

*RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE*  
*Ministère de l'Enseignement Supérieur et de la Recherche Scientifique*

**UNIVERSITÉ MOULOU D MAMMERI DE TIZI-OUZOU**  
**FACULTÉ DE GENIE ELECTRIQUE ET D' INFORMATIQUE**  
**DÉPARTEMENT D'AUTOMATIQUE**

## **THÈSE**

Présentée pour obtenir le diplôme de

**Doctorat 3<sup>ème</sup> Cycle LMD**

Spécialité : AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE

Par :

**ABERKANE SOFIANE**

**THEME**

**Contribution à la commande**  
**sous contraintes d'une classe de systèmes**  
**à événements discrets.**

SOUTENUE DEVANT LE JURY :

Président	DJENNOUNE Saïd	Professeur, Université Mouloud Mammeri de Tizi Ouzou
Rapporteur	KARA Redouane	Professeur, Université Mouloud Mammeri de Tizi Ouzou
Co-Rapporteur	AMARI Saïd	Maitre de conférences HdR, Université Sorbonne Paris Nord
Examineur	ACHOUR Hakim	Professeur, Université Mouloud Mammeri de Tizi Ouzou
Examinatrice	GHOMRI Latéfa	Professeur, Université Abou bekr Belkaid de Telmcen

ANNÉE : 10 JUILLET 2023



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## *Remerciements*

Ce travail de thèse a été effectué au Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP) de la faculté de Génie Électrique et Informatique de l'Université Mouloud Mammeri de Tizi-Ouzou.

Tout d'abord, je tiens à exprimer ma gratitude envers Monsieur Redouane KARA, Professeur à l'Université Mouloud Mammeri de Tizi-Ouzou, pour avoir dirigé cette thèse. Il est difficile de trouver des mots suffisamment forts pour exprimer le plaisir et la gratitude que j'ai éprouvé en travaillant sous sa direction. Je lui serai toujours reconnaissant de m'avoir accordé sa confiance en acceptant de superviser ce travail, de m'avoir prodigué de multiples et précieux conseils, ainsi que pour sa rigueur scientifique, sa persévérance et sa patience qui m'ont guidé tout au long de cette thèse. J'espère que ces quelques lignes lui témoigneront toute ma profonde gratitude.

Je suis également extrêmement reconnaissant envers Monsieur Saïd Amari, Maître de conférences HdR à l'université Sorbonne Paris Nord/LURPA ENS Paris Saclay, qui est mon co-directeur de thèse. Je tiens à exprimer ma gratitude pour sa collaboration précieuse, sa disponibilité, son attention constante envers mes travaux et son écoute attentive, son implication a été cruciale pour l'aboutissement de cette thèse. Collaborer avec lui est un réel enchantement.

Je remercie chaleureusement Monsieur Saïd DJENNOUNE, Professeur à l'université Mouloud Mammeri de Tizi-Ouzou pour avoir accepté de présider le jury et d'évaluer notre travail. Je suis honoré et reconnaissant de l'opportunité qui m'a été donnée de bénéficier de son expertise et de sa sagesse. Ses remarques et suggestions précieuses contribueront certainement à améliorer la qualité de ce manuscrit. Je tiens à exprimer ma profonde gratitude à travers ces lignes.

Mes sincères remerciements vont également à Monsieur Hakim ACHOUR, Professeur à l'université Mouloud Mammeri de Tizi-Ouzou ainsi que Madame Latéfa GHOMRI, Professeur à l'université de Tlemcen, d'avoir accepté de faire partie de ce jury de soutenance et pour l'honneur qu'ils me font par leurs présences. Je leur suis infiniment reconnaissant de porter leurs regards d'experts sur notre travail et notre manuscrit de thèse. Leurs suggestions intéressantes, leurs remarques avisées et leurs discussions constructives m'aideront certainement à améliorer ce

travail et me serviront à l'avenir pour en réaliser d'autres.

Je tiens à exprimer ma gratitude envers Madame Ouardia MEGHERBI et le professeur Said GUERMAH pour leur sollicitude et leurs aides précieuses qui m'ont permis de mener à bien cette thèse.

Je souhaiterais également exprimer ma reconnaissance envers tous mes amis et collègues du Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP) sans exception, pour leur expertise remarquable, leur bienveillante et leur attitude positive qui ont grandement contribué à mon avancement dans mes travaux dans un environnement propice et agréable.

Pendant toute la durée de ma thèse, j'ai bénéficié d'un soutien indéfectible de la part de ma famille, en particulier de mes chers parents, mes chères soeurs, mon cher frère et ma fiancée. Leur encouragement constant et leur dévouement m'ont permis de mener à bien ce travail. Je suis infiniment reconnaissant envers eux pour leur confiance, et il est difficile pour moi de trouver les mots pour exprimer toute ma gratitude.

Je souhaiterais exprimer ma reconnaissance envers toutes les personnes qui ont contribué, de près ou de loin, à l'élaboration de cette thèse.

