

République Algérienne Démocratique et Populaire  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique  
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes  
de MASTER ACADEMIQUE**  
Spécialité : **Automatique et informatique  
industrielle.**

*Présenté par*  
**Sofiane ABERKANE**  
**Célia OUCHENE**

Mémoire dirigé par Redouene KARA et codirigé par Said AMARI

Thème

**Contribution à la commande sous  
contraintes de temps des RGETC dans  
l'algèbre Max-plus.**

*Mémoire soutenu publiquement le 03 juillet 2018 devant le jury composé de :*

**Mr Saïd DJENOUNE**  
Professeur, ummto, Président

**Mr Redouene KARA**  
Professeur, ummto, Rapporteur

**Melle Karima TEBANI**  
MRB, ummto, Examineur

**Melle Aldjia NAIT ABDSELAM**  
MAA, ummto, Examineur

**Promotion 2017/2018**

# *Remerciements*

*Nous tenons tout d'abord à remercier le bon Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.*

*En second lieu, nous tenons à remercier notre encadreur Monsieur : REDOUANE KARA, ses précieux conseils et son aide durant toute la période du travail ainsi Monsieur AMARI SAID pour l'orientation, la confiance et la patience qui nous a accordé.*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont portés à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.*

# *Dédicace*

*Nulle œuvre n'est exaltante que celle réalisée avec le soutien moral et financier des personnes qui nous sont proches.*

*Mes parents :*

*Ma mère Dahbia, qui a œuvré pour ma réussite, par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.*

*Mon père Mouloud, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.*

*Mes frères et sœurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité, mon frère Mounir et mes sœurs : Nadia, Malika, Karima, Farida, Houria.*

*Mes tantes, oncles, cousins et cousines, neveux et nièces.*

*Mes amis et amies avec j'ai partagé les meilleures moments de à moi cursus : Bilal, Hocine, Fatima, Katia, Mustapha, Ismaïl, Mohammed.*

*Sans oublier mon binôme Celia et toutes sa familles.*

*Et à tous ceux qui ont attribué de près ou de loin pour que ce projet soit possible, je vous dis merci.*

*Sofiane*

# *Dédicace*

*Nulle œuvre n'est exaltante que celle réalisée avec le soutien moral et financier des personnes qui nous sont proches.*

*Mes parents :*

*Ma mère Nadia , qui a œuvré pour ma réussite, par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.*

*Mon père Belkacem, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.*

*Mes frères et sœurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité, mes frères Fayçal, Massinissa, Ridha, Younes. Et mes sœurs : Kahina, Warda, Sonia, Iman.*

*Mes tantes, oncles, cousins et mes cousines*

*Mes amis et amies avec j'ai partagé les meilleures moments de à moi cursus : Katia, Ghiles, Sarah, Kamilia, Samia.*

*Sans oublier mon binôme Sofiane et toutes sa familles.*

*Et à tous ceux qui ont attribué de près ou de loin pour que ce projet soit possible, je vous dis merci.*

*Celia*

# Table des matières

<b>Introduction générale</b>	<b>4</b>
<b>1 Modélisation des systèmes à événements discrets dans les dioïdes</b>	<b>6</b>
1.1 Introduction . . . . .	6
1.2 Algèbre des dioïdes . . . . .	6
1.2.1 Rappels algébrique . . . . .	6
1.2.2 Propriétés des dioïdes . . . . .	8
1.2.2.1 Relation d'ordre dans un dioïde : ( dioïde comme structure ordonnée). . . . .	8
1.2.2.2 Dioïde complet . . . . .	8
1.2.2.3 Sous dioïde . . . . .	8
1.2.2.4 Dioïdes des matrices . . . . .	9
1.2.3 Résolution d'équations Linéaires dans les dioïdes . . . . .	9
1.3 Réseaux de Petri et graphe d'événements temporisés . . . . .	10
1.3.1 Réseaux de Petri (RDP) . . . . .	10
1.3.1.1 Quelques propriétés des RDP . . . . .	12
1.3.2 Graphes d'événements temporisés (GET) . . . . .	13
1.3.2.1 Propriétés des graphes d'événements temporisés . . . . .	13
1.3.2.2 Représentation de la dynamique : . . . . .	14
1.3.3 Exemple 1.9 . . . . .	18
1.3.3.1 Fonction compteur . . . . .	18
1.3.3.2 Fonction dateur . . . . .	19
1.4 Les graphes d'événements temporisés en conflit . . . . .	20
1.4.1 Problème de ressource partagé . . . . .	20
<b>2 Commande de graphes d'événements temporisés dans l'algèbre Max-plus</b>	<b>22</b>
2.1 Introduction . . . . .	22
2.2 La commande des systèmes à événements discrets sous contraintes de temps dans l'algèbre $(\text{Max}, +)$ : [1] [2] . . . . .	22
2.2.1 Contrainte temporelle . . . . .	23

2.2.2	Commande causale . . . . .	24
2.3	Commande temporelle des systèmes Max-Plus linéaires . . . . .	24
2.3.1	Commande monovariable . . . . .	24
2.3.1.1	Une seule contrainte et une seule commande . . . . .	25
2.3.1.2	Plusieurs contraintes et une seule commande . . . . .	27
2.3.2	Commande multivariable . . . . .	31
2.3.2.1	Une seule contrainte plusieurs commandes . . . . .	31
2.3.2.2	Plusieurs contraintes et plusieurs commandes . . . . .	32
<b>3</b>	<b>Modélisation des RGETC dans l'algèbre Max-Plus</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Méthode 1 . . . . .	37
3.2.1	Modélisation de RGETC avec des équations récurrentes . . . . .	37
3.2.2	Équation (Max-Plus) récurrente des RGETC . . . . .	38
3.2.2.1	La Règle du Premier Arrivé Premier Servi (PAPS) . . . . .	41
3.3	Méthode 2 . . . . .	41
3.3.1	Modélisation de RGETC dans l'algèbre Max-plus à base de systèmes à commutation . . . . .	41
3.3.2	Modélisation de RGETC . . . . .	45
3.4	Le comportement de RGETC comme un automate . . . . .	46
3.4.1	Exemple 3.2 . . . . .	47
<b>4</b>	<b>Proposition d'une loi de Commande des RGETC dans l'algèbre Max- plus</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	Représentation linéaire du GET dans l'algèbre (Max,+) . . . . .	49
4.2.1	Synthèse de la commande en présence de contraintes temporelles . . . . .	50
4.2.1.1	Formalisation de la commande . . . . .	50
4.2.1.2	Commande par retour d'état . . . . .	51
4.2.2	Commande de RGETC qui partage une place en conflit sauf . . . . .	51
4.3	Application à la gestion d'un croisement de lignes ferroviaires . . . . .	53
4.3.1	Modélisation de RGETC . . . . .	55
4.3.1.1	Mode 1 . . . . .	55
4.3.2	Mode 2 . . . . .	57
4.3.3	La Commande de RGETC . . . . .	58
4.3.3.1	Mode 1 ( Franchissement de $GET_1$ et $GET_3$ ) . . . . .	59
4.3.3.2	Mode 2 ( Franchissement du $GET_2$ ) . . . . .	60
	<b>Conclusion générale</b>	<b>63</b>

# Table des figures

1.1	Assemblage d'une paire de bicyclette . . . . .	7
1.2	Représentation de marquage d'un RDP . . . . .	11
1.3	Évolution du marquage . . . . .	11
1.4	Exemple d'un RDP . . . . .	13
1.5	Exemple d'un GET . . . . .	15
1.6	Exemple d'un GET fonction dateur . . . . .	18
1.7	Exemple d'un RGETC . . . . .	20
1.8	Exemple de conflit effective . . . . .	21
2.1	Contrainte temporelle imposé sur une place . . . . .	23
2.2	Exemple d'un GET plusieurs contraintes . . . . .	29
2.3	Exemple d'un GET étendu . . . . .	29
2.4	La loi de commande du GET . . . . .	31
2.5	Graphe d'événements temporisé dans Max-plus . . . . .	35
3.1	Exemple de conflit . . . . .	38
3.2	Un RGETC avec deux GET et deux places de conflit . . . . .	40
3.3	Forward Commutator . . . . .	42
3.4	Backword Commutator . . . . .	42
3.5	Place avec commutation avant et commutation arrière . . . . .	43
3.6	Exemple d'un RGETC . . . . .	43
3.7	Exemple d'un RGETC soumis à des contraintes temporelles . . . . .	45
3.8	Exemple d'un automate . . . . .	47
3.9	Système à trois unités de stocks avec conflit . . . . .	47
3.10	Automate de trois unités de stocks . . . . .	48
4.1	Graphe d'événements témporisé . . . . .	50
4.2	Exemple de conflit . . . . .	51
4.3	Croisement de 3 rails ferroviaires . . . . .	54
4.4	RDP P_ temporisé de Croisement de 3 rails ferroviaires . . . . .	55
4.5	Application de la commande sur RGETC . . . . .	62

# Introduction générale

Les réseaux de Petri ont été introduits par Carl Adam Petri, pour modéliser d'une manière concise les phénomènes de synchronisations, parallélismes et partages de ressources qu'on retrouve dans les systèmes à événements discrets. Ils ont été utilisés avec succès dans des problèmes de commande, d'analyse et appliqués dans différents domaines tel que : les systèmes de production [13] (atelier flexible, lignes d'assemblages,...), les systèmes de transports [14] (routier, ferroviaire,...). Une classe des RDP permet de modéliser des phénomènes de synchronisation et de parallélisme et non le partage de ressource est la classe des GETs. Il est connu que le comportement des GETs peut être exprimé par des équations linéaires classiques dans les dioïdes  $(\text{Max},+)$ ,  $(\text{Min},+)$  [9]. En revanche, toute place d'un GET ne peut être reliée au maximum qu'à une transition en amont et une transition en aval. Les GETs ne permettent donc pas de représenter les phénomènes de concurrences ou de choix multiples qui caractérisent les systèmes avec partage de ressources. Une approche de modélisation de réseau de graphes d'événements temporisés en conflits est proposée dans [8] qui permet d'obtenir des équations linéaires par morceau, le système est divisé en plusieurs GET avec une gestion périodique des jetons de conflits. Dans de nombreuses applications, il est important de prendre en compte explicitement le temps, par exemple, on cherche à évaluer la production par unité de temps d'un atelier manufacturier ou le délai moyen de transmission des informations dans un réseau, le temps est souvent pris en compte à travers la durée des tâches qui sont effectuées. Ces durées sont des contraintes nécessairement satisfaites, ce problème est déjà résolu pour la commande des GETs par certains auteurs [1][2]. Un autre problème à traiter est celui des contraintes imposées sur les réseaux de graphes d'événements temporisés en conflits (RGETC). Notre objectif est de proposer une loi de commande qui satisfait ces contraintes de temps, ce mémoire est structuré en quatre chapitres comme suit :

- Le premier chapitre introduit des définitions et notations de base sur l'algèbre des dioïdes, les réseaux de Petri et les graphes d'événements temporisés ainsi que la représentation de leurs comportements de ces derniers par des équations linéaires dans l'algèbre  $(\text{Max},+)$ ,  $(\text{Min},+)$ . Ainsi ,Nous définissons les réseaux de graphes d'événements temporisés en conflits et quelques notions qui seront utiles par la suite .

- Le second chapitre, met l'accent sur la méthode de AMARI [1] concernant la commande des graphes à événements temporisés sous contraintes de temps dans le dioïde  $(\text{Max}, +)$ .
- Le troisième chapitre introduit deux méthodes proposées à la modélisation des réseaux des graphes d'événements temporisés en conflits [6] [8].
- Le quatrième chapitre est consacré à la proposition d'une méthode de synthèse de lois de commande, en présence de contraintes temporelles dont nous avons proposé une méthode . les résultats obtenus sont appliquées par la suite à un système de gestion de croisement des lignes ferroviaires . nous terminons par une conclusion générale.

# Chapitre 1

## Modélisation des systèmes à événements discrets dans les dioïdes

### 1.1 Introduction

Dans ce chapitre, on introduit quelques définitions et propriétés sur les structures algébriques des dioïdes (Max-plus, Min-plus). Nous avons rassemblé quelques outils de modélisation et d'analyse mathématique, des notions utiles à la compréhension de la théorie de base des systèmes à événements discrets (SED). Ensuite on traite le comportement des graphes d'événements temporisés par des équations linéaires définies dans l'algèbre des dioïdes (Max-plus, Min-plus), et nous représentons quelques définitions de base sur la description des graphes d'événements temporisés (GET). Ensuite nous introduisons un modèle pour les réseaux des graphes d'événements temporisés en conflits.

### 1.2 Algèbre des dioïdes

Le premier objectif dans cette section, est de donner un rappel sur les concepts de base et les propriétés de l'algèbre des dioïdes, qui sont utiles pour la résolution des équations dans cette algèbre.

#### 1.2.1 Rappels algébrique

**Définition 1.1 :** *Un dioïde (ou semi-anneau idempotent) est un ensemble  $D$  muni de deux opérations internes notées  $\oplus$  appelée "addition" et  $\otimes$  appelée "multiplication", tels que, pour tout  $[a, b, c] \in D^3$ .*

- La loi  $\oplus$  est associative :  $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ .
- La loi  $\oplus$  est commutative :  $a \oplus b = b \oplus a$ .
- La loi  $\oplus$  est idempotente :  $a \oplus a = a$ .
- La loi  $\oplus$  admet un élément neutre notée " $\varepsilon$ " et appelé zéro :  $a \oplus \varepsilon = \varepsilon \oplus a = a$ .

- La loi  $\otimes$  est associative :  $(a \otimes b) \otimes c = a \otimes (b \otimes c)$ .
- La loi  $\otimes$  admet un élément neutre notée "e" est appelé identité :  $a \otimes e = e \otimes a = a$ .
- La loi  $\otimes$  est distributive à gauche et à droite par rapport à la loi  $\oplus$  :  $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ ,  $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ .
- L'élément "ε" est absorbent pour la multiplication :  $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$ .

**Exemple 1.1 (Dioïde  $\bar{\mathbb{R}}_{min}$ )**  $\bar{\mathbb{R}}_{min} = (\bar{\mathbb{R}} \cup \{+\infty\}, min, +)$  est un dioïde commutatif, où la loi additive  $\oplus$  correspond à l'opération minimum et la loi multiplicative  $\otimes$  est équivalente à l'addition usuelle. L'élément zéro de  $\bar{\mathbb{R}}_{min}$  est  $\varepsilon = +\infty$  et l'élément identité est  $e = 0$ . Ce dioïde est appelé l'algèbre  $(min, +)$ . ( $2 \oplus 6 = min(2, 6) = 2$ ), et la loi  $\otimes$  est la somme usuelle ( $4 \otimes 2 = 4 + 2 = 6$ ).

### • Application 1.1

On s'intéresse à la production de bicyclettes (1.1), on peut dire qu'une bicyclette est la "somme" d'une paire de roues et d'un cadre. Cette nouvelle "somme" notée  $\oplus$  pour la distinguer du (+) habituel, correspond à l'opération d'assemblage. On notera que l'on se permet alors d'ajouter des grandeurs ne s'exprimant pas dans les mêmes unités. Si on a 2 roues et 1 cadre, on ne peut fabriquer qu'une seule bicyclette, donc  $1 \oplus 2 = 1$ , ce qui est équivalent dans l'algèbre usuelle au  $min(2, 1) = 1$ .

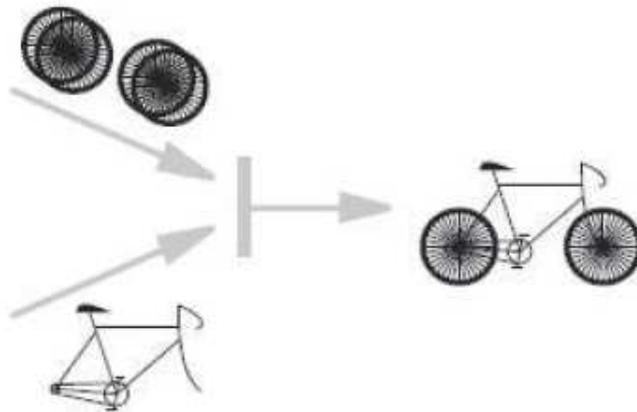


FIGURE 1.1 – Assemblage d'une paire de bicyclette

## 1.2.2 Propriétés des dioïdes

### 1.2.2.1 Relation d'ordre dans un dioïde : ( dioïde comme structure ordonnée).

Dans un dioïde  $(D, \oplus, \otimes)$ , on définit la relation d'ordre  $\preceq$  par :  $a \preceq b \iff a \oplus b = b$ .

Dans  $\bar{\mathbb{R}}_{max}, (\bar{\mathbb{R}} \cup \{-\infty\}, max, +)$  la relation d'ordre coïncide avec l'ordre usuel.  $2 \oplus 1 = 2$  si et seulement si  $2 \succcurlyeq 1$ .

Dans  $\bar{\mathbb{R}}_{min}, (\bar{\mathbb{R}} \cup \{+\infty\}, min, +)$  la relation d'ordre est l'inverse de l'ordre usuel.  $2 \oplus 1 = 1$  si et seulement si  $2 \preceq 1$ .

La relation  $R$  est une relation d'ordre sur un ensemble  $S$  si les trois axiomes suivants sont vérifiés,  $\forall x, y, z \in S$ .

- $R$  est réflexive :  $xRx$ .
- $R$  est transitive si :  $xRy$  et  $yRz$  alors  $xRz$ .
- $R$  est antisymétrique si :  $xRy$  et  $yRx$  alors  $x = y$ .

### 1.2.2.2 Dioïde complet

#### Définition 1.2 : [7]

Un dioïde  $(D, \oplus, \otimes)$  est dit complet si il est fermé par les sommes infinies, et si la loi  $\otimes$  distribue (à gauche et à droite) sur les sommes infinies, c'est-à-dire que si pour tout  $(b \in D)$  et tout sous-ensemble  $(A \subset D)$ .

$$b \otimes \left( \bigoplus_{a \in A} a \right) = \bigoplus_{a \in A} (b \otimes a)$$

$$\left( \bigoplus_{a \in A} a \right) \otimes b = \bigoplus_{a \in A} (a \otimes b)$$

**Exemple 1.2 : (dioïde complet de  $\bar{\mathbb{R}}_{min}$ )** Le dioïde  $(\bar{\mathbb{R}} \cup \{-\infty, +\infty\}, min, +)$ , avec  $\varepsilon = +\infty$ ,  $e = 0$  et  $T = -\infty$  tel que  $(+\infty) + (-\infty) = -\infty$  avec  $T$  est l'élément maximum d'un dioïde, est un dioïde complet qu'on notera  $\bar{\mathbb{R}}_{min}$ .

### 1.2.2.3 Sous dioïde

**Définition 1.3 :** Soit  $(D, \oplus, \otimes)$  un dioïde. Le sous-ensemble  $C \subset D$  est un sous-dioïde  $(C, \oplus, \otimes)$  du dioïde  $(D, \oplus, \otimes)$ , si et seulement si, les éléments neutres de  $D$

appartiennent à  $C$ ,  $(\{e, \varepsilon\} \in C)$  et le sous-ensemble  $C$  est fermé pour les lois  $\oplus$  et  $\otimes$ , c'est-à-dire :

$$\forall a, b \in C, a \oplus b \in C$$

$$a \otimes b \in C.$$

### 1.2.2.4 Dioïdes des matrices

Soit  $(D, \oplus, \otimes)$  un dioïde, l'ensemble des matrices de dimension  $n \times p$  à coefficients dans  $D$  forme un dioïde de matrice notée  $D^{n \times p}$ , dont les lois  $\oplus$  et  $\otimes$  sont définies comme suit  $A, B \in D^{n \times p}$  et  $\forall i, .j = \{1, n\}$ .

$$((A \oplus B)_{ij} = A_{ij} \oplus B_{ij} .$$

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^n (A_{ik} \otimes B_{kj}) = C. \text{ avec } A \in D^{n \times p}, B \in D^{p \times s} \text{ et } C \in D^{n \times s}.$$

$A_{ij}$  désigne l'élément de la  $i^{ime}$  ligne et de la  $j^{ime}$  colonne des matrices A, B respectivement et  $C_{ij}$  désigne la  $i^{ime}$  ligne et de la  $j^{ime}$  colonne de la matrice C.

