oplus 3x_{BM_2}(k-1) \\ x_{BM_1}(k) &=& 11x_{BM_1}(k-1) \oplus 14x_{AM_1}(k-1) \oplus 13x_{BM_2}(k-1) \\ x_{BM_2}(k) &=& 20x_{BM_2}(k-1) \oplus 21x_{AM_1}(k-1) \oplus 18x_{BM_1}(k-1) \end{cases}$$

Pour pouvoir utiliser les analyse qui donne une représentation d'état suivante :

$$X(k) = \begin{pmatrix} x_{AM1}(k) \\ x_{AM2}(k) \\ x_{BM1}(k) \\ x_{BM2}(k) \end{pmatrix} = \begin{pmatrix} 10 & \epsilon & 7 & 9 \\ 4 & \epsilon & \epsilon & 3 \\ 14 & \epsilon & 11 & 13 \\ 21 & \epsilon & 18 & 20 \end{pmatrix} X(k-1) = A_1 X(k-1) \tag{2.1}$$

Pour trouver le temps de cycle, nous proposons d'analyser le ou les valeurs propres de A. Nous obtenons avec ScicosLab et la fonction ***karp*** le résultat suivant :

$$\lambda(A_1) = 20$$

OBSERVATION NECESSAIRE!!!!!!!

2.2.2 Ordonnancement 2

L'ordonnancement 2 n'est pas réalisable.

2.2.3 Ordonnancement 3

FIGURE 2.3 – Graphe des Événements Temporisé de l'ordonnancement 3 (2.1)

$$\begin{cases} x_{AM_1}(k) &=& 10x_{AM_1}(k-1) \oplus 9x_{BM_2}(k-1) \\ x_{AM_2}(k) &=& 4x_{AM_1}(k-1) \oplus 3x_{BM_2}(k-1) \\ x_{BM_1}(k) &=& 4x_{AM_1}(k-1) \oplus 3x_{BM_2}(k-1) \\ x_{BM_2}(k) &=& 10x_{BM_2}(k-1) \oplus 11x_{AM_1}(k-1) \end{cases}$$

Ce système d'équation du GET peut être représenté sous forme espace d'état avec le même vecteur que d'état que dans 2.1. Nous obtenons :

$$X(k) = \begin{pmatrix} 10 & \epsilon & \epsilon & 9 \\ 4 & \epsilon & \epsilon & 3 \\ 4 & \epsilon & \epsilon & 3 \\ 11 & \epsilon & \epsilon & 10 \end{pmatrix} X(k-1) = A_2 X(k-1) \tag{2.2}$$

Comme pour l'ordonnancement précédent, nous calculons la valeurs propres du système pour le temps de cycle :

$$\lambda(A_2) = 11$$

2.2.4 Ordonnancement 4

FIGURE 2.4 – Graphe des Événements Temporisé de l'ordonnancement 4(2.1)

$$\begin{cases} x_{AM_1}(k) &=& 20x_{AM_1}(k-1) \oplus 19x_{BM_2}(k-1) \oplus 15x_{AM2}(k-1) \\ x_{AM_2}(k) &=& 14x_{AM_1}(k-1) \oplus 13x_{BM_2}(k-1) \oplus 9x_{AM2}(k-1) \\ x_{BM_1}(k) &=& 4x_{AM_1}(k-1) \oplus 3x_{BM_2}(k-1) \\ x_{BM_2}(k) &=& 10x_{BM_2}(k-1) \oplus 11x_{AM_1}(k-1) \oplus 6x_{AM2}(k-1) \end{cases}$$

$$X(k) = \begin{pmatrix} 20 & 15 & \epsilon & 19 \\ 14 & 8 & \epsilon & 13 \\ 4 & \epsilon & \epsilon & 3 \\ 11 & 6 & \epsilon & 10 \end{pmatrix} X(k-1) = A_3 X(k-1) \tag{2.3}$$

Chapitre 3

Un problème de commande

Chapitre 4

Conclusion

Annexes

Annexe 1 - TITRE

Annexe 2 - TITRE