# Table des matières

<b>Symboles et Notations</b>	<b>ix</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Algèbre des dioïdes et réseaux de Petri</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 Réseaux de Petri . . . . .	5
1.2.1 Définitions et notations . . . . .	6
1.2.2 Propriétés des RdPs . . . . .	10
1.2.3 Matrice d'incidence . . . . .	11
1.2.4 La notion de conflit dans les RdPs . . . . .	12
1.2.5 Les graphes d'événements . . . . .	14
1.3 Algèbre des dioïdes . . . . .	15
1.3.1 Quelques définitions des Dioïde . . . . .	16
1.3.2 Résolution d'équations Linéaires dans les dioïdes $\mathbb{R}_{max}$ et $\mathbb{R}_{min}$ . . . . .	18
1.4 Représentation linéaire des Graphes d'événements temporisés . . . . .	19
1.4.1 Définitions et notations . . . . .	20
1.4.2 Fonctions dateur, domaine événementiel . . . . .	20
1.4.3 Fonction compteur, domaine temporel . . . . .	23
1.5 Conclusion . . . . .	26
<b>2 Modélisations des graphes d'événements temporisés avec conflit</b>	<b>27</b>
2.1 Introduction . . . . .	27
2.2 Les RdPs avec partage de ressources . . . . .	28
2.2.1 Partage de ressources . . . . .	28

2.3	Modélisation des RGETC . . . . .	29
2.3.1	Méthode de Addad . . . . .	30
2.3.2	Méthode de Moradi . . . . .	33
2.3.3	Modélisation des RGETCs par les systèmes linéaires Max-plus à commutation (méthode proposée) . . . . .	36
2.4	Comparaison des trois méthodes présentées . . . . .	45
2.5	Conclusion . . . . .	45
<b>3</b>	<b>Commande sous contraintes temporelles des RGETCs</b>	<b>47</b>
3.1	Introduction . . . . .	47
3.2	Synthèse de lois de commande pour RGETC sous contraintes temporelles . . . . .	48
3.2.1	Contraintes temporelles . . . . .	48
3.3	Cas des RGETCs complètement observables . . . . .	49
3.3.1	Synthèse de commande pour les RGETCs à temps critique . . . . .	49
3.3.2	Une transition source unique dans chaque GET . . . . .	50
3.3.3	Plusieurs transitions sources pour chaque GET . . . . .	53
3.4	Exemple d'application : Gestion d'un croisement de lignes ferroviaires . . . . .	55
3.4.1	Modèles linéaires Max-Plus à commutations du réseau ferroviaire . . . . .	55
3.4.2	Commande par retour d'état du réseau ferroviaire à temps critique . . . . .	59
3.5	RGETC partiellement observable . . . . .	64
3.5.1	Une transition source dans chaque GET . . . . .	65
3.5.2	plusieurs transitions sources dans chaque GET . . . . .	67
3.6	Exemple d'illustration . . . . .	68
3.6.1	Premier cas : toutes les transitions sont considérées observables . . . . .	68
3.6.2	Deuxième cas : Les transitions ne sont pas toutes considérées comme observables. . . . .	72
3.7	Conclusion . . . . .	77
	<b>Conclusion Générale</b>	<b>79</b>
	<b>Annexe 1. Les autmates et quelque définitions des dioides</b>	<b>86</b>

# Table des figures

1.1	Exemple d'un RdP . . . . .	6
1.2	Les RdPs marqué et non marqué . . . . .	7
1.3	Franchissement d'une transition . . . . .	8
1.4	Un exemple de RdP comme graphe d'événements . . . . .	9
1.5	Évolution des marquages accessible . . . . .	10
1.6	Un RdP borné . . . . .	11
1.7	(A) et (B) Pas de conflits effectifs, (C) Conflit effectif, (D) Conflit effectif mais persistant. . . . .	13
1.8	GET T-Temporisé . . . . .	14
1.9	GET P-Temporisé . . . . .	14
1.10	Exemple de RGETC . . . . .	15
1.11	Modèle de GET . . . . .	22
1.12	Décomposition de la place $p_2$ . . . . .	23
1.13	Décomposition de la place $p_1$ de GET . . . . .	25
2.1	Une machine nécessitant une ressource . . . . .	29
2.2	Deux machines en concurrence . . . . .	29
2.3	Machine à double concurrence . . . . .	30
2.4	Réseau de Graphes d'Événements Temporisés avec Conflits . . . . .	30
2.5	Exemple de RGETC pour illustrer la méthode de Addad . . . . .	32
2.6	Un RdP avec des ressources partagées . . . . .	34
2.7	Un RDP avec partage de ressources pour illustrer la méthode de Moradi . . . . .	35
2.8	Comutateur avant et arrière . . . . .	37
2.9	Exemple de configuration . . . . .	38

2.10	Dateur généralisé . . . . .	38
2.11	Un modèle de RGETC . . . . .	42
2.12	Automate à états finis des modes du RGETC . . . . .	42
2.13	Exemple de RGETC . . . . .	43
2.14	Représentations du système à commutation comme Automate . . . . .	44
3.1	Contrainte temporelle . . . . .	48
3.2	Croisement de trois lignes ferroviaires . . . . .	56
3.3	RGETC de croisement de trois lignes ferroviaires . . . . .	57
3.4	RGETCs commandés du croisement du réseau ferroviaire . . . . .	62
3.5	Évolution des dateurs $t_3$ et $t_4$ . . . . .	63
3.6	Évolution des dateurs de $t_5$ et $t_6$ . . . . .	63
3.7	Évolution des dateurs $t_{11}$ et $t_{12}$ . . . . .	63
3.8	Zone de production . . . . .	69
3.9	RGETC correspondant à la zone de production . . . . .	69
3.10	Loi de commande pour la configuration 1 . . . . .	73
3.11	Loi de commande pour la configuration 2 . . . . .	73
3.12	Évolution des dateurs $t_2$ et $t_6$ . . . . .	76
3.13	Évolution des dateurs $t_4$ et $t_6$ . . . . .	76
3.14	Évolution dateurs $t_8$ et $t_{10}$ . . . . .	77
3.15	Automate fini . . . . .	87