**Exemple 1.4 :** soit  $A, B$  deux matrices dans le dioïde  $(\min, +)$  tel que :

$$A = \begin{bmatrix} 2 & 5 \\ 3 & 4 \end{bmatrix} \text{ et } B = \begin{bmatrix} e & 2 \\ 1 & 6 \end{bmatrix}$$

On à :

$$A \oplus B = \begin{bmatrix} \min(2, e) & \min(5, 2) \\ \min(3, 1) & \min(4, 6) \end{bmatrix} = \begin{bmatrix} e & 2 \\ 1 & 4 \end{bmatrix}$$

$$A \otimes B = \begin{bmatrix} \min(2 + e, 5 + 1) & \min(2 + 2, 5 + 6) \\ \min(3 + e, 4 + 1) & \min(3 + 2, 4 + 6) \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 3 & 5 \end{bmatrix}.$$

### 1.2.3 Résolution d'équations Linéaires dans les dioïdes

Certaines équations définies dans des dioïdes complets admettent des solutions particulières extrêmes, des recherches qui ont été faites pour bénéficier à la théorie des dioïdes, il permettent notamment la résolution des équations de type point fixe  $x = a \otimes x \oplus b$  qu'on va utiliser par la suite.

**Définition 1.5 (Étoile de Kleene)** Soit  $D$  un dioïde complet et  $a$  un élément de  $D$ , l'étoile de Kleene de  $a$ , notée  $a^*$ , est définie comme suit :  $a^* = \bigoplus_{k \in \mathbb{N}} a^k$  avec  $a^0 = e$ . l'étoile de Kleene d'une matrice carrée  $A \in D^{n \times n}$ , notée  $A^*$ , est définie par  $A^* = \bigoplus_{(i \in \mathbb{N})} A^i$  avec  $A^0 = I_n$  et  $I_n$  désigne la matrice identité ou  $(I_n(i, j) = e \text{ si } i = j) \text{ et } (\varepsilon \text{ si non})$ .

**Théorème 1.1** [15]

La solution minimale (au sens de l'ordre) de l'équation est donnée par

$$X = A^* \otimes B$$

avec  $A \in \mathbb{R}_{max}^{n \times n}$  et  $B \in \mathbb{R}_{max}^{n \times m}$ .

Pour plus de détail, voir [1].

## 1.3 Réseaux de Petri et graphe d'événements temporisés

### 1.3.1 Réseaux de Petri (RDP)

Les réseaux de Petri ont été introduits par Carl Adam Petri. Ils permettent une modélisation graphique et mathématique des systèmes de production. dans les systèmes à événements discrets. Et pour l'étude des (SED) dans l'algèbre des dioïdes, une classe des réseaux de Petri appelée graphes d'événements temporisés (GET) sont utilisées comme outil de modélisation, ces graphes peuvent être représentés par des équations linéaires dans cette algèbre. En effet, on s'intéresse à modéliser le système étudié par un GET puis à partir du graphe obtenu, on établit les équations linéaires dans le dioïde qui permettent l'analyse de ce système.

**Définition 1. 6 (Réseau de Petri )** *Un réseau de Petri est un graphe avec deux types de sommets : les places (représentées par des cercles), et les transitions (représentées par des barres). Un ensemble d'arcs qui relient les places aux transitions et des transitions aux places, le nombre de transitions et de places est fini et non nul. Chaque place peut comporter un ou plusieurs jetons représentés par des points. L'état d'un réseau de Petri est défini par son marquage. Un marquage associé à chaque place est un nombre entier positif, que l'on représente graphiquement par des jetons.*

**Définition formelle :** Un réseau de Petri est un graphe orienté biparti ordinaire défini par un quadruplet  $R = (P, T, \text{Pré}, \text{Post})$ , où :

- $P$  est un ensemble fini de places,  $P_{ij} = \{p_{12}, p_{23}, \dots, p_n\}$ .
- $T$  est un ensemble fini de transitions,  $T = \{t_1, t_2, \dots, t_m\}$ .
- $\text{Pré}$  est une application de  $: P \times T \rightarrow \{0, 1\}$  dite d'incidence avant, notée  $\text{Pré}(p,t)$ .
- .
- $\text{Post}$  est une application de  $: P \times T \rightarrow \{0, 1\}$  dite d'incidence arrière, notée  $\text{Post}(p,t)$ .
- .
- $p_{ij}$  est une place d'entrée de la transition  $t_j$ , si  $\text{Pré}(p_{ij}, t_j) > 0$ .
- $p_{ij}$  est une place de sortie de la transition  $t_j$ , si  $\text{Post}(p_{ij}, t_j) > 0$ .

**Marquage :** Dans un RDP chaque place contient un nombre entiers positifs ou nul des jetons. On appelle marquage du réseau de Petri une distribution donnée de jetons dans les places. Les jetons circulent de place en place le long des arcs orientés en franchissant les transitions.

**Exemple 1.5 :**

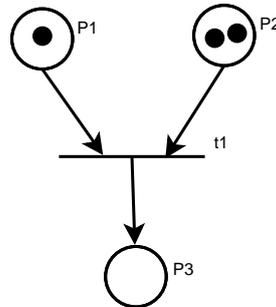


FIGURE 1.2 – Représentation de marquage d’un RDP

Le marquage initiale de ce RDP est :

$$M_0 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

**Dynamique des Réseaux de Petri (Évolution du marquage) [10]**

— **Validation d’une transition**

L’évolution de l’état d’un réseau de Petri correspond à l’évolution de son marquage. Les jetons qui matérialisent l’état du réseau à un instant donné, peuvent passer d’une place à une autre par franchissement ou tir d’une transition.

Une transition est validée (on dit aussi sensibilisée ou franchissable ou encore tirable) si toutes ses places d’entrée contiennent au moins un jeton.

**Exemple 1.6**

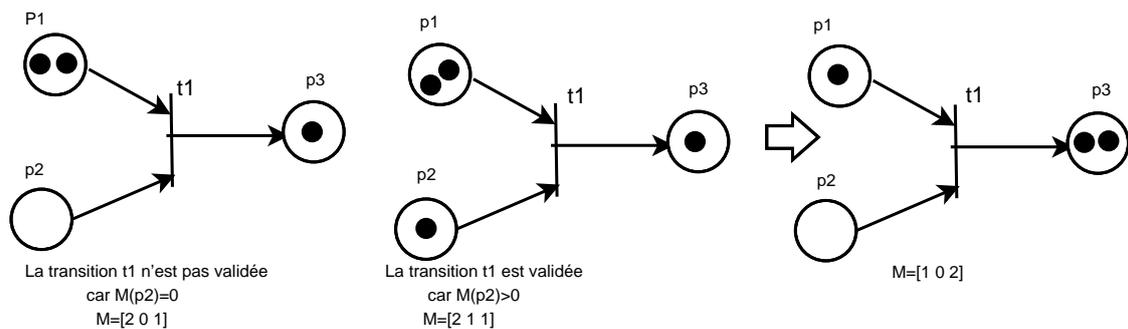


FIGURE 1.3 – Évolution du marquage

**Le franchissement** Le franchissement d'une transition ou le tir d'une transition, consiste à enlever un jeton dans chacune des places d'entrées de la transition et à ajouter un jeton dans chacune des places de sortie de la même transition.

**Séquence de franchissement [10]** Le franchissement successif des transitions (sensibilisées) dans un ordre donné à partir du marquage initiale constitue une séquence de franchissements.

### 1.3.1.1 Quelques propriétés des RDP

Le graphe des marquages associé à un RdP fournit des indications essentielles sur le fonctionnement du système qu'il représente. Suivant que le graphe des marquages est fini ou infini, qu'il présente ou non des circuits il reflète certaines des propriétés caractéristiques du système modélisé.

**Définition 1.7 (vivacité)** Une transition  $t_j$  d'un réseau de Petri marqué est dite vivant, pour un marquage initial  $m_0$ , si et seulement si  $\forall m_k \in R(m_0) \exists S_k H_k \in S_k$  est un marquage  $m_l \in R(m_0)$ ,  $m_k [S_k \succ m_l]$ . Un réseau de Petri marqué est dit vivant si et seulement si, tous ses transitions sont vivantes.

**Définition 1.8 (Blocage)** Un état de blocage, si aucune transition n'est franchissable à partir d'un marquage.

**Définition 1.9 (Bornitude)** Une place  $p_{ij}$  d'un réseau de Petri marqué est bornée, pour un marquage initial  $m_0$ , si et seulement si,  $\forall m_k \in R(m_0)$ ,  $m_{ij}(p_{ij}) \leq C_{ij}$ . Où  $C_{ij}$  est un entier non négative. Un réseau de Petri marqué est borné pour un marquage initial  $m_0$  si et seulement si tous ses places sont bornées.

**Définition 1.10 (Réseau de Petri temporisé)** Un réseau de Petri est dit temporisé si des temporisations sont associées aux places et/ou aux transitions.

**Exemple 1.7** On associe des temporisations  $\tau_i$  à chaque place  $p_{ij}$ , tel que dans notre exemple  $[\tau_1, \tau_2, \tau_3, \tau_4] = [1, 5, 10, 0]$

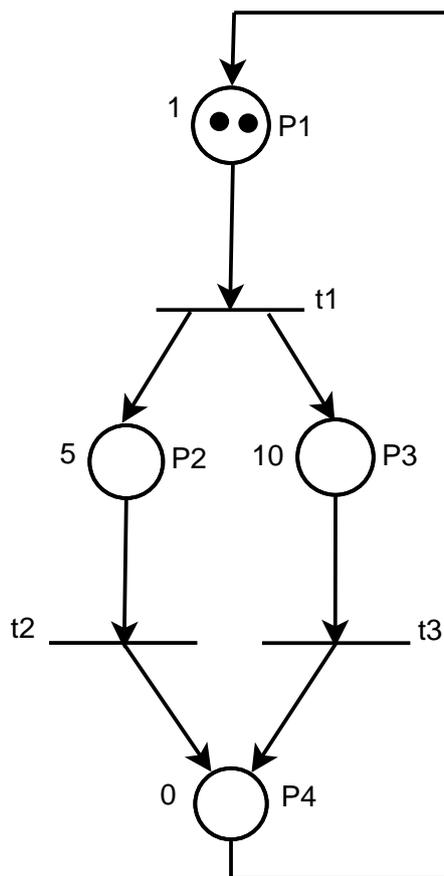


FIGURE 1.4 – Exemple d'un RDP

### 1.3.2 Graphes d'événements temporisés (GET)

On s'intéresse dans notre travail à une sous-classe des RDP dite « Les Graphes d'événements temporisés », qui est souvent utilisée comme outil de modélisation. Le comportement dynamique de ces graphes peut être décrit par un système d'équation Max-plus ou Min-plus linéaire, dans cette sous section, nous représentons quelques définitions et notations sur les GET et sa représentation dans l'algèbre des dioïdes.

**Définition 1.11** *Un graphe d'événements est un réseau de Petri ordinaire où chaque place possède exactement une transition d'entrée et une transition de sortie. Un graphe d'événements est dit temporisé si des temporisations sont associées aux places et/ou transitions.*

#### 1.3.2.1 Propriétés des graphes d'événements temporisés

Nous rappelons brièvement quelques caractéristiques des graphes d'événements temporisés à travers les propositions suivantes :

— **Chemin :**

Un chemin entre deux transitions  $t_j$  et  $t_i$  d'un réseau de Petri, est une succession de places et de transitions relié par des arcs.

— **Circuit :**

Un circuit est un chemin qui commence et se termine au même sommet .

**Proposition 1.1**

Le nombre de jetons d'un circuit est constant dans un graphe d'événements temporisé.

**Démonstration**

Rappelons qu'un circuit est un chemin qui commence et se termine au même sommet. Alors, si une transition franchissable appartenant à un circuit est franchie, son franchissement prend un jeton dans la place en amont de la transition pour le remettre dans la place en aval de la transition.

— **Chemin élémentaire**

Un chemin élémentaire est un chemin ou aucun arc n'est répété.

**Proposition 1.2**

Un graphe d'événements temporisé est vivant si et seulement si, chaque circuit élémentaire contient au moins une place initialement marquée.

**Démonstration**

Supposons qu'aucune place d'un circuit élémentaire d'un GET ne soit initialement marquée (c'est-à-dire que chaque sommet n'est rencontré qu'une seule fois dans le circuit). En référence à la **proposition 1.1**, ce circuit ne contiendra jamais de jeton et donc aucune de ses transitions est franchissables, le GET n'est donc pas vivant.

Inversement, dans un GET non vivant une transition qui n'est jamais franchie possède obligatoirement en amont une transition également jamais franchie. En remontant ainsi d'une transition jamais franchie à une autre située en amont, on aboutit inéluctablement à une transition appartenant à un circuit qui est donc nécessairement sans jeton.

**1.3.2.2 Représentation de la dynamique :**

Afin de faciliter la représentation des éléments du modèles graphiques, tels que les graphes d'événements temporisés qui sont utilisés comme outils de modélisation graphique, Ensuite la description de ces graphes sous forme d'équations mathématiques, c'est à dire le comportement d'un graphe d'événements temporisé peut s'exprimer sous forme des équations linéaires dans les dioïdes  $\overline{\mathbb{R}}_{min}$  ou  $\overline{\mathbb{R}}_{max}$ . On considère  $\tau_i$  la temporisation associée à la place  $p_{ij}$  avec un marquage initial  $m_{ij}$ , la plus grande temporisation du graphe d'événements considéré est notée  $\tau^{max}$  définie par  $\tau^{max} = max(\tau_i)$ . Le plus grand marquage initial des places du graphe d'événements considérés est défini par  $m = max(m_{ij})$ . En générale le comportement d'un graphe d'événements temporisé est représenté sous l'une des deux formes définies dans la suite .

**a) Fonctions compteurs, domaine temporel** Pour chaque transition du graphe d'événements temporisé, on associe une fonction dépendante du temps  $\theta(t) \in \overline{\mathbb{R}}_{min}^n$ , cette fonction est appelée compteur. Elle correspond au nombre de franchissement de la transition jusqu'à l'instant considéré  $t$ . Les compteurs correspondants aux transitions sources [transition de commande  $t_u$ ] sont les composants du vecteur  $u(t) \in \overline{\mathbb{R}}_{min}^m$ .

On considère le graphe d'événements suivants (1.5), dont le comportement est représenté par les équations :

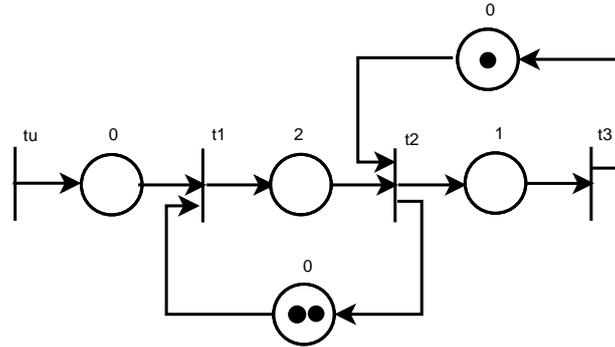


FIGURE 1.5 – Exemple d'un GET

$$\begin{cases}
 \theta_1(t) \leq \min(0 + u(t), 2 + \theta_2(t)) \\
 \theta_2(t) \leq \min(0 + \theta_1(t - 2), 1 + \theta_3(t)) \\
 \theta_3(t) \leq (\theta_2(t - 1))
 \end{cases}$$

Nous utilisons des opérations algébriques de Min-plus ( $\oplus$ ) pour l'opérateur ( $\min$ ) et ( $\cdot$ ) pour la somme classique (+), on réécrit ces équations comme suit :

$$\begin{cases}
 \theta_1(t) \leq e u(t) \oplus 2 \theta_2(t) \\
 \theta_2(t) \leq e \theta_1(t - 2) \oplus 1 \theta_3(t) \\
 \theta_3(t) \leq \theta_2(t - 1)
 \end{cases}$$

On obtient pour ce graphe l'équation matricielle suivante :

$$\theta(t) \leq \begin{bmatrix} \varepsilon & 2 & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(t) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix} \theta(t - 1) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(t - 2) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix} u(t)$$

Cette équation est sous la forme générale :

$$\theta(t) \leq \bigoplus_{\tau=0}^{\tau^{max}} (A_{\tau} \theta(t - \tau) \bigoplus B_{\tau} u(t - \tau)) \quad (1.1)$$

On s'intéresse à l'évolution au plus tôt de ces graphes. Cette évolution correspond à la solution maximale de l'équation (1.1) dans  $\bar{\mathbb{R}}_{min}$ , elle satisfait l'équation linéaire suivante :

$$\theta(t) = \bigoplus_{\tau=0}^{\tau^{max}} (A_{\tau} \theta(t - \tau) \bigoplus B_{\tau} u(t - \tau)) \quad (1.2)$$

où  $A_{\tau} \in \bar{\mathbb{R}}_{min}^{n \times n}$  est la matrice dont le terme  $A_{\tau_{ij}}$  est égal à  $m_{ij}$  qui est le marquage initial de la place  $p_{ij}$  temporisée à  $\tau$ . Si cette place n'existe pas, la composante  $A_{\tau_{ij}}$  est égale à  $\varepsilon$ . Les termes des matrices  $B_{\tau} \in \bar{\mathbb{R}}_{min}^{n \times m}$  correspondent aux marquages initiaux des places en aval des transitions sources.

L'équation (1.2) est implicite, elle est souvent remplacé par sa solution explicite suivante :

$$\theta(t) = \bigoplus_{\tau=1}^{\tau^{max}} (A_0^* A_{\tau} \theta(t - \tau) \bigoplus A_0^* B_{\tau} u(t - \tau)) \quad (1.3)$$

ou :  $A_0^*$  est l'étoile de Kleene de la matrice  $A_0$ .

**Équation d'état dans le dioïde  $\bar{\mathbb{R}}_{min}$**  L'équation explicite (1.3) peut être transformée en une forme d'état, par analogie avec la théorie des systèmes linéaires classiques, nous décomposons toutes les temporisations  $\tau > 1$  en  $m$  place temporisée à 1, nous ajoutons donc  $(\tau - 1)$  transitions intermédiaires, et on associe des fonctions compteurs à ces transitions intermédiaires, et le vecteur d'état résultant, noté  $x(t) \in \bar{\mathbb{R}}_{min}^N$ , avec  $N = n + n'$ ,  $n'$  le nombre de transitions intermédiaires qui sont les composantes du vecteur  $\tilde{\theta}(t) \in \bar{\mathbb{R}}_{min}^{n'}$ , ce vecteur est représenté par  $x(t) = \begin{bmatrix} \theta(t) \\ \tilde{\theta}(t) \end{bmatrix}$ . Le comportement dynamique du GET étendu est décrit par une équation de la forme suivante :

$$x(t) = \hat{A}_0 x(t) \bigoplus \hat{A}_1 x(t - 1) \bigoplus \hat{B}_0 u(t) \quad (1.4)$$

Qui peut s'écrire sous la forme explicite suivante :

$$x(t) = A x(t - 1) \bigoplus B u(t) \quad (1.5)$$

Avec

$$A = \hat{A}_0^* \hat{A}_1 \text{ et } B = \hat{A}_0^* \hat{B}_0$$

**b) Fonctions dateurs, domaine évènementiel :** Dans ce cas on associe à chaque transition une fonction  $\theta(k) \in \bar{\mathbb{R}}_{max}^n$  appelée dateur, qui représente des dates d'occurrences

des événements du GET. Les dateurs correspondants aux transitions sources sont les composantes des dateurs  $u(k) \in \bar{\mathbb{R}}_{max}^m$ , la dynamique d'un graphe d'événements temporisé est représentée par l'équation suivante :

$$\theta(k) \geq \bigoplus_{l=0}^{m^{max}} (A_l \theta(k-l) \bigoplus B_l u(k-l)) \quad (1.6)$$

$A_l \in \bar{\mathbb{R}}_{max}^{n \times n}$  est la matrice dont la composante  $A_{l,ij}$  est égale à  $\tau_{ij}$ , tel que  $\tau_{ij}$  correspond à la temporisation de la place  $p_{ij}$  marquée a  $l$ , si cette place n'existe pas, le terme  $A_{l,ij}$  est égale à  $\varepsilon$ , les termes des matrices  $B_l \in \bar{\mathbb{R}}_{max}^{n \times m}$  correspondant aux temporisations des places de sortie des transitions sources .

Généralement, on s'intéresse à l'évolution au plus tôt des graphes d'événements temporisés, cette évolution correspond à la solution minimale dans  $\bar{\mathbb{R}}_{max}$  de l'inéquation précédente (1.6), cette solution est donnée par l'équation linéaire suivante :

$$\theta(k) = \bigoplus_{l=0}^{m^{max}} (A_l \theta(k-l) \bigoplus B_l u(k-l)) \quad (1.7)$$

cette équation est implicite, et elle est souvent remplacé par sa solution explicite suivante :

$$\theta(k) = \bigoplus_{l=0}^{m^{max}} (A_0^* A_l \theta(k-l) \bigoplus A_0^* B_l u(k-l)) \quad (1.8)$$

ou  $A_0^*$  est l'étoile de kleene de la matrice  $A_0$ .

**Équation d'état dans  $\bar{\mathbb{R}}_{max}$  :** L'équation explicite (1.8) peut être transformée en une forme d'état. Nous décomposons chaque place  $p_{ij}$  en  $m_{ij}$  place marquée à 1. et nous ajoutons  $(m_{ij} - 1)$  transition intermédiaire, on associe des dateurs à ces transitions intermédiaires et le vecteurs  $\theta(k) \in \mathbb{R}_{max}^{n'}$  avec  $n'$  est le nombre des composants du vecteur d'état résultant :  $x(k) \in \bar{\mathbb{R}}_{max}^N$ ,  $N = n + n'$  .

$$x(k) = \begin{bmatrix} \theta(k) \\ \tilde{\theta}(k) \end{bmatrix} \quad (1.9)$$

Dans notre cas, nous considérons des graphes d'événements dont le marquage des place qui ont des transitions sources en amont est nul. Le comportement dynamique du graphe d'événements temporisé étendu est décrit par une équation de la forme :

$$x(k) = \hat{A}_0 x(k) \bigoplus \hat{A}_1 x(k-1) \bigoplus \hat{B}_0 u(k). \quad (1.10)$$

Qui peut s'écrire sous la forme explicite suivante :

$$x(k) = A x(k-1) \oplus B u(k) \quad (1.11)$$

Avec  $A = \hat{A}_0^* \hat{A}_1$  et  $B = \hat{A}_0^* \hat{B}_0$ .

### Propriété 1.1 [1]

Pour un graphe d'événements temporisé avec, l'équation d'état (1.5) est équivalente à la formulation suivante :

$$\mathbf{x}(\mathbf{k}) = \mathbf{A}^\tau x(k-\tau) \oplus \left[ \bigoplus_{k'=0}^{\tau-1} A^{k'} B u(k-k') \right] \quad (1.12)$$

Pour tout entier tel que  $\tau \geq 1$ .

### 1.3.3 Exemple 1.9

Considérons le graphe d'événements temporisé suivant (1.6), constitué de trois transitions internes, une transition source  $tu$  et quatre places temporisées ( $|p| = 4$ ), pour cet exemple nous avons  $\tau^{max} = 1$ ,  $m^{max} = 1$ .

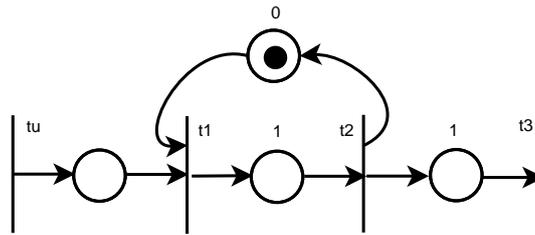


FIGURE 1.6 – Exemple d'un GET fonction dateur

#### 1.3.3.1 Fonction compteur

Une fonction compteur  $\theta_i(t)$  est associée à chaque transition  $t_i$  de ce GET, on considère le franchissement au plus tôt. Les fonctions compteurs vérifient les équations suivantes :

$$\begin{cases} \theta_1(t) = 1 \theta_2(t) \oplus u(t) \\ \theta_2(t) = \theta_1(t-1) \\ \theta_3(t) = \theta_2(t-1) \end{cases}$$

L'équation (1.2) donne

$$\theta(t) = \begin{bmatrix} \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(t) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon \end{bmatrix} \theta(t-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix} u(t)$$

L'équation (1.3) donne

$$\theta(t) = \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(t-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix} u(t)$$

Nous obtenons alors l'équation d'état (1.5) suivante

$$x(t) = \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} x(t-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix} u(t)$$

avec  $x_i(t)$  les fonctions compteurs des transitions  $t_i$ , pour  $i = 1$  à  $3$  et  $u(t)$  est l'équation compteur de la transition d'entrée.

### 1.3.3.2 Fonction dateur

On associe une fonction dateur  $\theta_i(k)$  pour chaque transition  $t_i$ ,  $i = 1$  à  $3$ ,  $u(k)$  est la fonction dateur attribuée à la transition source, pour un franchissement au plus tôt, les fonctions dateurs vérifient les équations suivantes :

$$\begin{cases} \theta_1(k) = \theta_2(k-1) \oplus u(k) \\ \theta_2(k) = 1 \theta_1(k) \\ \theta_3(k) = 1 \theta(k) \end{cases}$$

L'équation (1.7) donne

$$\theta(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon \end{bmatrix} \theta(k) \oplus \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(k-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix} u(k)$$

L'équation explicite (1.8) est donnée par

$$\theta(k) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & 2 & \varepsilon \end{bmatrix} \theta(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 2 \end{bmatrix} u(k)$$

Alors l'équation d'état est la suivante

$$x(k) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & 2 & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 2 \end{bmatrix} u(k)$$

Avec  $x_i(k)$  les fonctions dateurs des transitions  $t_i$  pour  $i = 1$  à  $3$  et  $u(k)$  est l'équation dateur de la transition d'entrée.

## 1.4 Les graphes d'événements temporisés en conflit

Le comportement des graphes d'événements temporisés est décrit par des équations linéaires dans l'algèbre des dioïdes [2] [1]. En revanche, toute place d'un GET ne peut être reliée au maximum qu'à une transition en amont et une transition en aval. Les GET ne permettent donc pas de représenter les phénomènes des concurrences ou de choix multiples qui caractérisent les systèmes avec partage de ressources. Pour une étude analytique des systèmes de commande plus complexes une modélisation des conflits est nécessaire.

### 1.4.1 Problème de ressource partagé

Le problème des ressources partagées est couramment rencontré dans les SED, il survient à chaque fois que deux utilisateurs ou plus souhaitent accéder à la même ressource au même moment.

#### Exemple 1.10

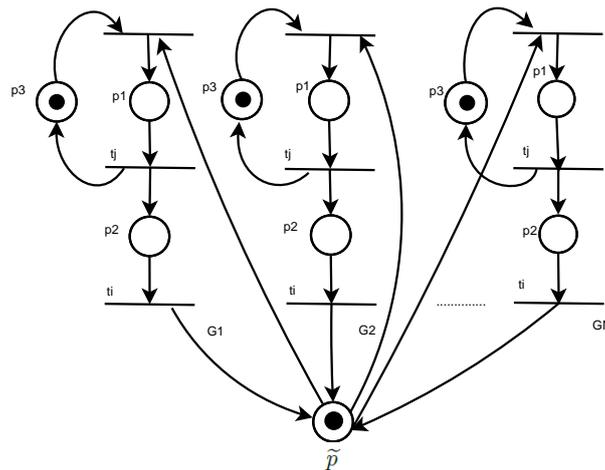


FIGURE 1.7 – Exemple d'un RGETC

#### Définition 1.13

Une place est dite *sauf si seulement si*,  $\forall$  l'évolution du RDP ( $\forall m \in R(m_0)$ ), le marquage initial est inférieur ou égale à 1.

$$m(p_{ij}) \leq 1$$

#### Définition 1.14 [9]

Un *conflit structurel* correspond à l'existence d'une place avec au-moins deux transitions de sortie notée :  $k = \{\tilde{p}, \{t_1, t_2, \dots\}\}$ .

**Définition 1.15**

Un conflit effectif désigne par  $k^E = \{\tilde{p}, \{t_1, t_2, \dots\}, M\}$  est un conflit structurel  $k = \{\tilde{p}, \{t_1, t_2, \dots\}\}$ , avec un marquage  $m$  tel que le nombre de marquage de  $\tilde{p}$  est inférieure au nombre de transition de sortie de  $\tilde{p}$ .

**Exemple 1.11** Cette exemple représente un conflit effectif.

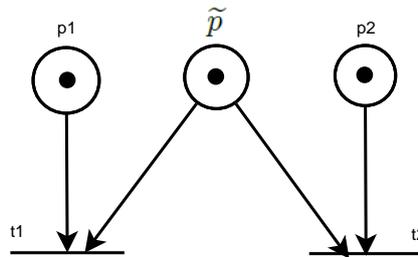


FIGURE 1.8 – Exemple de conflit effective

**Conclusion** Dans ce chapitre, nous avons présenté quelques définitions et propriétés des RDP ainsi que de l'algèbre des dioïdes. Nous avons aussi introduit les graphes d'événements temporisés et leur propriétés, par la suite, on montre comment obtenir à partir d'un GET des modèles linéaires dans l'algèbre (Max-plus, Min-plus). Des définitions sur les réseaux des graphes d'événements temporisés en conflit sont présentées et utilisées par la suite.

# Chapitre 2

## Commande de graphes d'événements temporisés dans l'algèbre Max-plus

### 2.1 Introduction

La commande est un concept fondamental dans la structure et la conception des systèmes à événements discrets. Cette commande consiste à établir un contrôleur qui agit avec le système considéré pour satisfaire les spécifications imposées sur le cahier des charges. Plusieurs approches ont été développées pour synthétiser un superviseur afin de garantir le bon fonctionnement de cette catégorie de systèmes [5] [4] [3]. Le fonctionnement de la plupart des processus industriels exige la satisfaction d'un certain nombre de contraintes temporelles ce qui implique que le temps est une composante primordiale dans les systèmes à événements discrets. Dans ce chapitre, des lois de commande qui permettent de satisfaire des contraintes temporelles dans le cas des graphes d'événements temporisés sont présentés. Les résultats de la technique présentés sont due à [1].

### 2.2 La commande des systèmes à événements discrets sous contraintes de temps dans l'algèbre $(\text{Max}, +)$ : [1] [2]

Le comportement au plus tôt des graphes à événements temporisés peuvent être représentés par l'équation suivante :

$$x(k) = Ax(k-1) \oplus Bu(k) \quad (2.1)$$

$$A \in \mathbb{R}_{max}^{n \times n}, B \in \mathbb{R}_{max}^{n \times q}, x \in \mathbb{R}_{max}^n, u \in \mathbb{R}_{max}^{n \times q}$$

les valeurs  $n$  et  $q$  sont des entiers qui représentent respectivement le nombre des transitions internes et le nombre des transitions de commandes.

ou :

$u(k)$  représente la commande.

$x(k)$  est le vecteur d'état considéré.

### 2.2.1 Contrainte temporelle

Plusieurs approches ont été développées pour la commande des GET sous contraintes temporelles imposées sur des places, des méthodes proposés dans [2] sont basé sur l'utilisation des GET et de l'algèbre des diodes linéaires. On considère  $p_{ij}$  la place soumise a une contrainte temporelle stricte, un intervalle de temps  $[\tau_{ij}^{min}, \tau_{ij}^{max}]$  est associé à cette place comme illustré sur la figure (2.1) suivante :

$$\tau_{ij}^{min} \leq \tau_{ij}^{max}$$

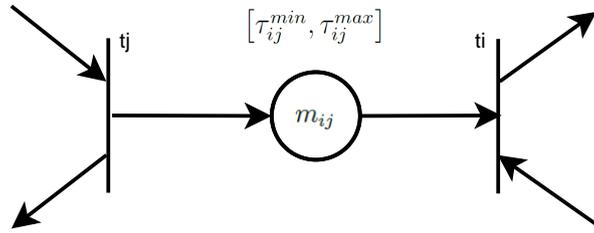


FIGURE 2.1 – Contrainte temporelle imposé sur une place

Cette contrainte se traduit dans l'algèbre  $\bar{\mathbb{R}}_{max}$  par l'équation suivante :

$$\tau_{ij}^{min} x_j(k - m_{ij}) \leq x_i(k) \leq \tau_{ij}^{max} x_j(k - m_{ij})$$

Où  $m_{ij}$  est le marquage initial de la place  $p_{ij}$ . L'inéquation de droite est satisfaite en tenant compte de la définition d'un graphe d'événements temporisé et du modèle (Max-Plus) linéaire associé à son comportement. La deuxième inéquation donnée par l'égalité (2.2) :

$$x_i(k) \leq \tau_{ij}^{max} x_j(k - m_{ij}) \quad (2.2)$$

dont le produit est défini dans  $\bar{\mathbb{R}}_{max}$ , est une contrainte supplémentaire à satisfaire, la condition (2.2) impose le temps nécessaire  $\tau_{ij}^{max}$  à ne pas dépassée pour franchir un nombre de jetons  $m_{ij}$  dans la place  $p_{ij}$ . Il faudrait alors calculer des lois de commande causales  $u(k)$  pour garantir le respect de cette contrainte temporelle.

### 2.2.2 Commande causale

Nous considérons un processus modélisé par une équation Max-Plus linéaire est soumis à une ou plusieurs contraintes temporelles. Nous souhaitons déterminer les valeurs de  $u(k)$  pour garantir le respect de ces contraintes de temps. Nous cherchons des lois de commande qui sont réalisables et bien posées sous la forme suivante :

$$u(k) = F x(k - 1)$$

pour  $k > 1$ , ou  $F \in \bar{\mathbb{R}}_{max}^{m \times N}$  et le produit est au sens de l'algèbre Max-Plus.

La commande à un  $k^{\text{ème}}$  franchissement est en fonction des variables d'état à un  $(k - 1)$  franchissement des jetons, elle représente un retour d'état que nous appellerons parfois feedback. Cette commande est réalisable et elle peut s'interpréter par un graphe de commande. Ce graphe de commande est connecté au graphe d'événements temporisé initial pour le contrôler et assurer le respect des contraintes temporelles.

## 2.3 Commande temporelle des systèmes Max-Plus linéaires

Considérant le cas de commande des graphes d'événements temporisés à une seule transition d'entrée, ce dernier est soumis à une seule contrainte temporelle où à plusieurs contraintes temporelles, qui sont modélisés par des équations Max-Plus. Ensuite, généralisant l'étude pour le cas de commande de GET à plusieurs transitions d'entrées.

### Hypothèse 1 [1]

Il existe au moins un chemin  $\alpha$  qui relie la transition de commande  $t_u$  à la transition d'entrée  $t_j$  de la place sous contrainte.

### Hypothèse 2

Chaque transition de graphe d'évènement temporisé est franchie dès qu'elle est franchissable (franchement au plus tôt).

### 2.3.1 Commande monovariante

Comme on n'a vu déjà le comportement d'un graphe d'événements temporisé peut être représenté par une équation linéaire dans l'algèbre Max-Plus. Nous considérons l'équation linéaire suivante :

$$x(k) = A x(k - 1) \oplus B u(k) \tag{2.3}$$

où  $x \in \bar{\mathbb{R}}_{max}^n$ ,  $A \in \bar{\mathbb{R}}_{max}^{n \times n}$ ,  $B \in \bar{\mathbb{R}}_{max}^{n \times m}$ , et les valeurs  $n$  et  $m$  sont des entiers qui représentent respectivement le nombre de transitions internes, le nombre de transitions commande. Et la composante  $x_i(k)$ ,  $i = 1$  à  $N$ , désigne la date du  $k^{\text{ème}}$  franchissement de la transition  $t_i$  et le coefficient  $u_l(k)$ , pour  $l = 1$  à  $m$ , définit la fonction dateur associée à la transition de commande  $tu_l$ .

### 2.3.1.1 Une seule contrainte et une seule commande

Considérant un graphe d'événement temporisé  $G$  comportant une seule contrainte temporelle, ce graphe est modélisé par des équations Max-Plus. Cette contrainte est imposée sur la place du chemin  $\rho$  qui relié la transition  $t_j$  à la transition  $t_i$  du graphe  $G$ . La temporisation associée à ce chemin est notée  $\tau_\rho$ , cette temporisation représente la somme des temporisations associées aux places de ce chemin. Les transitions du graphe  $G$  ne sont pas commandables, la transition d'entrée notée  $t_u$  est la seule transition commandable. Notre objectif est de synthétiser un contrôleur  $u(k)$  qui garantit la satisfaction de la contrainte temporelle.

On appliquant la **propriétés 1.1**, pour  $\tau = \phi$ , avec  $\phi \geq 1$ , on obtient l'équation (Max-plus) suivante :

$$x(k) = A^\phi x(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} A^{k'} B u(k - k') \right] \quad (2.4)$$

La  $i^{\text{ème}}$  composante des termes sont données par

$$x_i(k) = \bigoplus_{r=1}^N A_{ir}^\phi x_r(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i u(k - k') \right] \quad (2.5)$$

Ce GET est soumis sous à la contrainte temporelle suivante :

$$x_i(k) \leq \tau_\rho^{max} x_j(k - m_\rho) \quad (2.6)$$

Cette contrainte exprime la durée maximale à ne pas dépasser par un jeton entrant dans le chemin  $\rho$ .

On remplace  $x_i(k)$  dans la contrainte on obtient

$$\bigoplus_{r=1}^N A_{ir}^\phi x_r(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i u(k - k') \right] \leq \tau_\rho^{max} x_j(k - m_\rho) \quad (2.7)$$

Les contraintes temporelles sont équivalentes aux

$$\bigoplus_{r=1}^N A_{ir}^\phi x_r(k - \phi) \leq \tau_\rho^{max} x_j(k - m_\rho) \quad (2.8)$$

$$\left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} B)_i u(k - k') \right] \leq \tau_\rho^{max} x_j(k - m_\rho) \quad (2.9)$$

A partir de **L'hypothèse 1**, et de la définition des fonctions dateurs, il existe une relation qui relie la transition source  $t_u$  et la transition d'entrée  $t_j$  qui est donnée sous la forme suivante :

$$x_j(k) \geq (A^{m_\alpha} B)_j u(k - m_\alpha) \quad (2.10)$$

Et on prend en considération de **l'hypothèse 2** (franchissement au plutôt), on remplaçant la fonction  $x_j(k)$  par son équation (2.10) dans (2.7) et on obtient :

$$\bigoplus_{r=1}^N A_{ir}^\phi x_r(k - \phi) \bigoplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} B)_i u(k - k') \right] \leq \tau_\rho^{max} (A^{m_\alpha} B)_j u(k - m_\rho - m_\alpha) \quad (2.11)$$

Donc la contrainte (2.6) est satisfaite si les deux inéquations suivantes sont vérifiées :

$$\bigoplus_{r=1}^N A_{ir}^\phi x_r(k - \phi) \leq \tau_\rho^{max} (A^{m_\alpha} B)_j u(k - m_\rho - m_\alpha) \quad (2.12)$$

$$\left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} B)_i u(k - k') \right] \leq \tau_\rho^{max} (A^{m_\alpha} B)_j u(k - m_\rho - m_\alpha) \quad (2.13)$$

à partir de ces inéquations on obtient

$$u(k - m_\rho - m_\alpha) \geq \bigoplus_{r=1}^N (A_{ir}^\phi - \tau_\rho^{max} - (A^{m_\alpha} B)_j) .x_r(k - \phi) \quad (2.14)$$

$$\bigoplus_{k'=0}^{\phi-1} (A^{k'} B)_i u(k - k') \leq \tau_\rho^{max} (A^{m_\alpha} B)_j u(k - m_\rho - m_\alpha) \quad (2.15)$$

Dans le cas où les conditions 2.14-2.15 sont vérifiées, la contrainte temporelle est satisfaite.