# Liste des tableaux



# Symboles et Notations

---

$\mathbb{R}$ :	Ensemble des nombres réels
$\mathbb{N}$ :	Ensemble des nombres Naturelle
$\mathbb{D}$ :	Dioïde
$\preceq$ :	Relation d'ordre liée à un dioïde
$\mathbb{R}_{max}$ :	Dioïde Max-plus
$\bar{\mathbb{R}}_{max}$ :	Dioïde complet Max-plus
$\mathbb{R}_{min}$ :	Dioïde Min-plus
$\bar{\mathbb{R}}_{min}$ :	Dioïde complet Min-plus
$\mathbb{R}_{max}^{n \times m}$ :	Ensemble des matrices dans Max-plus de dimensions $n \times m$
$\mathcal{C}_i$ :	Une configuration
$B$ :	Un commutateur arrière (Backward)
$F$ :	Un commutateur avant (Forwad)
$Q_i$ :	Un état d'un automat
$P$ :	Un ensemble fini de places $p_i$ d'un réseau de petri
$T$ :	Un ensemble fini de transitions $t_i$ d'un réseau de petri
$ \cdot $ :	Cardinalité d'un ensemble
$M_i$ :	Nombre de jetons à la place $p_i$
$M$ :	L'ensemble de marquage d'un réseau de Petri
$W$ :	Matrice d'incidence d'un réseau de Petri
$M_0$ :	L'ensemble de marquage initial d'un réseau de Petri
$M_0$ :	L'ensemble de marquage initial d'un réseau de Petri
$\tau_i$ :	Le délai associé à la place $p_i$
$k$ :	L'indice d'événement
$\oplus$ :	Addition dans un Dioïde
$\otimes$ :	Multiplication dans un Dioïde
$\varepsilon$ :	Elément neutre pour la loi $\oplus$
$e$ :	Elément neutre pour la loi $\otimes$
$a^*$ :	Etoile de kleene.
$x(t)$ :	Fonction compteur associée à la transition $t_i$
$x(k)$ :	Fonction dateur associée à la transition $t_i$
$u(k)$ :	Fonction dateur associée à la transition source $t_u$
$A^{\gamma(k)}$ :	Matrice dynamique d'un mode $\gamma(k)$
$B^{\gamma(k)}$ :	Matrice de commande d'un mode $\gamma(k)$

<i>SED</i> :	Systeme à événement discret
<i>RdP</i> :	Réseaux de Petri
<i>GET</i> :	Graphes d'Événements Temporisés
<i>RGETC</i> :	Réseaux de Graphes d'Evénements Temporisés avec Conflit
<i>SLMC</i> :	Systemes Linéaires Max-plus à Commutation



# Introduction générale

Un système à événements discret (SED) est un type de système dynamique qui évolue par occurrence d'événements distincts et discontinus. Les événements représentent les changements d'état, les transitions de mode, les arrivées de messages où des actions déclenchées par une horloge. Les systèmes à événements discrets sont couramment utilisés pour modéliser des processus dans des domaines tels que l'ingénierie, comme les ateliers de fabrication flexibles [1, 2], les systèmes temps réel [3, 4], le transport urbain ou ferroviaire [5, 6, 7], les réseaux informatiques et de communications [8, 9] et les procédés de traitement thermique ou chimique [10, 11]. Il existe plusieurs formalismes et méthodologies de modélisation pour étudier cette catégorie de systèmes (SEDs), tels que les réseaux de Petri (RdPs), les automates à états, les diagrammes d'états, l'algèbre Max-plus, les réseaux de files d'attente, les chaînes de Markov et les modèles bayésiens. Il est actuellement admis que les réseaux de Petri (RdPs) sont un outil puissant pour l'analyse et la commande des systèmes à événements discrets. Ils sont souvent utilisés pour représenter des phénomènes de synchronisation, de parallélisme, de concurrence et de partage des ressources.

Les Graphes d'Événements Temporisés (GETs) sont une catégorie de Réseaux de Petri (RdP) qui se focalisent exclusivement sur la représentation des phénomènes de synchronisation et de parallélisme. Cela a conduit à une théorie générale pour les systèmes linéaires Max-plus où de nombreux concepts de la théorie des systèmes standards ont été adaptés aux GETs. Malheureusement, les GETs sont des RdPs temporisés dont chaque places possède exactement une transition en amont et une transition en aval. Ils ne sont donc pas appropriés pour représenter les phénomènes de partage de ressources. Les systèmes avec des ressources partagées sont souvent rencontrés dans les applications réelles, par exemple : dans un système de communication en réseau, les processeurs et les routeurs sont partagés entre plusieurs utilisateurs. Dans un système de transport ferroviaire, de nombreux trains partagent la même voie.

Dans cette thèse, nous étudions une classe spéciale des SEDs, les réseaux de graphes d'événements temporisés avec conflit (RGETCs), une classe de systèmes à événements discrets impliquant des ressources partagées.