### **Théorème 2.1**

Nous considérons un graphe d'événement temporisé  $G$  avec une seule transition de commande et soumis à une seule contrainte temporelle de la forme (2.6). Nous supposons qu'il existe un chemin qui relie la transition de commande  $t_u$  à la transition d'entrée de chemin  $\alpha$  sous contrainte avec ( $m_\alpha = 0$ ). Le chemin  $\alpha$  n'est pas marqué initialement. Le contrôleur décrit par l'équation suivante :

$$u(k) = \bigoplus_{r=1}^N F x_r(k-1)$$

Avec  $F_r = \max(0, A_{ir} - \tau_\rho^{\max} - B_j)$ , satisfait la contrainte temporelle 2.6 si la condition suivante

$$B_i \leq \tau_\rho^{\max} B_j$$

est vérifiée.

### Démonstration

On cherche un contrôleur sous la forme  $u(k) = F x(k-1)$  et pour cela on prend  $\phi = 1$ , ce qui signifie que le marquage  $m_\alpha = m_\rho = 0$  alors  $0 \leq k' \leq \phi - 1$  ce qui implique  $k' = 0$ .

Nous remplaçons  $\phi$ ,  $m_\rho$ ,  $m_\alpha$  par leur valeurs dans les inéquations (2.14) et (2.15) et on obtient la loi de commande suivante :

$$u(k) \geq \bigoplus_{r=1}^N (A_{ir} - \tau_\rho^{\max} - B_j) x_r(k-1) \quad (2.16)$$

Ce feedback est vérifié, si les deux conditions

$$\begin{cases} \phi = m_\rho + m_\alpha + 1 \\ (A^\phi)_{ir} \geq \tau_{ij}^{\max} (A^{m_\alpha} B)_j \end{cases}$$

avec

$r = 1$  à  $N$ , pour  $(A^\phi)_{ir} \neq \varepsilon$  sont vérifiées

L'inéquation (2.15) elle est satisfaite, si les inéquations suivantes sont respectées

$$(A^{k'} B)_i \leq \tau_{ij}^{\max} (A^{m_\alpha} B)_j \quad (2.17)$$

$$u(k-k')_i \leq u(k-m_\alpha-m_\rho) \quad (2.18)$$

avec  $k' = 0$  à  $\phi - 1$ .

#### 2.3.1.2 Plusieurs contraintes et une seule commande

Considérant  $S$  contraintes temporelles d'un graphe d'événement temporisé, avec une seule transition de commande  $t_u$  soumis à plusieurs contraintes temporelles  $p_s$  pour  $s = 1$  à  $S$ .

Pour chaque place  $p_s$  une contrainte. Nous notons par  $m_s$ ,  $\tau_s^{\min}$ ,  $\tau_s^{\max}$ , respectivement le marquage initial, la temporisation minimale et la temporisation maximale,  $x_s$  et  $x_{s'}$  représente les fonctions dateurs des transitions  $t_s$  et  $t_{s'}$ . Les contraintes temporelles sont exprimées par les inéquations suivantes :

$$\tau_s^{\min} x_s(k - m_s) \leq x_{s'}(k) \leq \tau_s^{\max} x_s(k - m_s) \quad (2.19)$$

avec  $s = \{1, 2, \dots, S\}$ .

L'inéquation

$$\tau_s^{\min} x_s(k - m_s) \leq x_{s'}(k) \quad (2.20)$$

vérifie la contrainte temporelle .

Supposons que le système (2.3) est vérifié, la contrainte temporelle se ramène alors à

$$x_{s'}(k) \leq \tau_s^{\max} x(k - m_s) \quad (2.21)$$

avec  $s = \{1, 2, \dots, S\}$ .

pour chaque contrainte  $s$ , on suppose qu'il y a un chemin noté  $\alpha_s$  qui relie la transition  $t_u$  à  $t_s$  avec  $m_{\alpha_s}$  le marquage initial du chemin.

$$(A^{m_{\alpha_s}}.B)_s u(k - m_{\alpha_s}) \leq x_s(k) \quad (2.22)$$

on note  $u_s(k)$  les lois de commandes calculées, pour satisfaire la  $S^{\text{ème}}$  contrainte temporelle (2.21), dans le **Théorème 2.2** nous déterminons un feedback qui vérifie tous les contraintes temporelles .

### Théorème 2.2

L'inéquation

$$u(k) = \bigoplus_{s=1}^S u(k)$$

avec  $\phi_s = 1$  et

$$u_s(k) = \bigoplus_{r=1}^N (A_{s'r} - B_s - \tau_s^{\max}) x(k - 1) \quad (2.23)$$

définit une commande causale qui respecte toute les contraintes temporelles, si les conditions suivantes sont satisfaite pour  $s = \{1, 2, \dots, S\}$ , on a

$$\begin{cases} m_s = m_{\alpha_s} = 0 \\ A_{s'r} \geq \tau_s^{\max} B_s \end{cases}$$

pour  $r = 1$  à  $N$ .

**Exemple 2.1** Nous considérons le graphe d'événements temporisé donné par la figure (2.2), ce graphe est soumis à une contrainte temporelle qui est donnée par l'inéquation suivante :

$$x_2(k) \leq 4x_1(k)$$

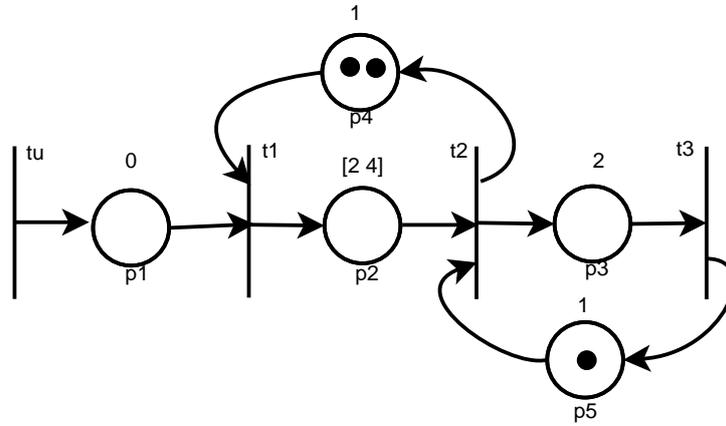


FIGURE 2.2 – Exemple d'un GET plusieurs contraintes

Nous calculons des lois de commande causales pour satisfaire cette contrainte temporelle. Nous décomposons les places ayant plus d'un jeton pour avoir un nouveau graphe d'événements temporisés équivalent avec des places marquées, nous décomposons la place  $p_4$  en deux place notées  $p_{42}$  et  $p_{14}$  qui sont marquées à 1. Et on obtient le graphe étendu suivant :

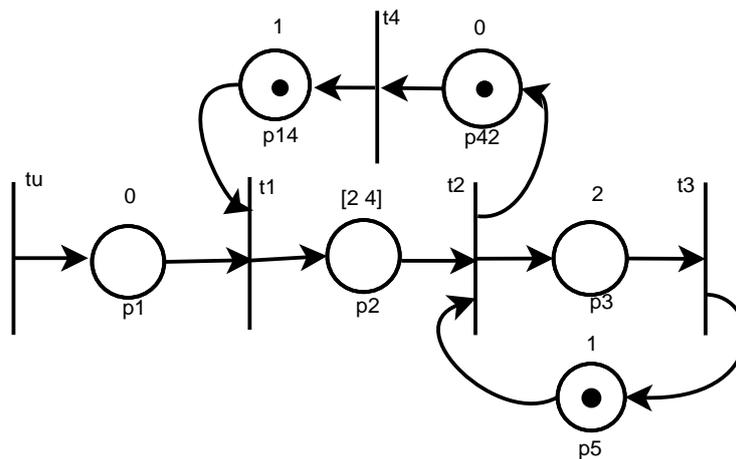


FIGURE 2.3 – Exemple d'un GET étendu

L'équation d'état de ce graphe d'événements temporisés étendu est :

$$x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 2 & 2 \\ \varepsilon & \varepsilon & 3 & 3 \\ \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 3 \\ \varepsilon \end{bmatrix} u(k)$$

où  $x(k) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$  est un vecteur d'état qui sont associées aux transitions  $t_1, t_2, t_3, t_4$ ,

et  $u(k)$  est le dateur de la transition source  $t_u$ , la temporisation maximale associée à la place  $p_{ij}$  dans notre exemple est  $\tau_{ij}^{max} = \tau_{12}^{max} = 4$ .

Il existe un chemin qui relie la transition source  $t_u$  à la transition  $t_j$ , la temporisation dans ce cas est égale à  $\tau_\alpha = 0$ , le marquage initial  $m_\alpha = 0$ .

$$\tau_\alpha + \tau_{ij}^{max} = \tau_\alpha + \tau_{12}^{max} = 4$$

$A_{ir} = A_{2r} = (\varepsilon, \varepsilon, 2, 2)$  et  $B_i = B_2 = 1$  pour  $r = \{1..4\}$ .

On voit que les conditions données dans le **Théorème 2.6** sont bien vérifiées, donc la loi de commande est donnée par :

$$u(k) = x_3(k-1) \oplus x_4(k-1)$$

Ce contrôleur peut être simplifié, d'après le graphe de la figure (2.4), c'est clair que, la date de franchissement de la transition  $t_3$  est supérieure à la date de franchissement de la transition  $t_4$ ,  $x_3(k-1) \geq x_4(k-1)$ , Alors la loi de commande est donnée comme suit :

$$u(k) = x_3(k-1)$$

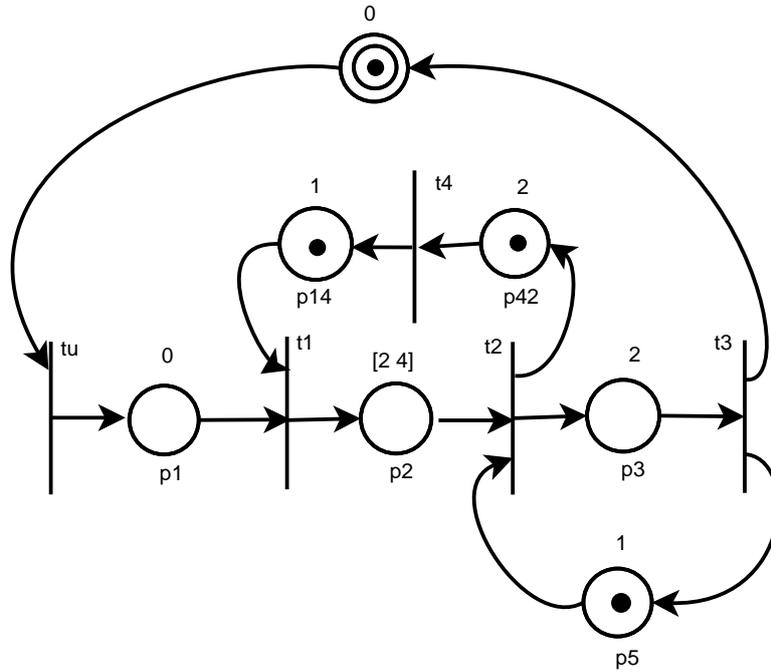


FIGURE 2.4 – La loi de commande du GET

### 2.3.2 Commande multivariable

Dans cette sous section, on considère un GET avec plusieurs commandes, le comportement de ce graphe est représenté par l'inéquation Max-plus linéaire, ce graphe est soumis à des contraintes temporelles.

#### 2.3.2.1 Une seule contrainte plusieurs commandes

Nous supposons un GET soumis à des contraintes temporelles dans une seule place  $p_{ij}$ , on associe chaque transition source une fonction dateur  $u_l(k)$ , avec  $l = 1$  à  $m$ .

$$x_j(k) \geq (A^{m_\alpha} B)_{js} u_s(k - m_\alpha) \quad (2.24)$$

On suppose que  $m_\alpha = m_{ij} = 0$ , et  $\phi_s = m_\alpha + m_{ij} + 1$ .

Les inéquations sont données par :

$$\bigoplus_{r=1}^N A_{ir} x_r(k-1) \leq \tau_{ij}^{max} B_{js} u_s(k) \quad (2.25)$$

$$\bigoplus_{l=1}^m B_{il} u_l(k) \leq \tau_{ij}^{max} B_{js} u_s(k) \quad (2.26)$$

La contrainte temporelle (2.21) est impliquée aux deux conditions (2.25) et (2.26).

La  $S^{ème}$  composantes du vecteur de commande  $u(k)$  est donnée par :

$$u_s(k) \geq \bigoplus_{r=1}^N (A_{ir} - B_{js} - \tau_{ij}^{max}) x_r(k-1)$$

Cette équation permet de vérifier la condition (2.25) si la condition (2.26) est vérifiée

$$A_{ir} \geq \tau_{ij}^{max} B_{js} \quad (2.27)$$

La condition (2.26) est vérifiée si les conditions suivantes sont satisfaites

Pour  $l = 1$  à  $m$

$$B_{il} \leq \tau_{ij}^{max} B_{js} \quad (2.28)$$

$$u_l(k) \leq u_s(k) \quad (2.29)$$

### **Théorème 2.3 [1]**

Soit  $G$  un GET avec  $m$  transitions d'entrées et soumis à une seule contrainte sous la forme de (2.21). Le respect de la contrainte temporelle (2.21) est garanti s'il existe un indice  $s$  tel que

$$u_s(k) = \bigoplus_{r=1}^N ((A^\phi)_{ir} - (A^{\tau_\alpha} B)_{js} - m_{ij}) x_r(k-1)$$

avec :

$$\phi = \tau_\alpha + \tau_{ij}^{max} + 1, u_l(k) = \varepsilon, \text{ pour } l \neq s.$$

#### **2.3.2.2 Plusieurs contraintes et plusieurs commandes**

Nous considérons un GET soumis à  $s$  contraintes temporelles, ce graphe est modélisé par des équations Max-Plus avec  $s \geq 1$ , chaque place contrainte  $p_s$  admet  $t_s$  comme transition d'entrées et  $t_{s'}$  transition de sortie de la place  $p_s$ , nous supposons pour chaque contrainte  $s$  qu'il existe un chemin noté  $\alpha_s$  qui relie la transition de commande  $t_{z_s}$  et la transition d'entrée  $t_s$ , ou  $z_s$  désigne l'indice associé à la transition de commande qui agit sur la contrainte  $s$ , on note  $\tau_s$  la temporisation de ce chemin avec un marquage initiale  $m_{\alpha_s}$  avec  $s = 1$  à  $S$ , on aura cette condition

$$x_s(k) \geq (A^{m_{\alpha_s}} \cdot B)_{s z_s} u_{z_s}(k - m_{\alpha_s}) \quad (2.30)$$

sous la contrainte temporelle suivante

pour  $s = 1$  à  $S$

$$x_{s'}(k) \leq \tau_s^{max} x_s(k - m_{\alpha_s}) \quad (2.31)$$

L'équation (2.18) peut s'écrire sous cette forme

$$x(k) = \bigoplus_{r=1}^N ((A^{\phi_s}) x(k - \phi_s)) \bigoplus \left[ \bigoplus_{k=0}^{\phi_s-1} ((A^{k'} B) u(k - k')) \right]$$

L'équation explicite de la transition de sortie  $t_{s'}$  est :

$$x_{s'}(k) = \bigoplus_{r=1}^N (A^{\phi_s})_{s'r} x_r(k - \phi_s) \bigoplus \left[ \bigoplus_{k=0}^{\phi_s-1} \left( \bigoplus_{l=1}^m (A^{k'} B)_{s'l} u_l(k - k') \right) \right] \quad (2.32)$$

Nous remplaçons dans l'équation (2.31) la fonction  $x_{s'}(k)$  par son équation précédente et on obtient ces conditions suivantes

$$\bigoplus_{r=1}^N ((A^{\phi_s})_{s'r} x_r(k - \phi_s)) \leq \tau_s^{max} x_s(k - m_{\alpha_s}) \quad (2.33)$$

$$\left[ \bigoplus_{k=0}^{\phi_s-1} \left( \bigoplus_{l=1}^m (A^{k'} B)_{s'l} u_l(k - k') \right) \right] \leq \tau_s^{max} x_s(k - m_{\alpha_s}) \quad (2.34)$$

Nous traitons chacune des contraintes temporelles séparément, pour cela nous utilisons les résultats pour le cas de la commande d'une seule contrainte obtenus précédemment, pour  $s = 1$  à  $S$ , la contrainte temporelle  $s$  est équivalente aux inégalités suivantes

$$\bigoplus_{r=1}^N (A^{\phi_s})_{s'r} x_r(k - \phi_s) \leq \tau_s^{max} (A^{m_{\alpha_s}} B)_{s z_s} u_{z_s}(k - m_s - m_{\alpha_s}) \quad (2.35)$$

$$\bigoplus_{k=0}^{\phi_s-1} \left( \bigoplus_{l=1}^m (A^{k'} B)_{s'l} u_l(k - k') \right) \leq \tau_s^{max} (A^{m_{\alpha_s}} B)_{s z_s} u_{z_s}(k - m_s - m_{\alpha_s}) \quad (2.36)$$

Le feedback donné par l'inéquation suivante

$$u_{z_s}(k) \geq \bigoplus_{r=1}^N [((A^{\phi_s})_{s'r} - (A^{m_{\alpha_s}} B)_{s z_s} - \tau_s^{max}) x_r(k - 1)] \quad (2.37)$$

vérifie la condition (2.36) si les deux conditions suivantes sont vérifiées

$$\phi_s = m_s + m_{\alpha_s} + 1$$

$$\bigoplus_{r=1}^N (A^{\phi_s})_{s'r} \geq \tau_s^{max} (A^{m_{\alpha_s}} B)_{s z_s} \quad (2.38)$$

avec  $s = 1$  à  $S$ .

La relation entre les composantes du vecteur de commande est traduite dans l'inéquation (2.36), et aussi elle exprime l'influence de chacune d'elle sur les autres, nous remarquons les équations suivantes

$$(A^{k'} B)_{s'l} \leq \tau_s^{max} (A^{m_{\alpha_s}} B)_z$$

$$u_l(k - k') \leq u_{z_s}(k - m_s - m_{\alpha_s})$$

avec  $k' = 0$  à  $\phi_s - 1$

et  $l = 1$  à  $m$

**Le Théorème 2.4** suivant définit un vecteur de commande  $u(k)$  qui valide toutes les contraintes temporelles.

**Théorème 2.4 [2]** Le vecteur  $u(k)$  donné par l'équation (2.39) définit une loi de commande causale qui garantit le respect de toutes les  $s$  contraintes temporelles

pour  $l = z_1$  à  $z_s$

$$u_l(k) = \bigoplus_{s=1}^S \left[ \bigoplus_{r=1}^N (A_{s'r} - B_{s_zs} - \tau_s^{max}) x_r(k - 1) \right] \quad (2.39)$$

avec

$m_s = m_{\alpha_s} = 0$ ,  $\phi_s = 1$  et  $u_l(k) = \varepsilon$

Si et seulement si, les conditions suivantes (2.40) et (2.41) :

pour  $s = 1$  à  $S$

$$A_{s'r} \leq \tau_s^{max} B_{s_zs} \quad (2.40)$$

$$B_{s'l} \leq \tau_s^{max} B_{s_zs} \quad (2.41)$$

Avec  $r = 1$  à  $N$ ,  $l = 1$  à  $m$ , sont vérifiées.