L'intégration des technologies de l'information et de la communication dans l'ingénierie génère une complexité accrue, en particulier en ce qui concerne la dimension temps réel de ces systèmes. Ces exigences sont exprimées à travers diverses spécifications et contraintes, telles que des états interdits, des intervalles de temps ou des périodes de validité. Dans les systèmes à événements discrets temporisés, les opérations doivent être exécutées dans un délai spécifique et, parfois, elles doivent être terminées avant une date prédéfinie. La violation d'une contrainte temporelle peut entraîner des défaillances et des conséquences parfois graves. Dans de nombreux domaines d'application, il est crucial de prendre en considération de manière explicite la dimension temporelle. Par exemple, il est essentiel d'évaluer la productivité par unité de temps d'un atelier ainsi que le temps moyen de transmission des informations au sein d'un réseau, le temps de traitement d'un processus, le temps de transmission des messages à travers une ligne de communication, etc. Il convient de noter que ces durées peuvent parfois représenter des contraintes physiques qui doivent obligatoirement être respectées.

Diverses approches ont été proposées dans la littérature pour la commande des SEDs, en prenant en considération différentes contraintes et spécifications. Plusieurs auteurs ont étudié la synthèse des lois de commande pour assurer le respect des exigences temporelles. Les auteurs dans [12, 13] ont discuté des problèmes de synthèse de commande en boucle fermée pour garantir que les systèmes considérés évoluent sans violer les contraintes temporelles imposées aux états du système. Les travaux dans [14, 15, 16] concernent la synthèse de lois de commande en boucle fermée pour les GETs soumis à des contraintes temporelles strictes. Dans ces approches, les contraintes temporelles sont exprimées dans l'algèbre Max-plus par des inégalités linéaires sur le vecteur d'état. Dans [17], les auteurs ont traité la conception de lois de commande en boucle fermée pour satisfaire un ensemble de contraintes temporelles dans des GETs observables et partiellement observables. Le problème de la commande de RGETC sous contraintes temporelles est un problème nouveau puisque la plupart des travaux sur la commande des SEDs sous contraintes temporelles considèrent les GETs.

L'objectif de cette thèse consiste à étendre l'utilisation de l'algèbre Max-plus pour décrire le comportement et la commande par retour d'état pour les réseaux des graphes d'événements

---

temporisées avec conflit (RGETCs) soumis à des contraintes temporelles. Ce travail est une généralisation des méthodes de commande existantes [12, 14, 18, 15, 16, 19, 13, 20]) pour traiter une classe plus générale de réseaux de Petri soumis à des contraintes temporelles et intégrant le partage de ressources. Pour résoudre ce problème de commande des RGETC soumise à des contraintes temporelles nous avons effectué les travaux suivant [21, 22, 23].

Les trois points suivants résument les objectifs essentiels de cette thèse :

- i) De proposer un cadre de modélisation algébrique permettant de décrire le comportement des réseaux de graphes d'événements temporisés avec conflit (RGETC) à l'aide d'équations linéaires Max-plus à commutation, on peut affirmer que cette méthodologie est une extension de l'utilisation des formalismes Max-plus pour décrire la dynamique d'une classe de systèmes à événements discrets avec partage de ressource entre plusieurs processus [21, 22].
- ii) Le deuxième objectif, est de développer de nouvelles approches formelles pour la conception des lois de commande afin d'assurer les contraintes temporelles dans les RGETCs en utilisant l'algèbre Max-plus [21, 22].
- iii) Le troisième objectif, est de relâcher l'hypothèse sur l'observabilité des transitions dans les GETs et de concevoir une approche plus générale pour la commande des RGETCs et de développer une méthode plus générale pour la conception de lois de commande assurant des contraintes temporelles strictes pour les RGETC partiellement observables [23].

Le présent travail est organisé en trois chapitres :

Dans **le chapitre 1**, nous rappelons dans une première partie les notions de base des réseaux de Petri et leurs propriétés ainsi que les graphes d'événements temporisés. Dans la partie suivante, nous introduisons l'algèbre des dioïdes, plus précisément l'algèbre Max-plus et Min-plus. Nous montrons comment représenter les GETs avec des fonctions dateurs et compteurs dans l'algèbre des dioïdes.

**Le chapitre 2** est consacré à la présentation du premier résultat de notre thèse concernant la modélisation des réseaux des graphes d'événements temporisés avec conflits (RGETCs) par des systèmes Max-plus linéaires à commutations. Pour cela, nous avons introduit les notions de commutateurs arrières et avant et la notion de compteur généralisé. En outre, nous avons discuté des avantages et des inconvénients que présente les méthodes de modélisation proposées dans la littérature pour cette classe de systèmes.

**Le chapitre 3** contient la deuxième contribution de la thèse :

Nous avons présenté les principales contributions de cette thèse sur la commande des réseaux des graphes d'événements temporisés avec conflit soumis à des contraintes de temps stricts. Dans cette étude, on considère des contraintes imposées sur un ou plusieurs chemins pour les réseaux des graphes d'événements temporisés avec conflit. Ce travail est divisé en deux parties, la première partie consiste à synthétiser des lois de commande pour les RGETCs observables dans les différents cas. Une extension pour les REGTCs partiellement observables est proposée dans la deuxième partie et une procédure pour synthétiser les lois de commande est développée pour garantir les contraintes temporelles. Ce chapitre se termine par un exemple illustratif. Enfin, cette thèse se termine par une conclusion générale et quelques perspectives.

# Chapitre 1

## Algèbre des dioïdes et réseaux de Petri

### 1.1 Introduction

L'objectif principal de cette thèse consiste à aborder la commande des systèmes à événements discrets tout en respectant diverses spécifications. Pour y parvenir, il est essentiel d'avoir des outils de modélisation et d'analyse mathématiques appropriés. Dans ce travail, nous nous appuyons sur l'utilisation des Réseaux des Graphes d'Événements Temporisés avec Conflit, une classe de réseaux de Petri, comme outil de modélisation. De plus, nous utilisons l'algèbre des dioïdes pour décrire le comportement de ces graphes à l'aide d'équations linéaires. Le premier chapitre de cette thèse introduit les concepts fondamentaux de l'algèbre des dioïdes et des réseaux de Petri, qui seront utilisés tout au long de ce manuscrit.

### 1.2 Réseaux de Petri

Un réseau de Petri est un outil pratique de modélisation graphique et mathématique permettant de représenter facilement la concurrence, la synchronisation et les conflits dans le système modélisé. Une classe de réseaux de Petri connue sous le nom de Graphe d'Événements Temporisés (GET) est fréquemment utilisée comme outil de modélisation dans l'étude du SED dans l'algèbre des dioïdes [24, 25]. Cette algèbre permet de représenter ces graphes à l'aide d'équations linéaires. En effet, la procédure consiste à modéliser d'abord le système considéré avec un GET, puis à déterminer les équations linéaires dans les dioïdes propres à l'analyse de ce système à partir du graphe obtenu.

### 1.2.1 Définitions et notations

Les informations contenues dans cette sous section sont tirées des références [26],[27] [28] et [29].

**Définition 1.1** *Un Réseau de Petri (RdP) est une structure graphique comportant un ensemble de places et de transitions, reliées par des arcs orientés, éventuellement avec des poids. Ces arcs sont des liens entre places et transitions ou entre transitions et places exclusivement. Dans cette structure se déplacent des jetons (ou marques) qui apparaissent dans des places et sont susceptibles de franchir les transition selon des sémantiques bien définies.*

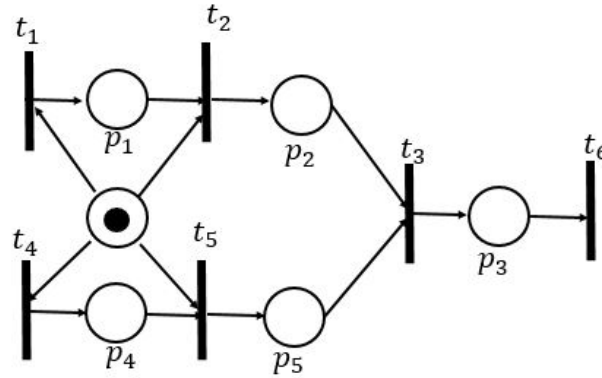


FIGURE 1.1: Exemple d'un RdP

De manière générale, les places sont généralement identifiées par des cercles et les transitions par des rectangles (Figure 1.1).

**Définition 1.2** *Formellement, un réseau de Petri marqué est un quintuplet  $R = (P, T, Pre, Post, M)$*

*où :*

- $P$  est un ensemble fini de places,  $P = \{p_1, p_2, \dots, p_{|P|}\}$ , où  $|\cdot|$  représente la cardinalité d'un ensemble.
- $T$  est un ensemble fini de transitions,  $T = \{t_1, t_2, \dots, t_{|T|}\}$ .
- $Pre : P \times T \rightarrow \{0, 1\}$  est l'application dite d'incidence avant, telle que :  $Pre(p_i, t_j) = 1$  si la valeur de l'arc dirigé depuis la transition  $t_j$  vers la place  $p_i$  existe et 0 sinon.
- $Post : P \times T \rightarrow \{0, 1\}$  est ce que l'on appelle l'application d'incidence arrière, telle que  $Post(p_i, t_j) = 1$  si la valeur de l'arc dirigé de la place  $p_i$  vers la transition  $t_j$  existe et 0 sinon.

–  $M_0$  correspond au vecteur des marquages initiaux dans un RdP.

**Définition 1.3** Une transition qui n'a pas de place en amont est appelée une transition source, tandis qu'une transition qui n'a pas de place en aval est appelée une transition puit.

## Marquage

Les place peuvent contenir des jetons (marque) la dynamique d'un RdP est véhiculé par le déplacement des marques. Le marquage d'un réseau de Petri est un vecteur  $M = [m_1, m_2, \dots, m_{|P|}]^T$ . On note  $M_0$  le marquage initiale d'un RdP,  $M_0 = [m_{0_1}, m_{0_2}, \dots, m_{0_{|P|}}]^T$ , où  $m_i$  et  $m_{0_i}$  représente successivement le marquage et le marquage initial de la place  $p_i$ .