**Exemple 2.2** Nous considérons le graphe d'événement temporisé donné par la figure .

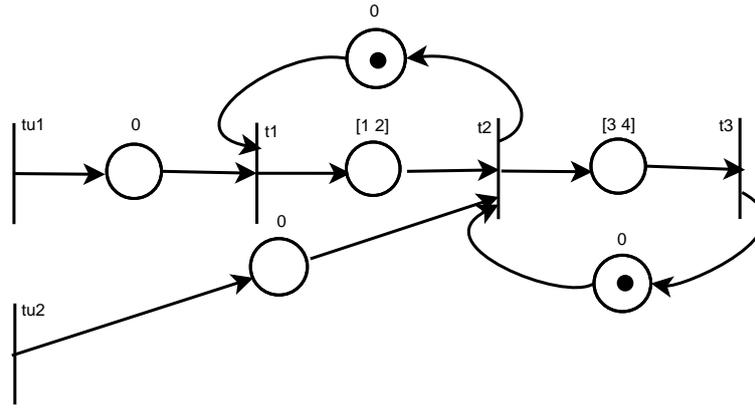


FIGURE 2.5 – Graphe d'événements temporisé dans Max-plus

Ce graphe est soumis à deux contraintes temporelles notées  $s_1$  et  $s_2$ , qui sont représenté par les deux inéquations Max-plus suivantes :

$$s_1 : x_2(k) \leq 2 x_1(k) \quad (2.42)$$

$$s_2 : x_3(k) \leq 4 x_2(k) \quad (2.43)$$

où :  $x_1(k)$ ,  $x_2(k)$ ,  $x_3(k)$  désignent les fonctions dateurs respectivement des transitions  $t_1$ ,  $t_2$ ,  $t_3$ .

Les fonctions dateurs vérifient :

$$\begin{cases} x_1(k) = u_1(k) \oplus x_2(k-1) \\ x_2(k) = 1 u_1(k) \oplus 1 x_2(k-1) \oplus x_3(k-1) \oplus u_2(k) \\ x_3(k) = 4 u_1(k) \oplus 4 x_2(k-1) \oplus 3 x_3(k-1) \oplus 3 u_2(k) \end{cases}$$

L'équation d'état Max-Plus qui représente la dynamique du graphe est donnée comme suit :

$$x(k) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & e \\ \varepsilon & 4 & 3 \end{bmatrix} x(k-1) \oplus \begin{bmatrix} e & \varepsilon \\ 1 & e \\ 4 & 3 \end{bmatrix} u(k) \quad (2.44)$$

Nous appliquons le **Théorème 2.4** pour déterminer des feedbacks qui vont assurer le respect des contraintes temporelles. Pour chaque contrainte temporelle  $s$ , pour  $s = 1$  à  $2$ , on vérifie d'abord qu'il existe un chemin  $\alpha_s$  qui relie une transition d'entrée  $tu_{zs}$  à la transition  $t_s$ . On voit bien dans la contraintes  $s_1$  qu'il existe un chemin  $a_1$  qui relie la transition d'entrée  $tu_1$  et la transition  $t_1$ , dans la contrainte  $s_2$  il existe un chemin  $a_2$  qui relie la transition  $tu_2$  à la transition  $t_2$ , avec un marquage initiale nul  $m_{a_1} = m_{a_2} = 0$ .

Loi de commande causale qui garantit le respect des contraintes temporelles existe.

Seulement si les conditions suivantes sont vérifiées :

$$A_{2r} \leq 1 + B_{11}$$

$$B_{2l} \leq 1 + B_{11}$$

$$A_{3r} \leq 1 + B_{22}$$

$$B_{3l} \leq 1 + B_{22}$$

Nous avons  $B_{11} = B_{22} = e$ ,  $B_{2l} = \begin{bmatrix} 1 & e \end{bmatrix}$ ,  $B_{3l} = \begin{bmatrix} 4 & 3 \end{bmatrix}$ ,  $A_{2r} = \begin{bmatrix} \varepsilon & 1 & e \end{bmatrix}$ ,  
 $A_{3r} = \begin{bmatrix} \varepsilon & 4 & 3 \end{bmatrix}$ .

Les composantes du vecteur de commande  $u(k)$  sont données comme suit :

$$u_1(k) = \bigoplus_{r=1}^3 [(A_{2r} - 1) x_r(k-1)] = [(1-1)x_2(k-1) \oplus (e-1)x_3(k-1)]$$

$$u_2(k) = \bigoplus_{r=1}^3 [(A_{3r} - 1) x_r(k-1)] = [(4-1)x_2(k-1) \oplus (3-1)x_3(k-1)]$$

Finalement on obtient le vecteur suivant :

$$\begin{cases} u_1(k) = e x_2(k-1) \oplus e x_3(k-1) \\ u_2(k) = 3 x_2(k-1) \oplus 2 x_3(k-1) \end{cases}$$

$$u(k) = \begin{cases} e x_2(k-1) \oplus e x_3(k-1) \\ 3 x_2(k-1) \oplus 2 x_3(k-1) \end{cases}$$

Définit un feedback causal qui garantit le respect des contraintes temporelles.

**Conclusion** Dans ce chapitre, nous avons présenté une approche de commande qui est basée sur l'utilisation des GETs dans l'algèbre Max-Plus. Cette approche permet de définir des lois de commande en boucle fermée, a fin de satisfaire les contraintes temporelles imposées sur le GET. Les lois de commande que nous avons obtenu peuvent être représentées par des places de contrôle connectées au graphe initial.

# Chapitre 3

## Modélisation des RGETC dans l'algèbre Max-Plus

### 3.1 Introduction

Ce chapitre traite des différentes manières de modélisation des graphes à événement temporisés en conflits dans l'algèbre  $(\text{Max}, +)$ . La première méthode est dû à [6] et permet le calcul du temps de cycle des graphes considérés. La deuxième manière de modélisation est dû à [8] et permet une représentation sous forme de système hybrides à commutation dans  $(\text{Max}, +)$ . Le modèle obtenu est utilisé pour calculer le temps de cycle de tel systèmes en connaissant une séquence de GET réalisable.

### 3.2 Méthode 1

#### 3.2.1 Modélisation de RGETC avec des équations récurrentes

Les GETs sont une classe des RDP au quelle on peut associé une représentation linéaire qui facilite leur étude. Des méthodes différentes ont été développé pour étudier le cas des réseaux des graphes d'événements temporisés en conflit dans [11] et [12], l'une de ces méthodes de modélisation est proposée dans [6], et nous permet le calcul du temps de cycle des graphes considérés.

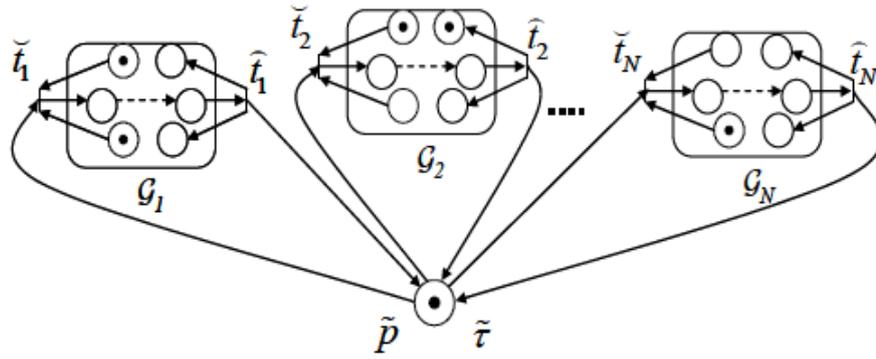


FIGURE 3.1 – Exemple de conflit

La figure (3.1) représente un réseau comprenant  $N$  graphes d'événements temporisés  $G_i$  pour consommer le jeton dans  $\tilde{p}$ .

- L'unique transition de GET qui est en aval de la place  $\tilde{p}$  est notée  $\tilde{t}_i$ , et l'unique transition en amont est notée  $\hat{t}_i$ , comme représenté sur la figure (3.1).
- Chaque GET est représenté par son équation d'état suivante :

$$x_i(k_i) = \widehat{A}_i \otimes x_i(k_i) \oplus \widehat{A}'_i \otimes x_i(k_i - 1) \oplus \widehat{B}_i \otimes u_i(k_i) \quad (3.1)$$

avec

$x_i = (x_i^1, x_i^2, \dots, x_i^{n_i})^t$  est le vecteur d'état de dimension  $n_i \times 1$  du GET  $G_i$ .

$u_i = (u_i^1, u_i^2, \dots, u_i^{m_i})^t$  est le vecteur de commande de dimension  $m_i \times 1$  du GET  $G_i$ .

dans la suite de l'étude, la  $i^{ime}$  ligne de la matrice  $M$  est notée  $M(i, :)$ ,  $M(i, j)$  désigne l'élément d'intersection entre la ligne  $i$  et la colonne  $j$  et  $M$  puissance  $i$  au sens  $(Max, +)$  est :

$$M \otimes M^{(i-1)} = \bigotimes_{j=1}^i M$$

### 3.2.2 Équation (Max-Plus) récurrente des RGETC

Considérons un réseau de graphes d'événements temporisés avec conflit  $G$ , on associe des dateurs aux transitions de chaque GET et un dateur à la place de conflit  $\tilde{p}$ , la date de disponibilité pour le  $k^{ème}$  fois du jeton de la place de conflit est notée  $\tilde{\Psi}(k)$ .

Supposons, sans perte de généralité, que la première composante du vecteur  $x_i$  (donc  $x_i^1$ ) est le dateur associé à la transition  $t_i$ . Si le  $k^{ème}$  jeton entrant dans  $\tilde{p}$  est consommé par le GET  $G_i$  et que ce dernier a déjà consommé  $(k_i - 1)$  jetons de  $\tilde{p}$ , nous avons :

$$x_i^1(k_i) = \widehat{A}_i(1, :) \otimes x_i(k_i) \oplus \widehat{A}'_i(1, :) \otimes x_i(k_i - 1) \oplus \widehat{B}_i(1, :) \otimes u_i(k_i) \oplus \tilde{\Psi}(k) \quad (3.2)$$

Comme les dates de tir des autres transitions de  $G_i$  ( $x_i^j$  avec  $j \geq 2$ ) ne dépendent pas

directement du jeton  $\tilde{p}$  (ces transitions n'étant pas en aval de  $\tilde{p}$ ), les composantes  $x_i^j$  de  $x_i(k_i)$  dans l'équation d'état (3.1) restent inchangées pour  $j \geq 2$  soit :

$$x_i^j(k_i) = \widehat{A}_i(j, :) \otimes x_i(k_i) \oplus \widehat{A}'_i(j, :) \otimes x_i(k_i - 1) \oplus \widehat{B}_i(j, :) \otimes u_i(k_i) \quad (3.3)$$

En combinant entre l'équation (3.3) et l'équation (3.2), nous aboutissons à la forme matricielle suivante :

$$x_i(k_i) = \widehat{A}_i \otimes x_i(k_i) \oplus \widehat{A}'_i \otimes x_i(k_i - 1) \oplus \widehat{B}_i \otimes u_i(k_i) \oplus \widehat{F}_i \otimes \Psi(k) \quad (3.4)$$

où toutes les matrices et vecteurs impliqués sont exactement les mêmes que ceux de l'équation (3.3). Les seuls nouveaux éléments sont  $\Psi(k)$  et la matrice  $\widehat{F}_i$  qui est de dimension  $n_i \times 1$ ,  $\widehat{F}_i = (e \ \varepsilon \ .\varepsilon)^t$  tel que toutes les composantes de  $\widehat{F}_i$  sont nulles ( $\varepsilon$ ) sauf la première composante qui est égale à ( $e$ ).

On appliquant le **Théorème 1.1** à l'équation (3.4), on obtient :

$$x_i(k_i) = A_i \otimes x_i(k_i - 1) \oplus B_i \otimes u_i(k_i) \oplus F_i \otimes \Psi(k) \quad (3.5)$$

$$A_i = \widehat{A}^*_i \otimes \widehat{A}'_i, \quad B_i = \widehat{A}^*_i \otimes \widehat{B}_i \quad \text{et} \quad F_i = \widehat{A}^*_i \otimes \widehat{F}_i.$$

Par ailleurs, sachant que le  $k^{\text{ème}}$  jeton de la place  $\tilde{p}$  est consommée par le GET  $G_i$ , le  $(k+1)^{\text{ème}}$  jeton de  $\tilde{p}$  est généré avec le tir, pour le  $k_i^{\text{ème}}$  fois de la transition  $\hat{t}_i$ , supposons que le dateur associé à  $\hat{t}_i$  est la dernière composante du vecteur  $x_i$  (soit  $x_i^{n_i}$ ), nous pouvons alors écrire :

$$\tilde{\Psi}(k+1) = \tilde{\tau} \otimes x_i^{n_i}(k_i) \quad (3.6)$$

Sa forme matricielle est :

$$\Psi(k+1) = G_i \otimes x_i(k_i) \quad (3.7)$$

ou

$G_i$  est une matrice de dimension  $1 \times n_i$  où tous ces composants son nuls sauf la dernière qui égale à la temporisation  $\tilde{\tau}$  de la place  $\tilde{p}$ , On obtient les équations suivantes pour le  $k^{\text{ème}}$  jeton de  $\tilde{p}$  qui est consommée par le GET  $G_i$  :

pour  $1 \leq i \leq N$  et  $k = \sum_{i=1}^N (k_i - 1)$

$$\begin{cases} x_i(k_i) = A_i \otimes x_i(k_i - 1) \oplus B_i \otimes u_i(k_i) \oplus F_i \otimes \Psi(k) \\ \Psi(k+1) = G_i \otimes x_i(k_i) \end{cases} \quad (3.8)$$

Nous avons obtenu des équations (Max-plus) qui régissent l'évolution des RGETC indépendamment de l'arbitrage du conflit, et aussi on peut les obtenir pour des RGETC avec plusieurs ressource de manière similaire.

**Exemple 3.1** Représentons un réseaux de graphes d'événements temporisés avec conflits, ce dernier est constitué de deux GETs qui partagent deux places de conflits  $\tilde{p}_1$  et  $\tilde{p}_2$ .

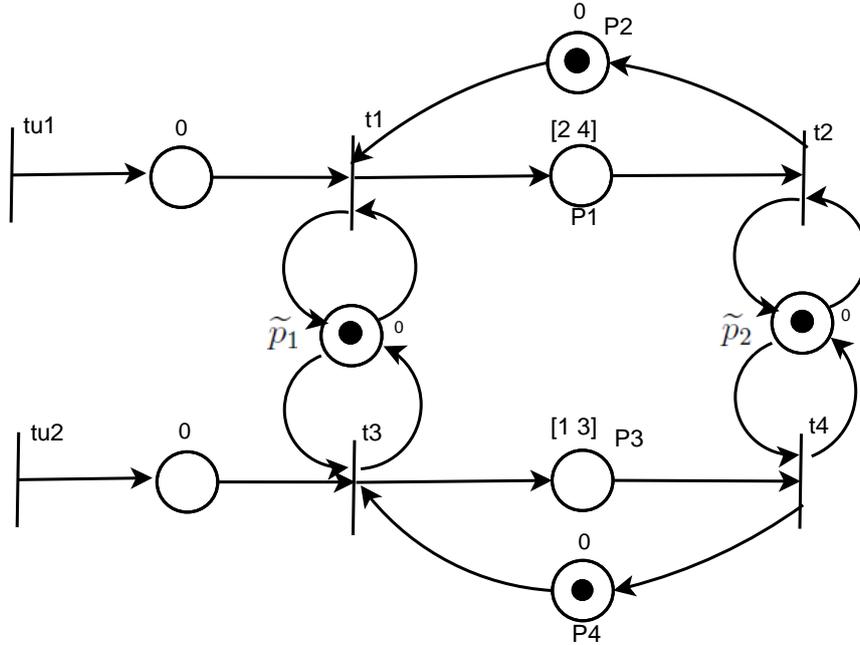


FIGURE 3.2 – Un RGETC avec deux GET et deux places de conflit

En associant des dateurs  $x_1^1, x_1^2, x_2^1$  et  $x_2^2$  aux transitions  $t_1, t_2, t_3$  et  $t_4$ . Si le  $k^{eme}$  jeton de  $\tilde{p}_1$  et  $\tilde{p}_2$  est consommé par le GET  $G_1$  et que ce dernier a été déjà consommé par  $(k_1 - 1)$  jetons des  $\tilde{p}_1$  et  $\tilde{p}_2$  nous avons :

$$\begin{cases} x_1^1(k_1) = u_2(k_2) \oplus x_2^2(k_2 - 1) \oplus \Psi_1(k) \\ x_1^2(k_1) = 2 \otimes u_1(k_1) \oplus 2 \otimes x_2^1(k_1 - 1) \oplus 2 \otimes \Psi_1(k) \oplus \Psi_2(k) \end{cases} \quad (3.9)$$

$$\begin{cases} x_1^1(k_1) = u_1(k_1) \oplus x_2^1(k_1 - 1) \oplus \Psi_1(k) \\ x_1^2(k_1) = 1 \otimes u_2(k_2) \oplus 1 \otimes x_2^2(k_2 - 1) \oplus 1 \otimes \Psi_1(k) \oplus \Psi_2(k) \end{cases} \quad (3.10)$$

Nous aboutissons finalement à la forme (3.8) suivante :

où :

$$A_1 = \begin{pmatrix} \varepsilon & e \\ \varepsilon & 2 \end{pmatrix}, A_2 = \begin{pmatrix} \varepsilon & e \\ \varepsilon & 1 \end{pmatrix}, B_1 = \begin{pmatrix} e \\ 2 \end{pmatrix}, B_2 = \begin{pmatrix} e \\ 1 \end{pmatrix}, F_1 = \begin{pmatrix} e & \varepsilon \\ 2 & e \end{pmatrix}, F_2 = \begin{pmatrix} e & \varepsilon \\ 1 & e \end{pmatrix}$$

$$G_1 = \begin{pmatrix} e & \varepsilon \\ \varepsilon & e \end{pmatrix}, G_2 = \begin{pmatrix} e & \varepsilon \\ \varepsilon & e \end{pmatrix}.$$

On va représenter une modélisation explicite d'un RGETC connaissant la politique appliquée pour l'arbitrage du conflit. On considère deux politiques, la règle du premier arrivé premier servi et une autre règle d'allocation périodique de la ressource.

### 3.2.2.1 La Règle du Premier Arrivé Premier Servi (PAPS)

La règle PAPS est exprimée comme suit : la transition  $\check{t}_i$  qui demande le jeton de la place du conflit  $\tilde{p}$  en premier gagne ce jeton, On dit qu'une transition demande le jeton de  $\tilde{p}$  pour la  $k^{\text{ème}}$  fois à la date  $\sigma_i(k_i)$ , si toutes les places en amont de  $\check{t}_i$  (exceptée  $\tilde{P}$ ) ont un jeton disponible. Sur la figure (3.2) par exemple, la transition  $t_1$  demande le jeton pour la  $k^{\text{ème}}$  fois à partir du moment où il y a un jeton disponible dans les places  $\tilde{p}_1$  et  $\tilde{p}_2$ .