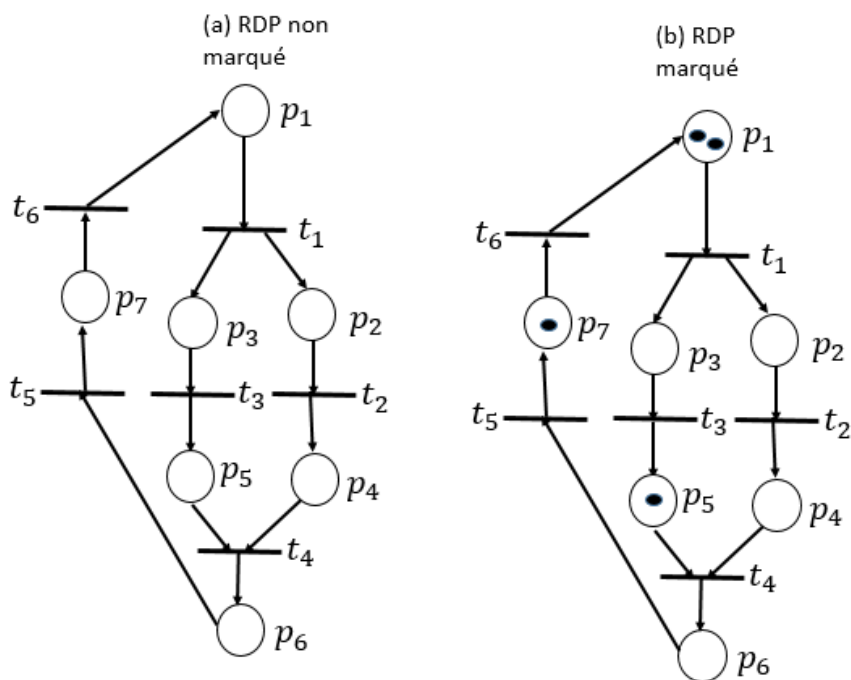


FIGURE 1.2: Les RdPs marqué et non marqué

La Figure 1.2 (a) représente un RdP non marqué où tous ses marquages sont nuls  $m_{0_i} = 0$ ,  $i = 1, 2 \dots 7$ .

Pour l'exemple de la Figure 1.2 (b), nous avons son marquage initial  $M_0$ , défini par le vecteur,  $M_0 = [2, 0, 0, 0, 1, 0, 1]^T$ .

### Franchissement des transitions :

Pour franchir une transition, il est nécessaire que chacune des places d'entrée de cette transition contienne au moins un jeton. On dit alors que la transition est franchissable ou activée. Une transition source est toujours activée. Dans les figures 1.3 (a), (b) et (c) (avant le franchissement), les transitions sont franchies car dans chaque cas, les places  $p_1$  et  $p_2$  contiennent au moins un jeton. Cependant, dans l'exemple de la figure 1.3 (d), la transition  $t_1$  n'est pas franchie car  $p_2$  ne contient aucun jeton.

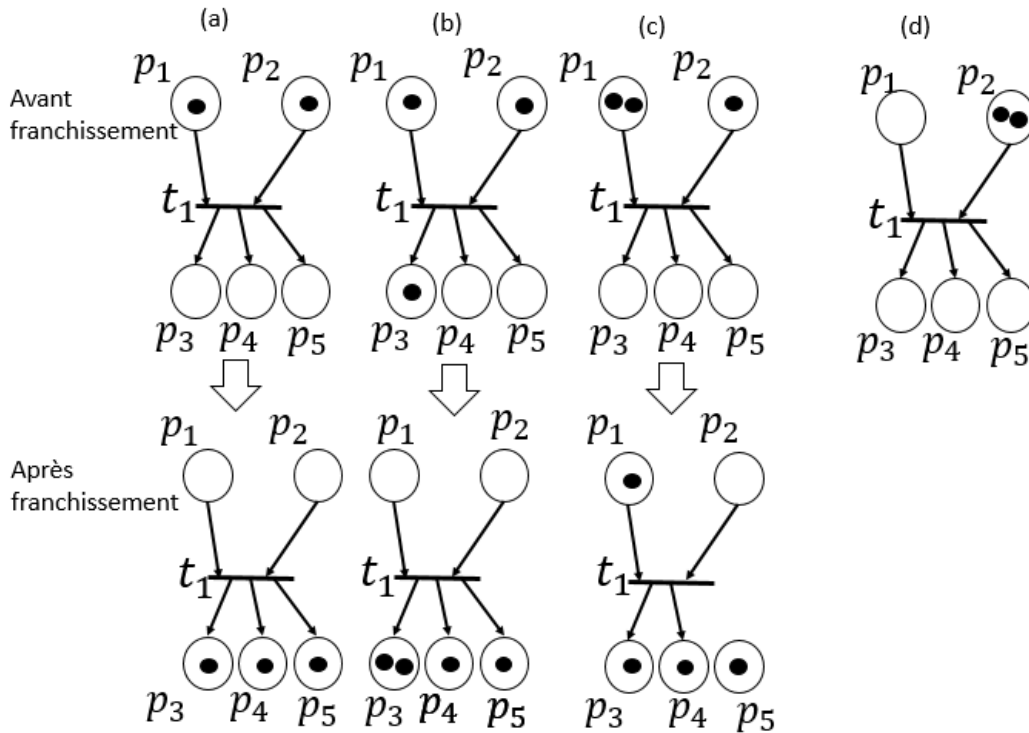


FIGURE 1.3: Franchissement d'une transition

**Définition 1.4** On dit que une transition est franchie par un marquage  $M$  donnée si : toute les places en entrée de la transition sont marquées. On note par  ${}^o t_j$  l'ensemble des places qui sont en amont de  $t_j$  et  $t_j^o$  l'ensemble des places en aval  $t_j$ .  $t_j$  est franchie si  $\forall p_i \in {}^o t_j$  on a  $M_i \geq 1$ . Le franchissement d'un transition validée  $t_j$  consiste à enlever une marque de toute place appartenante à  ${}^o t_j$  et ajoute une marque à chaque place  $p_i \in t_j^o$

Si on prend l'exemple de la Figure 1.3 on a  $t_1^o = \{p_3, p_4, p_5\}$  et  ${}^o t_1 = \{p_1, p_2\}$ .

**Remarque 1.5** La notion d'état est véhiculé par la notion de marquage  $M_i \xrightarrow{t_i} M_{i+1}$ , on écrit

$$M_i [t_i > M_{i+1}.$$

### Notion d'atteignabilité

Un marquage  $M_q$  étant donné est dit atteignable depuis le marquage initial  $M_0$  dans un RdP si il existe une séquence de transitions  $t_1 t_2 \cdots t_s$  telle que  $M_0 [t_1 > M_2 [t_2 > M_3 \cdots M_{q-1} [t_s > M_q$ .

On écrit :

- $\gamma = t_1 t_2 \cdots t_s$  et  $M_0 [\gamma > M_q$ . On prend l'exemple de la Figure 1.5 une séquence de franchissements s'écrit par exemple :

$$\gamma = \langle t_1 t_3 t_2 t_4 \rangle$$

- L'ensemble de toutes les séquences possibles (ou réalisables) est noté  $L(M_0) \subset T^*$ .
- L'ensemble de tous les marquages atteignables est noté  $R(M_0)$ .

**Exemple 1.1** La Figure 1.5 représente l'évolution des marquages correspondant à l'exemple de la Figure 1.4. Où  $T = \{t_1, t_2, t_3, t_4\}$ ;  $P = \{p_1, p_2, p_3, p_4\}$ ;  $M_0 = [1, 0, 0, 0]^T$

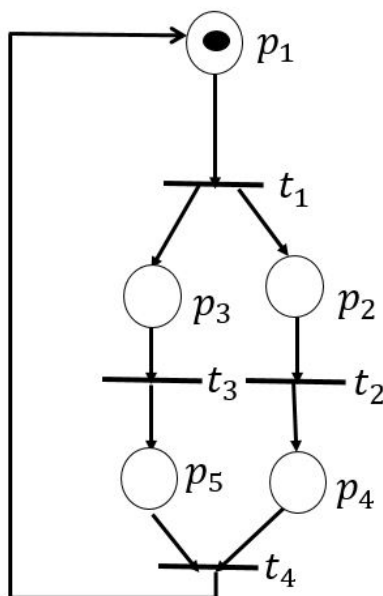


FIGURE 1.4: Un exemple de RdP comme graphe d'événements

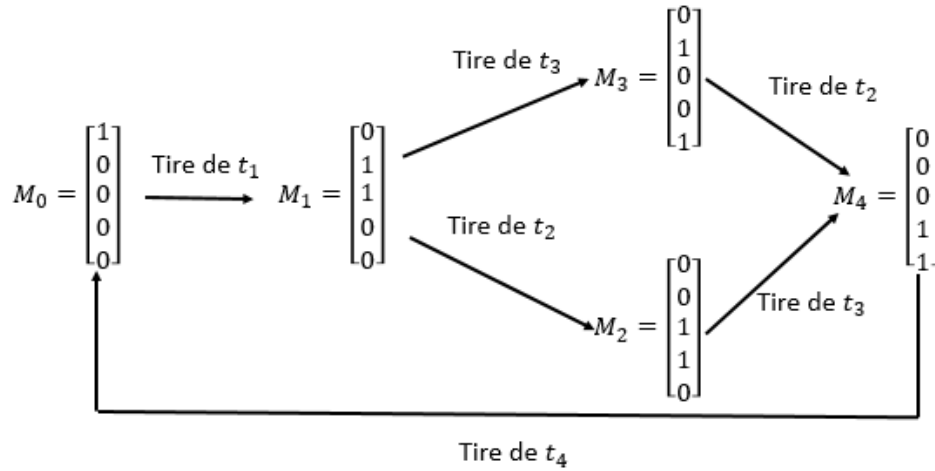


FIGURE 1.5: Évolution des marquages accessible

$$\mathcal{R}(M_0) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

### 1.2.2 Propriétés des RdPs

- Un conflit structurel correspond à l'existence d'une place qui possède au moins deux transitions de sortie. Un conflit structurel est représenté par une paire composée d'une place et d'un ensemble de transitions de sortie de cette place. On note un conflit structurel par :  $K = (p_i, \langle t_j, t_k, \dots \rangle)$ .
- Un Réseau de Petri est dit borné pour un marquage initial  $M_0$ , si  $\forall M_i \in \mathcal{R}(M_0)$  alors  $M_i \leq a_i | a_i \in \mathbb{N}$ .  
Dans l'exemple de la Figure 1.6, on a  $M_i \leq 2$  pour tout  $M_0$ , ce qui signifie que le Réseau de Petri est borné.
- Un Réseau de Petri est considéré comme sauf pour un marquage initial  $M_0$  si, pour chaque marquage accessible, chaque place contient au plus un jeton. Un réseau sauf est borné puisque  $M_i \in [0, 1], \forall M_i \in \mathcal{R}(M_0)$ .
- Une transition  $t_j$  dans un Réseau de Petri est considérée comme vivante, pour un marquage