Soit à la date :

$$\sigma_1(k_1) = u_1(k_1) \oplus x_2^1(k_1 - 1)$$

.

Nous pouvons noter que  $\sigma_1(k_1)$  à la même expression que la date de rafraichissement de la transition  $t_1$ , en l'absence de la place de conflit dans le cas général nous avons la date :

$$\sigma_i(k_i) = A_i(1, :) \otimes x(k_1 - 1) \oplus B_i(1, :) \otimes u_i(k_1)$$

D'après la définition de la règle PAPS, le  $k^{\text{ème}}$  jeton de  $\tilde{p}$  est à consommer par la transition le demandant le plus tôt. En supposant que chaque transition  $\check{t}_i$  a déjà consommé  $(k_i - 1)$  jetons de  $\tilde{p}$ , la transition  $\check{t}_j$  qui satisfait la condition

$$\sigma_j(k_j) = \min_{1 \leq i \leq N} \sigma_i(k_i)$$

**Remarque** La suite de cette méthode nous ramène à des équations délicates. Elle ne sera pas utilisée dans notre travail. Pour plus de détails sur cette méthode voir [6].

## 3.3 Méthode 2

### 3.3.1 Modélisation de RGETC dans l'algèbre Max-plus à base de systèmes à commutation

Une autre méthode de modélisation des RGETC dans l'algèbre  $(\text{Max}, +)$ , se base sur le fonctionnement en modes et le système de commutation (switching) .

Pour chaque mode une seule transition sera franchit au minimum d'un seul GET. Un modèle  $(\text{Max}, +)$  avec des commutateurs entre les modes est défini par le comportement dynamique d'un RGETC, en utilisant une séquence d'allocation .

**Définition 3.1 (forward commutator) :** Commutateur avant (Forward commutator) est défini par une place de conflit avec une transition d'entrée et des transitions de sorties.

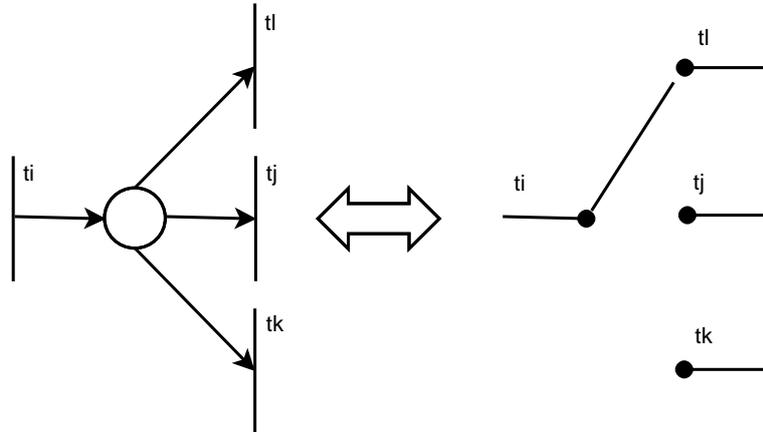


FIGURE 3.3 – Forward Commutator

**Définition 3.2 (backward commutator)** Commutateur arrière (Backward commutator) est un commutateur avec une place de conflit, plusieurs transitions d'entrées et une transition de sortie .

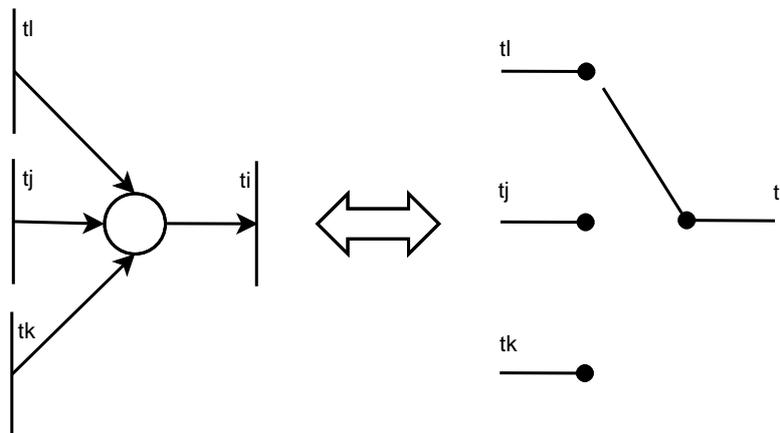


FIGURE 3.4 – Backword Commutator

**Définition 3.3** Une configuration correspond au positions des commutateurs.

$$C_1^{FW} = \{(t_i, t_j), (t_i, t_k), (t_i, t_l)\}.$$

$$C_2^{BW} = \{(t_l, t_i), (t_j, t_i), (t_k, t_i)\}.$$

Soit la figure (3.5) suivante, qui montre le passage d'une seule place en conflit avec deux transitions d'entrées et deux transitions de sorties à une configuration qui constitue de deux commutateurs en série, commutateur Forward et commutateur Backward

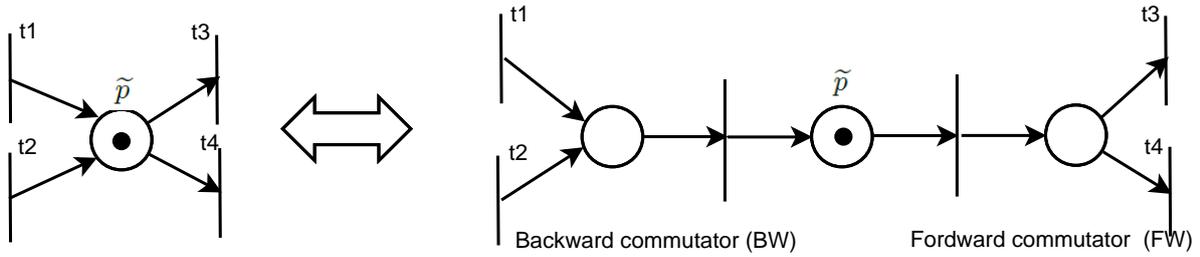


FIGURE 3.5 – Place avec commutation avant et commutation arrière

**Le fonctionnement** Un jeton arrive a la place  $\tilde{p}$ , on franchit l'une des transitions d'entrées  $t_1$  ou  $t_2$ , le jeton de  $\tilde{p}$  est consommé par le franchissement de l'une des transitions de sorties  $t_3$  ou  $t_4$ .

L'exemple de la figure (3.6) représente un RGETC qui constitue un nombre fini  $N$  de GET qui partagent une seule place de conflit  $\tilde{p}$  sauf .

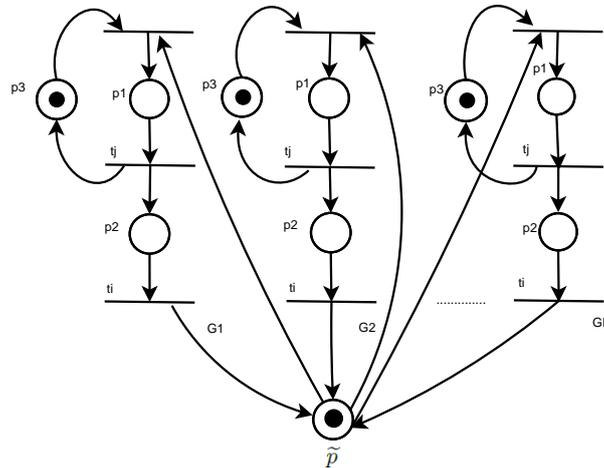


FIGURE 3.6 – Exemple d'un RGETC

— On suppose que le jeton de la place en conflit soit attribué suivant une séquence donnée  $S$  une exécution séquentielle des GET.  
par exemple  $S = G_1 G_5 G_1 \dots G_N$ .

**Définition 3.4** On appel un dateur généralisé pour une transition  $t_i$  une fonction  $x_i(k)$  non décroissante avec l'indication de la date de tire du  $k^{ème}$  exécution de réseau.

**Définition 3.5** On note un mode  $l$  dans une séquence  $S$  une exécution d'un nombre finit des GET dans un réseau.

par exemple :

Mode 1 activation de  $GET_1$ .

Mode 2 activation de GET5.

.....

Mode N activation de GETN.

— Si on dispose d'un commutateur avant  $C^{FW}$  (Forward commutator) comme montre la figure (3.3), avec  $t_j$  la seule transition d'entrée et  $t_i$  l'une des transitions de sorties.

$$x_i(k+1) = \begin{cases} \tau_i \cdot x_j(k) & , \text{ si la position du commutateur est sur } (t_i, t_j). \\ x_i(k) & , \text{ si non.} \end{cases}$$

— Si on dispose de commutateur arrière  $C^{BW}$  (Backward commutator) comme le montre la figure (3.4), on'a  $t_l$  l'une des transitions d'entrées et  $t_i$  la seul transition de sortie de ce commutateur. On auras donc

$$x_i(k+1) = \begin{cases} \tau_i \cdot x_l(k) & , \text{ si la position du commutateur sur } (t_l, t_i). \\ x_l(k) & , \text{ si non.} \end{cases}$$

**Théorème 3.1** Le comportement dynamique d'un réseau de graphes d'événements temporisés en conflit est décrit par un système linéaire hybride en commutation, la dynamique est donnée par le système d'équation suivant :

$$x(k+1) = A^l \cdot x(k) \bigoplus B^l \cdot u^l(k+1) \tag{3.11}$$

où :

$l = 1, 2, \dots, L$  est l'indice de  $l^{ieme}$  mode de fonctionnement.

$A^l \in \mathbb{R}_{max}^{n \times n}$  est la matrice d'état du mode  $l$ .

$B^l \in \mathbb{R}_{max}^{n \times m}$  est la matrice de commande du mode  $l$ .

$u^l$  les entrées de commande du mode  $l$ .

**Preuve** Chaque position du commutateur est représenté par un système d'équation Max-plus linéaire.

**Exemple** Considérons le réseau de graphes d'événements temporisés en conflits dans la figure suivante (3.7), qui est constitué de deux GETs reliés entre eux avec deux places de conflits sauf.

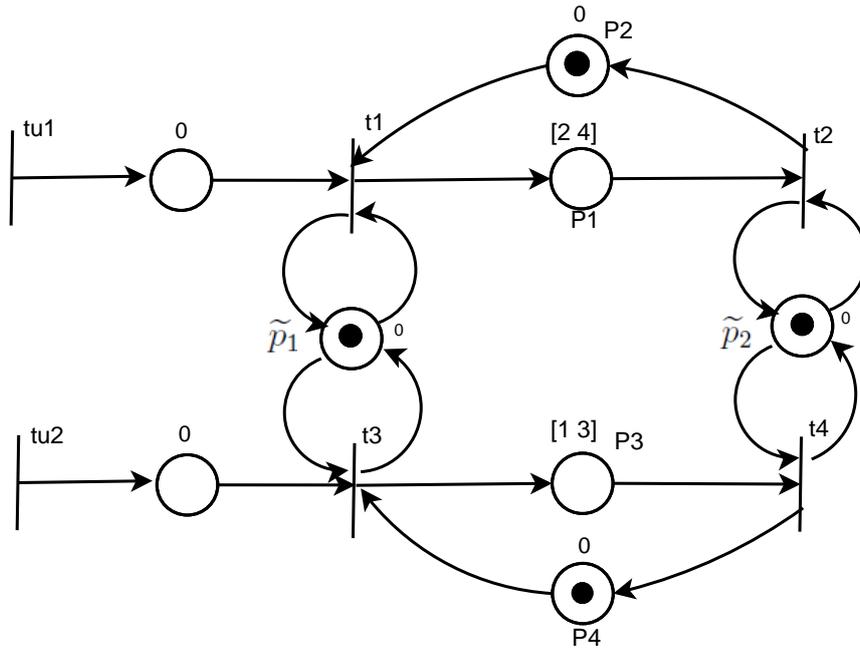


FIGURE 3.7 – Exemple d'un RGETC soumis à des contraintes temporelles

### 3.3.2 Modélisation de RGETC

Soient la séquence suivante :  $S = G_1G_2 G_1G_2 G_1G_2 G_1G_2$ .

#### Mode 1

Pour  $l = 1$  activation du réseau, les jetons de conflits  $\tilde{p}_1$  et  $\tilde{p}_2$  sont consommés par le  $GET_1$  ( le jeton de  $\tilde{p}_1$  est alloué à  $t_1$  et le jeton de  $\tilde{p}_2$  est alloué à  $t_2$  ) donc on obtient les équations suivantes :

$$\begin{cases} \Theta_1(k+1) = \Theta_2(k) \oplus \Theta_3(k) \oplus u_1(k+1) \\ \Theta_2(k+1) = 2\Theta_2(k) \oplus 2\Theta_3(k) \oplus 2u_1(k+1) \oplus \Theta_4(k) \\ \Theta_3(k+1) = \Theta_3(k) \\ \Theta_4(k+1) = \Theta_4(k) \end{cases} \quad (3.12)$$

Sa forme matricielle est donnée par :

$$\Theta(k+1) = \begin{bmatrix} \varepsilon & e & e & \varepsilon \\ \varepsilon & 2 & 2 & e \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \otimes \Theta(k) \oplus \begin{bmatrix} e & \varepsilon \\ 2 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} \otimes u^1(k+1)$$

#### Mode 2

Pour  $l = 2$  activation du réseau, les jetons de conflits  $\tilde{p}_1$  et  $\tilde{p}_2$  sont consommés par le  $GET_2$  (le jeton de  $\tilde{p}_1$  est alloué à  $t_3$  et le jeton de  $\tilde{p}_2$  est alloué à  $t_4$ ), donc on auras les

équations suivantes :

$$\begin{cases} \Theta_1(k+1) = \Theta_1(k) \\ \Theta_2(k+1) = \Theta_2(k) \\ \Theta_3(k+1) = \Theta_1(k) \oplus \Theta_4(k) \oplus u_2(k+1) \\ \Theta_4(k+1) = 1 \Theta_1(k) \oplus 1 \Theta_4(k) \oplus 1 u_2(k+1) \oplus \Theta_2(k) \end{cases} \quad (3.13)$$

Sous la forme matricielle

$$\Theta(k+1) = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & e \\ 1 & e & \varepsilon & 1 \end{bmatrix} \otimes \Theta(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & 1 \end{bmatrix} \otimes u^2(k+1)$$

### 3.4 Le comportement de RGETC comme un automate

**Définition 3.7** Un automate fini déterministe est défini par un quintuplet  $G = \{\Sigma, Q, q_0, F, \vartheta\}$ , où :

- $\Sigma$  est un ensemble fini des symboles (l'alphabet).
- $Q$  est un ensemble fini d'état.
- $q_0 \in Q$  est l'état initial.
- $F \subset Q$  est un ensemble des état finaux.
- $\vartheta$  est une fonction totale de  $Q \times \Sigma$ , appelée fonction des transitions.

Un automate fini correspond à un graphe orienté, dans lequel certains des noeuds (états) sont distingués et marqués comme initiaux ou finaux et dans lequel les arcs (transitions) sont étiquetés par des symboles.

- Nous pouvons représenter le RGETC qui suit une séquence  $S$  par un automate  $G$ , telle que les états de l'automate correspondent à un mode donné des réseaux, et les transitions de ce dernier correspondent aux commutateurs données (switching). Ce qui est représenté dans la figure suivante :

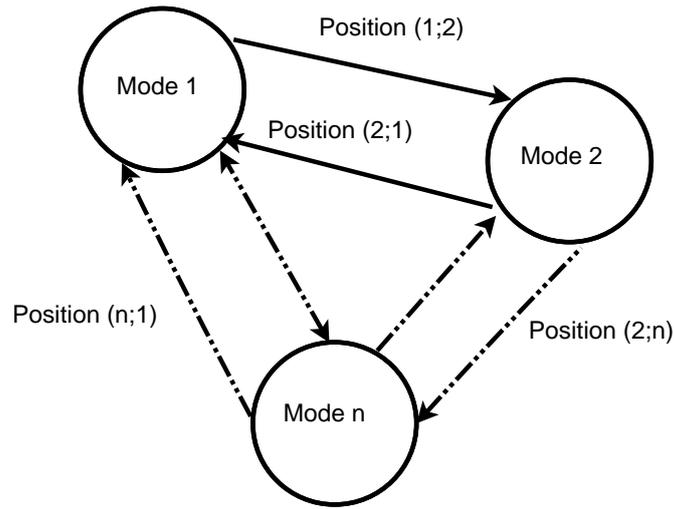


FIGURE 3.8 – Exemple d'un automate

### 3.4.1 Exemple 3.2

L'exemple suivant représente un RGETC, qui contient trois (3) unités de stocks d'informations, le principe de fonctionnement est d'utiliser un seul stock à la fois.

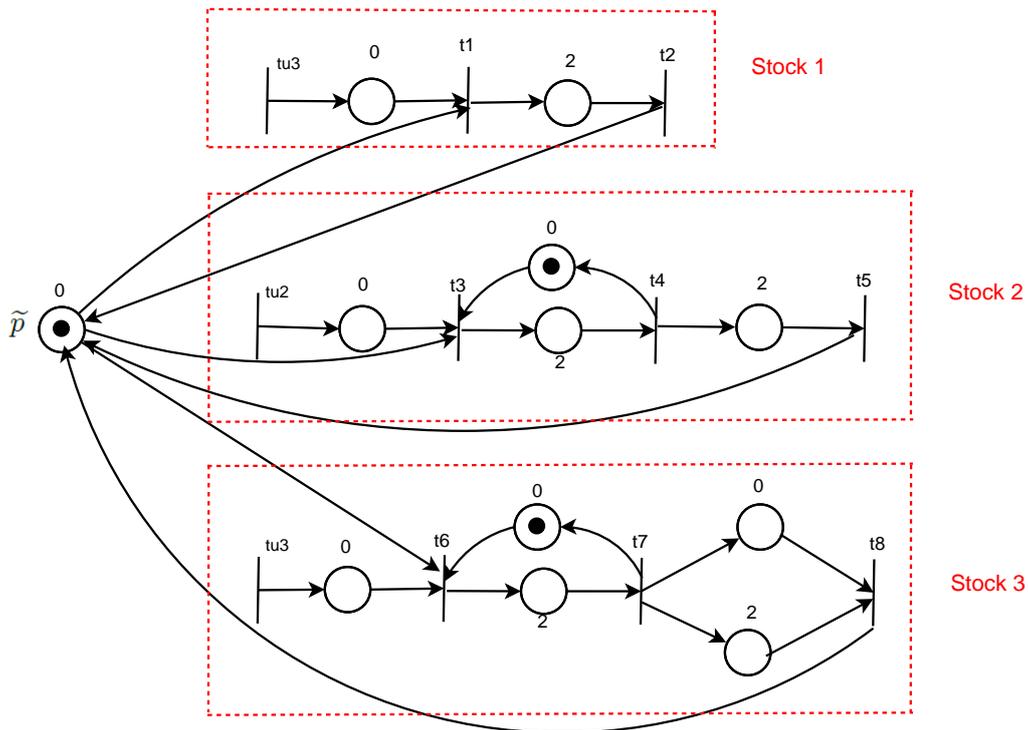


FIGURE 3.9 – Système à trois unités de stocks avec conflit

#### Le fonctionnement

- Pour  $l = 1$  (mode 1), la place du conflit  $\tilde{p}$  attribue le jeton à  $t_3$  après deux unités de temps, le jeton est récupéré avec le franchissement de  $t_4$  après  $t_5$ .

- Pour  $l = 2$  (mode 2), la place du conflit  $\tilde{p}$  attribue le jeton à  $t_1$  (exécution de  $GET_1$ ) après deux unités de temps, ce jeton est récupéré avec le franchissement de  $t_2$ .
- Pour  $l = 3$  (mode 3), la place du conflit  $\tilde{p}$  attribue le jeton à  $t_5$ , et après quatre unités de temps ce jeton est récupéré avec le franchissement de  $t_8$ .

Respectons la séquence suivante  $S = G_2G_1 G_3G_2 G_1G_3$ , et on représentant ce comportement par un automate  $G$  qui est dans la figure suivante :

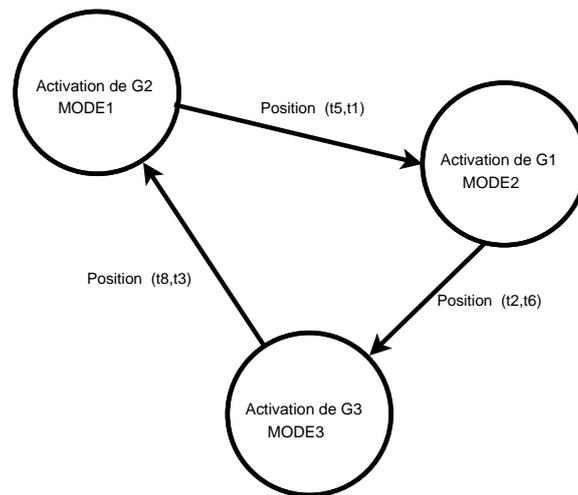


FIGURE 3.10 – Automate de trois unités de stocks

**Conclusion** Dans ce chapitre, on a proposé une méthode de modélisation par mode avec des systèmes de commutation, ou dans chaque mode on franchit au-moins un seul GET, au final on a trouvé une forme d'état équivalente qui résout le problème des réseaux de graphes d'évènements temporisés en conflit.

# Chapitre 4

## Proposition d'une loi de Commande des RGETC dans l'algèbre Max-plus

### 4.1 Introduction

Le problème de commande des systèmes à événements discrets est basé sur l'élaboration de lois de commande efficaces, pour garantir le respects des spécifications du cahier des charges. Dans le cas ou ces spécifications concernent le respects de contraintes temporelles, des lois de commande ont été synthétisées par les auteurs dans [1] [2] pour une sous classe de SED, qui est les graphes d'événements temporisés. Dans ce chapitre, le problème de commande des RGETC sous contraintes temporelles est posé, pour résoudre ce problème, on utilise la modélisation proposé par [8] pour la quelle une séquence de franchissement est définie. Dans chaque mode, un certain nombre de contraintes temporelles sont à satisfaire. On synthétise des lois de commande sous la forme  $u(k+1) = F_r x(k)$ , où  $r$  désigne le mode numéro qui permet de satisfaire dans chaque mode les contraintes temporelles.

### 4.2 Représentation linéaire du GET dans l'algèbre $(\text{Max}, +)$

Le comportement dynamique d'un GET peut être exprimé sous forme d'équation linéaire dans le dioïde  $(\text{Max-plus})$ , tel que on associe pour chaque transition du GET une fonction de la variable  $k \in N$ ,  $k \geq 1$ , cette variable correspond à la date du  $k^{\text{ème}}$  franchissement de la transition du GET considéré, cette fonction est appelée fonction dateur  $\theta(k) \in \bar{\mathbb{R}}_{\text{max}}^n$ . Les dateurs correspondants aux transitions sources sont les composantes des dateurs  $u(k) \in \bar{\mathbb{R}}_{\text{max}}^m$ . La représentation d'état  $(\text{Max}, +)$  des GETs est donnée par l'équation suivante :

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \quad (4.1)$$

**Exemple 4.1** Considérons le graphe d'événements temporisé représenté dans la figure 4.1 suivante, on associe pour chaque transition  $t_{u_1}, t_1, t_2$  des fonctions dateurs  $u(k), x_1(k), x_2(k)$ .

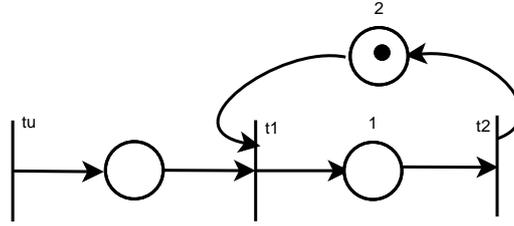


FIGURE 4.1 – Graphe d'événements temporisé

Le comportement considéré est décrit par les équations suivantes dans l'algèbre  $(\text{Max}, +)$  :

$$\begin{cases} x_1(k) = 2x_2(k-1) \oplus u_1(k) \\ x_2(k) = 1x_1(k) \end{cases}$$

$$x(k) = \begin{bmatrix} \varepsilon & \varepsilon \\ 1 & \varepsilon \end{bmatrix} \otimes x(k) \oplus \begin{bmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{bmatrix} \otimes x(k-1) \oplus \begin{bmatrix} e \\ \varepsilon \end{bmatrix} \otimes u(k) \quad (4.2)$$

En appliquant le **Théorème 1.1**, on obtient :

$$A = A_0^* A_1 = \begin{bmatrix} \varepsilon & 2 \\ \varepsilon & 3 \end{bmatrix}, B = A_0^* B_1 = \begin{bmatrix} e \\ 1 \end{bmatrix}$$

Ainsi, la forme explicite s'écrit sous la forme suivante :

$$x(k) = \begin{bmatrix} \varepsilon & 2 \\ \varepsilon & 3 \end{bmatrix} \otimes x(k-1) \oplus \begin{bmatrix} e \\ 1 \end{bmatrix} \otimes u(k) \quad (4.3)$$

## 4.2.1 Synthèse de la commande en présence de contraintes temporelles

### 4.2.1.1 Formalisation de la commande

Étant donnée un RGETC  $G$ , dont l'évolution est donnée par l'équation suivante :

$$x(k) = A^l x(k-1) \oplus B^l u(k) \quad (4.4)$$

$$l \in \{1, 2, \dots, L\}$$

où :

$$x \in \bar{\mathbb{R}}_{max}^n, A^l \in \bar{\mathbb{R}}_{max}^{n \times n}, B^l \in \bar{\mathbb{R}}_{max}^{n \times p}, k \in \mathbb{N}.$$

et soient les contraintes temporelles à satisfaire  $\zeta^l, l \in \{1, 2, \dots, L\}$  de la forme :

$$\zeta^l = \left\{ x \in \bar{\mathbb{R}}_{max}^n, C^l \otimes x \leq H^l \otimes x \right\} \quad (4.5)$$

où :

$$C^l, H^l \in \bar{\mathbb{R}}_{max}^{s_i \times n}.$$

Le problème revient donc à trouver des lois de commande  $u$  tel que

$$\begin{cases} x(k) = A^l x(k-1) \oplus B^l u(k) \\ C^l \otimes x(k) \leq H^l \otimes x(k) \\ l \in \{1, 2, \dots, L\} \end{cases}$$

#### 4.2.1.2 Commande par retour d'état

On propose une loi de commande de la forme

$$u(k+1) = F_i \otimes x(k). \quad i \in \{1, 2, \dots, N\}.$$

où :

$F_i$  sont des gains de retour d'état à calculer pour satisfaire les contraintes temporelles.

Supposons dans un premier temps que chaque GET,  $G_i$  possède une seule entrée et qu'il est soumis à une seule contrainte temporelle (sur une place).

#### 4.2.2 Commande de RGETC qui partage une place en conflit sauf

Soit la figure suivante (4.2) qui représente un RGETC, avec une place en conflit  $\tilde{p}$  sauf

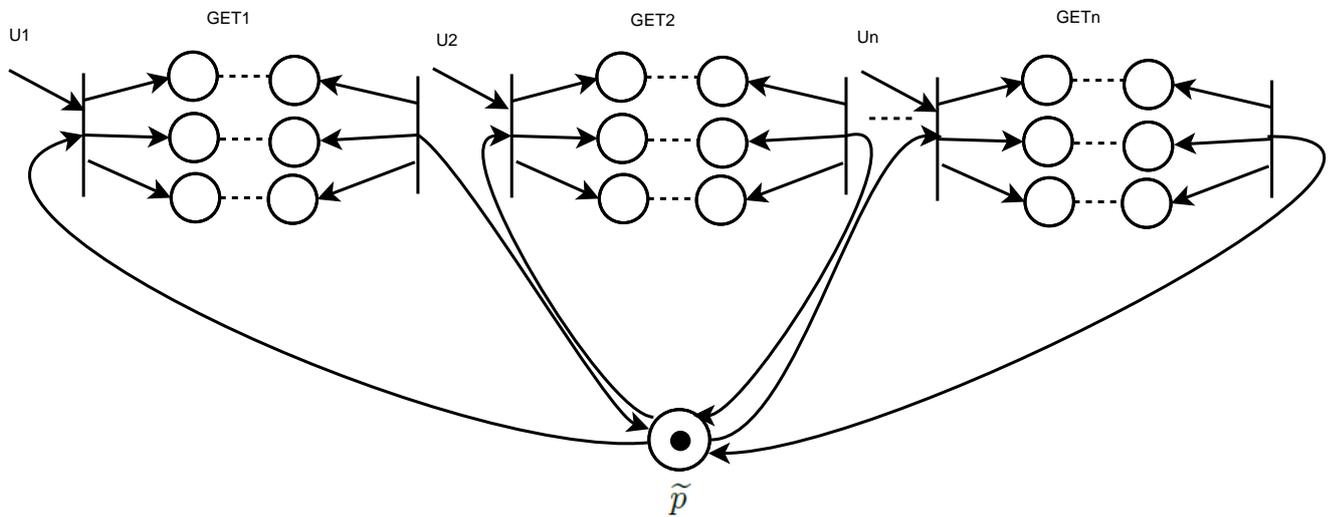


FIGURE 4.2 – Exemple de conflit

Chaque GET de ce réseau contient une seule commande  $tu_i$ , et supposons qu'il contient au-moins une seule contrainte de temps à respecter.

Soit l'évolution de la séquence  $S$ , pour chaque  $l^{ème}$  mode on peut s'écrire sous forme :

$$x(k+1) = A^l .x(k) \oplus B^l .u^l(k) \tag{4.6}$$

ou

$A^l \in \bar{\mathbb{R}}_{max}^{n \times n}$  est la matrice d'état du mode.

$B^l \in \bar{\mathbb{R}}_{max}^{n \times m}$  est la matrice de commande du mode.

$u^l \in \bar{\mathbb{R}}_{max}^{n \times p}$  est les entrées de commande du mode.

La  $i^{\text{ème}}$  composante de l'équation (4.6) est donnée par :

$$x_i(k+1) = \bigoplus_{\delta=1}^N A_{i\delta}^l x_\delta(k) \bigoplus_{\delta=1}^Z (B^l)_{i\delta} u^l(k+1) \quad (4.7)$$

ou

$Z$  : est le nombre de transitions sources reliée au réseau de graphes d'événements temporisés en conflit.

Notre objectif est de chercher une loi de commande pour satisfaire les contraintes de temps imposées sur chaque GET dans chaque mode  $l$ .

Les contraintes temporelles imposées sur chaque GET, peut se traduire sous la forme d'équation suivante :

$$x_i(k+1) \leq \tau_{max}^l x_j(k+1) \quad (4.8)$$

#### Hypothèse 4.1

Il existe dans chaque GET d'un RGETC au-moins un chemin  $\rho_i$  qui relie la transition source  $tu^l$  (commande) à la transition d'entrée  $t_j$  sous contrainte temporelle.

#### Hypothèse 4.2

Chaque transition du réseau est franchis dès qu'elle est franchissable (franchissement au plus tôt).

— Du fait qu'il existe un chemin  $\rho_i$  reliant la transition  $tu^l$  à celle de la  $t_j$ , et à partir de la définition des fonctions dateurs des transitions  $t_j$ . La relation entre la transition source et celle d'entrée est donnée par :

$$x_j(k+1) \geq \bigoplus_{\delta=1}^Z ((A^l)^{m_\rho} B^l)_{j\delta} u^l(k+1 - m_\rho) \quad (4.9)$$

Dans notre travail, considérons que le marquage initiale du chemin  $\rho$  est nul ( $m_\rho = 0$ ), en remplaçant dans l'équation (4.9), on aura

$$x_j(k+1) \geq \bigoplus_{\delta=1}^Z (B^l)_{j\delta} u^l(k+1) \quad (4.10)$$

Remplaçons l'équation (4.7) dans (4.8) on obtient :

$$\bigoplus_{\delta=1}^N A_{i\delta}^l x_\delta(k) \bigoplus_{\delta=1}^Z (B^l)_{i\delta} u^l(k+1) \leq \tau_{max}^l x_j(k+1) \quad (4.11)$$

En remplaçant l'inéquation (4.10) par son expression dans (4.11)

$$\bigoplus_{\delta=1}^N A_{i\delta}^l x_\delta(k) \bigoplus \bigoplus_{\delta=1}^Z (B^l)_{i\delta} u^l(k+1) \leq \bigoplus_{\delta=1}^Z \tau_{max}^l B_{j\delta}^l u(k+1) \quad (4.12)$$

Alors la contrainte temporelle est satisfaite si les deux inéquations suivantes sont vérifiées :

$$\bigoplus_{\delta=1}^N A_{i\delta}^l x_\delta(k) \leq \bigoplus_{\delta=1}^Z \tau_{max}^l (B^l)_{j\delta} u^l(k+1) \quad (4.13)$$

$$\bigoplus_{\delta=1}^Z (B^l)_{i\delta} u^l(k+1) \leq \bigoplus_{\delta=1}^Z \tau_{max}^l (B^l)_{j\delta} u^l(k+1) \quad (4.14)$$

Les deux conditions (4.13) et (4.14) peuvent s'écrire comme suit :

$$u^l(k+1) \geq \bigoplus_{\delta=1}^N [(A^l)_{i\delta} - \tau_{max}^l - B_{j\delta}^l] x_\delta(k) \quad (4.15)$$

$$\bigoplus_{\delta=1}^Z (B^l)_{i\delta} u^l(k+1) \leq \bigoplus_{\delta=1}^Z \tau_{max}^l (B^l)_{j\delta} u^l(k+1) \quad (4.16)$$

**Théorème 4.2** Considérons un RGETC soumis à des contraintes temporelles de la forme (4.8) et qu'il existe un chemin  $\rho_i$ , reliant la transition de commande  $tu_i^l$  à la transition d'entrée  $t_j$  sous contraintes, la loi de commande est donnée par :

$$u^l(k+1) = F^l x(k)$$

Avec  $F^l = \bigoplus_{\delta=1}^N ((A^l)_{i\delta} - \tau_{max}^l - B_{i\delta}^l)$

si la condition suivante est vérifiée :

$$\bigoplus_{\delta=1}^Z (B^l)_{i\delta} \leq \bigoplus_{\delta=1}^Z (\tau_{max}^l (B^l)_{j\delta}) \quad (4.17)$$

**Démonstration 4.2** Claire de ce qui précède.

### 4.3 Application à la gestion d'un croisement de lignes ferroviaires

Nous nous intéressons à un système de croisement des rails ferroviaires, qui se compose de trois lignes unidirectionnelles notées : L1, L2 et L3. On considère que chaque ligne comporte trois feux de signalisation, notés comme suit :

- L1 : a  $\rightarrow$  a'  $\rightarrow$  b,

- L2 :  $c \rightarrow d \rightarrow e$ ,
- L3 :  $f \rightarrow f' \rightarrow g$ .

Un feu de signalisation permet d'autoriser ou d'interdire a un train de franchir ce feu et de parcourir le tronçon en aval jusqu'au feu suivant. Par exemple, le train qui empreinte la ligne unidirectionnelle L2 va devoir franchir trois feux selon l'ordre suivant : c, d puis e. Le croisement ferroviaire comporte alors 9 feux de signalisation : a, a', b, c, d, e, f, f' et g auxquels on associe les durées respectives : 2, 1, 2, 2, 1, 1, 1, 1 et 2 unités de temps.

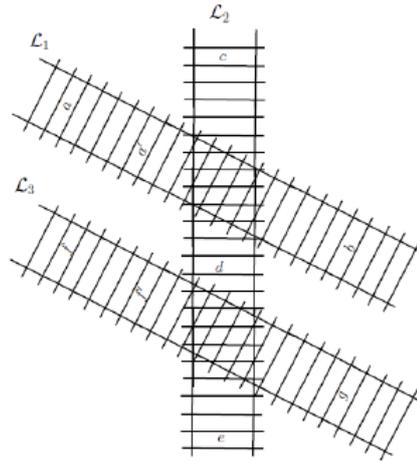


FIGURE 4.3 – Croisement de 3 rails ferroviaires

Une telle durée de passage traduit le temps nécessaire au train, s'il y est autorisé, pour traverser le feu et parcourir le tronçon jusqu'au prochain feu. Par exemple, le train traversant la ligne L1 met 2 unités de temps a franchir le feu a et arriver au feu a'. Afin d'éviter les collisions, le tronçon [cd] de la ligne L2 ne pourra pas être traversé si un train se présente sur la ligne L1 (c'est-à-dire, si un train est présent sur le tronçon [aa'] ou sur le tronçon [a'b]). De même, le tronçon [de] de la ligne L2 sera fermé si un train est présent sur la ligne L3. Réciproquement, la ligne L1 (respectivement L3) n'est pas autorisée quand un train passe sur le tronçon [cd] (respectivement [de]) de la ligne L2. D'autre part, les feux sont utilisés pour garantir la présence d'un seul train sur les tronçons de lignes. Plus précisément, le feu a (respectivement f) autorise le passage d'un train à la condition qu'aucun autre train ne soit présent sur les tronçons [aa'] et [a'b] (respectivement [ff'] et [f'g]). Sur la ligne L2, un seul train est autorisé par tronçon.

Sur la ligne L2, un seul train est autorisé par tronçon. Cette gestion des trains sur le croisement est modélisé par le réseau de Pétri de la figure 4.4

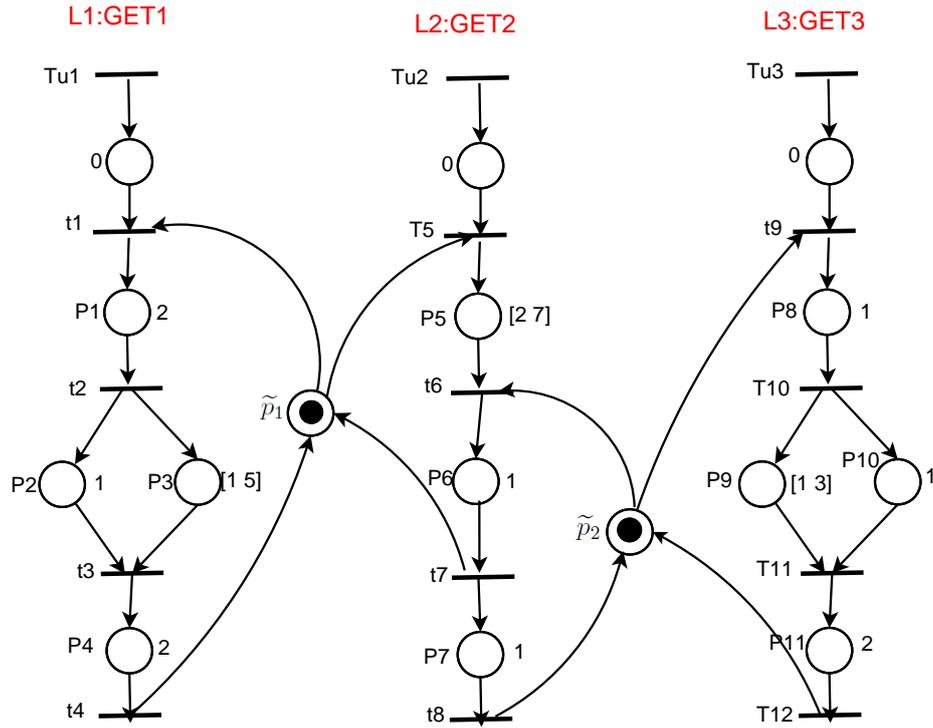


FIGURE 4.4 – RDP  $P_{\text{temporisé}}$  de Croisement de 3 rails ferroviaires

### 4.3.1 Modélisation de RGETC

Pour représenter le comportement dynamique du réseau de graphes d'évènements temporisés en conflits. Nous associons à chaque transition  $t_i$ , avec  $i = 1$  à  $12$ , une fonction dateur  $x_i(k)$ , qui représente le temps de franchissement de  $k^{\text{ème}}$  jeton. La fonction dateur attribuée à la transition d'entrée (la commande) est notée  $u(k)$ . Nous considérons un franchissement au plus tôt, c'est-à-dire que tout transitions validée est immédiatement franchie. soit la séquence  $S = l_1 l_2$  tel que ( $l_1 = L_1 L_3$ ) dans le mode 1 on franchit le  $GET_1$  et  $GET_3$  et pour le mode 2 on franchit le  $GET_2$  ( $l_2 = L_2$ ).

- Pour le (mode1)  $l = 1$  activation du réseau : les jetons des conflits  $\tilde{p}_1$  et  $\tilde{p}_2$  sont consommés successivement par l'activation  $GET_1$  et l'activation de  $GET_3$
- Pour le (mode2)  $l = 2$  activation du réseau : les jetons des conflits  $\tilde{p}_1$  et  $\tilde{p}_2$  sont consommés par l'activation  $GET_2$

#### 4.3.1.1 Mode 1

**Activation des deux GETs ( $GET_1$  et  $GET_3$ ) en parallèle** On obtient les équations suivantes :

$$\left\{ \begin{array}{l} x_1(k+1) = x_7(k) \oplus u_1^1 k + 1 \\ x_2(k+1) = 2.x_7(k) \oplus 2u_1^1 k + 1 \\ x_3(k+1) = 3.x_7(k) \oplus 3u_1^1 k + 1 \\ x_4(k+1) = 5.x_7(k) \oplus 5u_1^1 k + 1 \\ x_5(k+1) = 5.x_5(k) \\ x_6(k+1) = 5.x_6(k) \\ x_7(k+1) = 5.x_7(k) \\ x_8(k+1) = 5.x_8(k) \\ x_9(k+1) = x_8(k) \oplus u_3^1 k + 1 \\ x_{10}(k+1) = 1.x_8(k) \oplus 1u_3^1 k + 1 \\ x_{11}(k+1) = 2.x_8(k) \oplus 2u_3^1 k + 1 \\ x_{12}(k+1) = 4.x_8(k) \oplus 4u_3^1 k + 1 \end{array} \right.$$

Sous forme matricielle

$$A^1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \quad B^1 = \begin{bmatrix} e & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & 4 \end{bmatrix}$$

L'équation d'état est donnée par :

$$x(k+1) = A^l x(k) \oplus B^l u^l(k) \tag{4.18}$$

Remplaçons pour l= 1, on obtient :

$$x(k+1) = A^1 x(k) \oplus B^1 u^1(k) \tag{4.19}$$

$$x(k+1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot x(k) \oplus \begin{bmatrix} e & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & 4 \end{bmatrix} \cdot u^1(k) \quad (4.20)$$

### 4.3.2 Mode 2

Activation de  $GET_2$ , implique le passage d'un train par le chemin (L2) On obtient les équations suivantes

$$\left\{ \begin{array}{l} x_1(k+1) = x_1(k) \\ x_2(k+1) = x_2(k) \\ x_3(k+1) = x_3(k) \\ x_4(k+1) = x_4(k) \\ x_5(k+1) = x_4(k) \oplus u_2^2 k + 1 \\ x_6(k+1) = 2 \cdot x_4(k) \oplus 2u_2^2 k + 1 \oplus x_{12}(k) \\ x_7(k+1) = 3 \cdot x_4(k) \oplus 3u_2^2 k + 1 \oplus 1 \cdot x_{12}(k) \\ x_8(k+1) = 4 \cdot x_4(k) \oplus 4u_2^2 k + 1 \oplus 2 \cdot x_{12}(k) \\ x_9(k+1) = x_9(k) \\ x_{10}(k+1) = x_8(k) \\ x_{11}(k+1) = x_{11}(k) \\ x_{12}(k+1) = x_{12}(k) \end{array} \right.$$

Sa forme matricielle est donnée par :

$$A^2 = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & e \end{bmatrix}, B^2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & 2 & \varepsilon \\ \varepsilon & 3 & \varepsilon \\ \varepsilon & 4 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

L'équation d'état est donnée par :

Pour  $l = 2$

$$x(k + 1) = A^2 x(k) \oplus B^2 u^2(k) \tag{4.21}$$

et on obtient

$$x(k + 1) = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & e \end{bmatrix} x(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & 2 & \varepsilon \\ \varepsilon & 3 & \varepsilon \\ \varepsilon & 4 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} u^2(k)$$

### 4.3.3 La Commande de RGETC

Dans cette sous section, nous cherchons à trouver des lois de commandes qui satisfont les contraintes temporelles imposées sur les places  $p_2$ ,  $p_5$  et  $p_9$  des  $GET_1$ ,  $GET_2$  et  $GET_3$  respectivement de réseau de graphes d'événements temporisés en conflits dans la figure 4.4.

Notre objectif est de calculer une loi de commande  $u^l(k + 1)$ , Pour chaque mode qui satisfait les contraintes imposées. Le système possède trois transitions de commande

(chaque GET possède une transition source), on applique **le théorème 3.1** pour garantir les contraintes imposés sur le système de croisement.

- Le chemin  $\rho_1$  qui relie  $t_2$  à  $t_3$  est soumis à une contrainte de temps sur la place  $p_2$  du  $GET_1$ , qui peut s'écrire sous la forme d'inéquation suivante :

$$x_3(k+1) \leq 7x_2(k+1) \quad (4.22)$$

- Le chemin  $\rho_2$  qui relie  $t_5$  à  $t_6$  est soumis à une contrainte de temps sur la place  $p_5$  du  $GET_2$ , cette contrainte est donné par l'inéquation suivante :

$$x_6(k+1) \leq 7x_5(k+1) \quad (4.23)$$

- Le chemin  $\rho_3$  qui relie  $t_{10}$  à  $t_{11}$  est soumis à une contrainte de temps sur la place  $p_9$  du  $GET_2$ , cette contrainte est donné par l'inéquation suivante :

$$x_{11}(k+1) \leq 10x_{10}(k+1) \quad (4.24)$$

#### 4.3.3.1 Mode 1 ( Franchissement de $GET_1$ et $GET_3$ )

Cherchons à trouver une loi de commande qui satisfait les deux contraintes imposées sur les chemins  $\rho_1$  et  $\rho_3$  .

- Du fait qu'il existe un chemin  $\rho_1$  qui relie la transition  $tu_1$  à la transition  $t_2$  sous contrainte du temps, la temporisation minimal de ce chemin est  $\tau_p^{min} = 1$ , et la temporisation maximal  $\tau_p^{max} = 5$ ,  $j = 2$ ,  $i = 3$ .

$$B_{2\delta}^1 = \begin{bmatrix} 2 & \varepsilon & \varepsilon \end{bmatrix}, B_{3\delta}^1 = \begin{bmatrix} 3 & \varepsilon & \varepsilon \end{bmatrix}$$

$$A_{3\delta}^1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

On vérifie que :

$$\bigoplus_{\delta=1}^N B_{3\delta}^1 \leq \bigoplus_{\delta=1}^N \tau_{p_2}^{max} B_{2\delta}^1$$

La condition est bien vérifiée, alors il existe une loi de commande qui satisfait les contraintes du temps imposées sur  $GET_1$ , cette loi de commande est donnée par l'équation suivante :

$$u_1^1(k+1) = \bigoplus_{\delta=1}^N ((A^1)_{3\delta} - \tau_{p_2}^{max} - B_{2\delta}^1) x_\delta(k) \quad (4.25)$$

En remplaçant, on obtient

$$u_1^1(k+1) = \bigoplus_{\delta=1}^N ((A^3)_{3\delta} - 5 - 3) x_\delta(k)$$

On remplaçons le terme  $((A^3)_{3\delta} - 5 - 3) \leq 0$  par l'élément neutre  $e$ . Alors le contrôleur est donnée par l'équation suivante :

$$u_1^1(k+1) = e x_7(k) \quad (4.26)$$

— Du fait qu'il existe un chemin  $\rho_3$  qui relie la transition  $tu_3$  à la transition  $t_9$  sous contrainte de temps. La temporisation minimal de ce chemin est  $\tau_{p_9}^{min} = 1$  et la maximal  $\tau_{p_9}^{max} = 3, j = 10, i = 11$ .

$$B_{10\delta}^1 = \begin{bmatrix} \varepsilon & \varepsilon & 1 \end{bmatrix}, B_{11\delta}^1 = \begin{bmatrix} \varepsilon & \varepsilon & 2 \end{bmatrix}$$

$$A_{11\delta}^1 = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

Vérifions la condition du **Théorème 4.2**

$$\bigoplus_{\delta=1}^N B_{11\delta}^1 \leq \bigoplus_{\delta=1}^N \tau_{p_9}^{max} B_{10\delta}^1$$

La condition est bien vérifiée, alors il existe une loi de commande qui satisfait les contraintes de temps imposées sur  $GET_3$ , cette loi de commande est donnée par l'équation suivante :

$$u_3^1(k+1) = \bigoplus_{\delta=1}^N ((A^1)_{10\delta} - \tau_{p_9}^{max} - B_{9\delta}^1) x_\delta(k) \quad (4.27)$$

On remplaçons et on obtient

$$u_3^1(k+1) = \bigoplus_{\delta=1}^N ((A^1)_{10\delta} - 3 - 1) x_\delta(k)$$

On remplaçons le terme  $((A^1)_{10\delta} - 3 - 1) \leq 0$  par l'élément neutre  $e$ . Alors le contrôleur est donnée par l'équation suivante :

$$u_3^1(k+1) = e x_8(k) \quad (4.28)$$

Pour ce premier mode, on obtient la loi de commande suivante :

$$u^1(k+1) = \begin{bmatrix} u_1^1(k+1) \\ u_3^1(k+1) \end{bmatrix} = \begin{bmatrix} e x_7(k) \\ e x_8(k) \end{bmatrix} \quad (4.29)$$

#### 4.3.3.2 Mode 2 ( Franchissement du $GET_2$ )

On cherche a trouver une loi de commande qui satisfait la contrainte imposée sur le chemin  $\rho_2$ .

— Du fait qu'il existe un chemin  $\rho_2$  qui relie la transition  $tu_2$  à la transition  $t_5$  sous contrainte de temps. La temporisation minimal de ce chemin est  $\tau_{p_5}^{min} = 1$  et la temporisation maximal  $\tau_{p_5}^{max} = 5, j = 5, i = 6$ .

$$B_{5\delta}^2 = \begin{bmatrix} \varepsilon & e & \varepsilon \end{bmatrix}, B_{6\delta}^2 = \begin{bmatrix} \varepsilon & 2 & \varepsilon \end{bmatrix}$$

$$A_{6\delta}^2 = \left[ \varepsilon \quad \varepsilon \quad \varepsilon \quad 2 \quad \varepsilon \quad e \right]$$

Vérifions la condition du **Théorème 4.2**

$$\bigoplus_{\delta=1}^N B_{6\delta}^2 \leq \bigoplus_{\delta=1}^N \tau p_5^{max} B_{5\delta}^2$$

La condition est bien vérifiée, alors il existe une loi de commande qui satisfait les contraintes de temps imposées sur  $GET_2$ , cette loi de commande est donnée par l'équation suivante :

$$u_2^2(k+1) = \bigoplus_{\delta=1}^N ((A^2)_{6\delta} - \tau p_5^{max} - B_{5\delta}^2) x_\delta(k) \quad (4.30)$$

On remplaçons et on obtient

$$u_2^2(k+1) = \bigoplus_{\delta=1}^N ((A^2)_{3\delta} - 5 - B_{5\delta}^2) x_\delta(k)$$

On remplaçons le terme  $((A^2)_{3\delta} - 5 - B_{5\delta}^2) \leq 0$  par l'élément neutre  $e$ . Alors le contrôleur est donnée par l'équation suivante :

$$u_2^2(k+1) = e x_4(k) \bigoplus e x_{12}(k) \quad (4.31)$$

Pour le deuxième mode, on obtient la loi de commande suivante :

$$u^2(k+1) = \left[ u_2^2(k+1) \right] = \left[ e x_4(k) \bigoplus e x_{12}(k) \right]$$

Donc les lois de commande est donnée par :

$$u^l(k+1) = \begin{bmatrix} u^1(k+1) \\ u^2(k+1) \end{bmatrix} = \begin{bmatrix} u_1^1(k+1) \\ u_3^1(k+1) \\ u_2^2(k+1) \end{bmatrix} = \begin{bmatrix} e x_7(k) \\ e x_8(k) \\ e x_4(k) \bigoplus e x_{12}(k) \end{bmatrix}$$

Ces lois de commande peuvent être représentées par des places marquées et temporisées connectées au modèle de croisement de lignes ferroviaires (place représentée par un double cercle sur la figure (4.5) suivante)

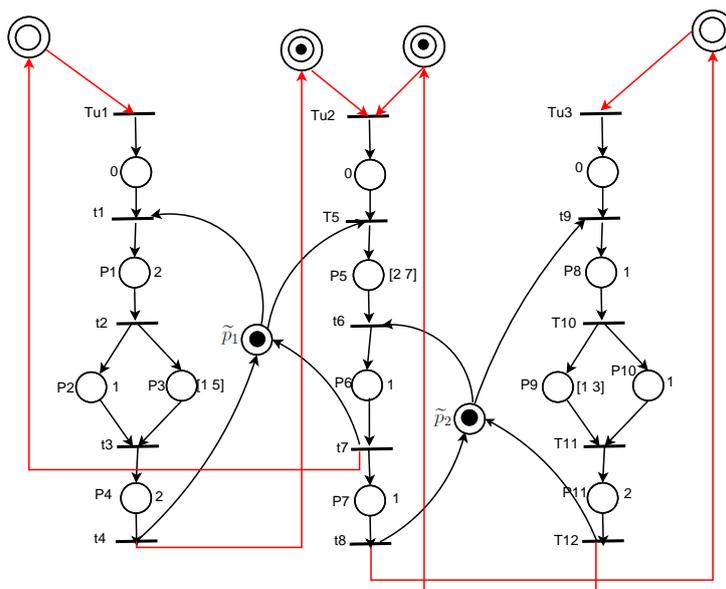


FIGURE 4.5 – Application de la commande sur RGETC

Remarque

Pour respecter la séquence il faut que :

$$M_1 + M_2 \leq 1 \text{ et } M_3 + M_4 \leq 1.$$

**Conclusion** Une approche de commande des réseaux de graphes d'événements temporisés en conflits (RGETC), dans l'algèbre  $(\text{Max}, +)$  est proposée dans ce chapitre. Cette méthode est basée sur le fonctionnement en commutation, sous l'évolution d'une séquence proposée.

# Conclusion générale

Ce travail a eu pour objectif, la proposition de lois de commande dans l'algèbre  $(\text{Max}, +)$  qui satisfait des contraintes de temps imposés sur des réseaux de graphes d'événements temporisés en conflit. En associant à chaque transitions un dateur généralisé, un système d'équations non linéaire est obtenu. Plus précisément, il s'agit d'un système Max-plus hybride. Des conditions nécessaires et suffisantes permettant de vérifier si des lois de commande par retour d'état (de la forme  $u(k) = F_i \cdot x(k - 1)$ ) existent et qui assure la satisfaction des contraintes de temps. Les lois de commande synthétisées peuvent être représentées par des places de contrôles connectées aux réseaux de Petri initiale, afin de garantir le bon fonctionnement du système. Par la suite, nous avons appliqués le résultat de notre approche à la commande d'un croisement de lignes ferroviaires. Les lois de commande calculées sont des feedback causaux qui peuvent être représentées par des places de contrôles connectées aux modèles de croisement de lignes ferroviaires. Plusieurs perspectives peuvent être envisagés par la suite de ce travail :

- Considérer d'autres hypothèses, le cas multivariable, chaque GET contient plusieurs transitions sources.
- Envisager des lois de commande pour les contraintes de marquages.

# Bibliographie

- [1] S. Amari, Commande des graphes d'événements temporisés sous contraintes temporelles, Thèse de doctorat, Ecole centrale de Nantes, 2005.
- [2] K. Tebani, Commande temps réel d'une classe de systèmes à événements discrets sous contraintes par l'algèbre des dioïdes, Thèse doctorat, ummto, Tizi-ouzou, 2017.
- [3] Ramadge, P.J, Wonham W.M, Modular feedback logic for discrete event systeme, SIAM journal of Control and Optimization,1987.
- [4] Yamalidou, k. Moody, J. Lemmon, M. Antsaklis, Feedback control petri net based on place invariant automatica (1996).
- [5] R. Kumar, V.K.Garg, Modeling and control of logical discrets event systèmes Kluwer. Academic. Publishers, USA. (1995).
- [6] B. Addad, S.Amari, J-J Lesage, Modélisation des réseaux de Graphe D'évènements Temporisé avec conflit dans l'algèbre (Max-Plus ), [2010], Cachan,France.2010.
- [7] R.Cuminghamme-Green, Minimax algèbras, dans lectures notes in economic and mathematical system, Spring-Verlag. 1979.
- [8] L.Mammeri-Beldi, R.Kara, S.Amari, Modelling and analysis of descrets event systems with shared ressources in (Max,+) algebra, 2018..
- [9] Stéphane Gaubert, Introduction aux systèmes dynamiques à évènement discret, 1999, France.
- [10] N.Bennis, Notes de cour sur les réseaux de petri, théorie et propriétés et application.
- [11] Hillon. H, Proth.J-M, performance evaluation of jobshop systeme using timed event graphe, IEEE, Transaction on automatic control.1989.
- [12] Trouillet. B, l'évaluation du comportement logic des systèmes de production manufacturière par méthodes exacte, Thèse de doctorat, université Lille 1, 2003.
- [13] Cohen, G, Dubois. D, Quadrat. J-P, Viot. M, Analyse du comportement périodique des systèmes de production par la théorie des dioïde, Rapport de recherche 191, INRIA, Le Chesnay, France, 1983.
- [14] Amari. S, I. Demongodin, J-J. Loiseau, Sizing, cycle time and plant control using dioïd algebra. Chapter 6 in Supply Chain Optimization, Series Applied Optimization. (2004).

- [15] Baccelli. B, Cohen. G, Gaujal. B, Recursive equations and basic properties of timed Petri nets, Discrete Event Dynamic Systems, Theory and Applications, vol.1(4), pp.145-439, (1992).

## Résumé

Une classe de réseaux de Petri (RDP), permet de modéliser des phénomènes de synchronisations et de parallélisme et non le partage de ressource est les graphes d'événement temporisés (GET). Une approche de modélisation de réseaux de graphes d'événements temporisés en conflits (RGETC) est proposée dans [Mammeri et Al], nous développons des équations régissant l'évolution des RGETC. Nous considérons une séquence à respecter par mode, ou dans chaque mode au-moins un seul GET, pour aboutir à une proposition d'une loi de commande des RGETC dans l'algèbre (Max-plus), qui satisfait des contraintes de temps imposés sur les RGETC, des conditions nécessaires et suffisantes permettant de vérifier si la loi de commande par retour d'état  $u(k) = F_i \cdot x(k - 1)$ , existent et qui assure la satisfaction des contraintes de temps. La loi de commande synthétisée peut être représentée par des places de contrôles connectées au réseau initial. Par la suite, nous avons appliqué le résultat de notre approche à la commande d'un croisement de lignes ferroviaires.

## Mots clés

Réseau de graphes d'événements temporisés en conflits, système à événements discrets, commande, algèbre des dioïdes.