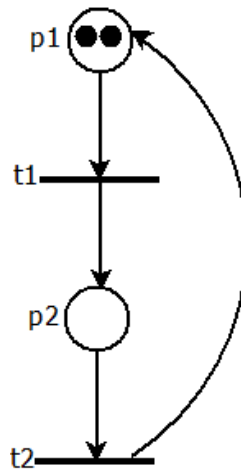


FIGURE 1.6: Un RdP borné

initial  $M_0$ , si pour tout marquage accessible, il existe une séquence de franchissements  $\gamma$  qui inclut la transition  $t_j$  à partir de ce marquage accessible :

$\forall M_i \in \mathcal{R}(M_0), \exists \gamma | t_j \in \gamma$  et un marquage  $M_k$  tel que  $M_i [\gamma > M_k$ .

- Si toutes les transitions d'un Réseau de Petri sont vivantes, pour un marquage initial  $M_0$ , alors le Réseau de Petri est vivant pour ce marquage initial.
- Un blocage, également connu sous le nom d'état puits, est un marquage  $M_p$  dans lequel aucune transition n'est activée ou franchissable.
- Un Réseau de Petri est considéré comme sans blocage, pour un marquage initial  $M_0$ , si aucun des marquages accessibles  $M_i$  ne correspond à un blocage.
- Un réseau de Petri possède un état d'accueil noté  $M_a$ , pour un marquage initial  $M_0$ , si  $\forall M_i \in \mathcal{R}(M_0), \exists \gamma | M_i [\gamma > M_a$ .
- Un Réseau de Petri est dit réinitialisable, pour un marquage initial  $M_0$ , si  $M_0$  correspond à un état d'accueil.

### 1.2.3 Matrice d'incidence

- La matrice suivante est appelée Matrice d'incidence avant :

$$W_{ij}^- = \begin{cases} 1 & \text{si } \exists \text{ un arc de la place } p_i \text{ à la transition } t_j \\ 0 & \text{sinon} \end{cases}$$

– La matrice suivante est appelée la Matrice d'incidence arrière :

$$W_{ij}^+ = \begin{cases} 1 & \text{si } \exists \text{ un arc de la transition } t_j \text{ à la place } p_i \\ 0 & \text{sinon} \end{cases}$$

Pour l'exemple de la Figure 1.4, on aura :

$$W_{ij}^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; W_{ij}^+ = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1.1)$$

Les lignes correspondent ainsi aux places, et les colonnes aux transitions. Notons que ces matrices ne sont pas forcément carrées.

– La matrice suivante est la matrice d'incidence :

$$W = W^+ - W^-$$

Pour la matrice d'incidence de la Figure 1.4, est donnée comme suit :

$$W = \begin{bmatrix} -1 & 0 & 0 & +1 \\ +1 & -1 & 0 & 0 \\ +1 & 0 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{bmatrix} \quad (1.2)$$

#### 1.2.4 La notion de conflit dans les RdPs

Après avoir introduit le concept de conflit structurel dans les propriétés des Réseaux de Petri 1.2.2, nous allons maintenant définir les notions de conflit effectif, de conflit général, de persistance et de concurrence. Nous rappelons qu'un conflit structurel se produit lorsqu'un ensemble d'au moins deux transitions, partagent une même place d'entrée. Cela peut être exprimé par la paire  $K = (p_i, \{t_j, t_k, \dots\})$ .

Dans un Réseau de Petri ordinaire, un conflit effectif se produit lorsqu'il existe un conflit structurel  $K$  et un marquage  $m_i$  où le nombre de jetons dans  $p_i$  est inférieur au nombre des transitions de sortie activées par  $m_i$ . Un conflit effectif est représenté par  $K^E = (p_i, \{t_j, t_k, \dots\}, m_i)$ .

**Définition 1.6** Un conflit effectif, représenté par  $K^E = (p_i, \{t_j; t_k \dots\}, m_i)$ , se produit lorsque le conflit structurel  $K = (p_i, \{t_j, t_k \dots\})$  coïncide avec un marquage  $m_i$  où les transitions de l'ensemble  $\{t_j; t_k \dots\}$  sont activées et où le nombre de jetons dans  $p_i$  est inférieur à la somme des arcs  $p_i \rightarrow t_j$  et  $p_i \rightarrow t_k$ .

La Figure 1.7 illustre la notion de conflit structurel  $K_1 = (p_2, t_1, t_2)$ . Ce conflit existe dans les figures 1.7 (A), (B) et (C). Dans la Figure 1.7 (A), seul le marquage de la place  $p_1$  est présent, ce qui signifie qu'il sera uniquement consommé par la transition  $t_1$ . Dans la figure (B), les deux transitions de sortie de la place de conflit sont activées car il y a deux jetons dans la place  $p_2$ , mais il n'y a pas de conflit effectif, ce qui permet le franchissement des deux transitions. Dans la Figure 1.7 (C), le marquage active à la fois  $t_1$  et  $t_2$ , mais étant donné qu'il n'y a qu'un seul jeton dans  $p_2$ , un conflit effectif  $K^E = (p_2, \{t_1; t_2\}, 1)$  se produit.

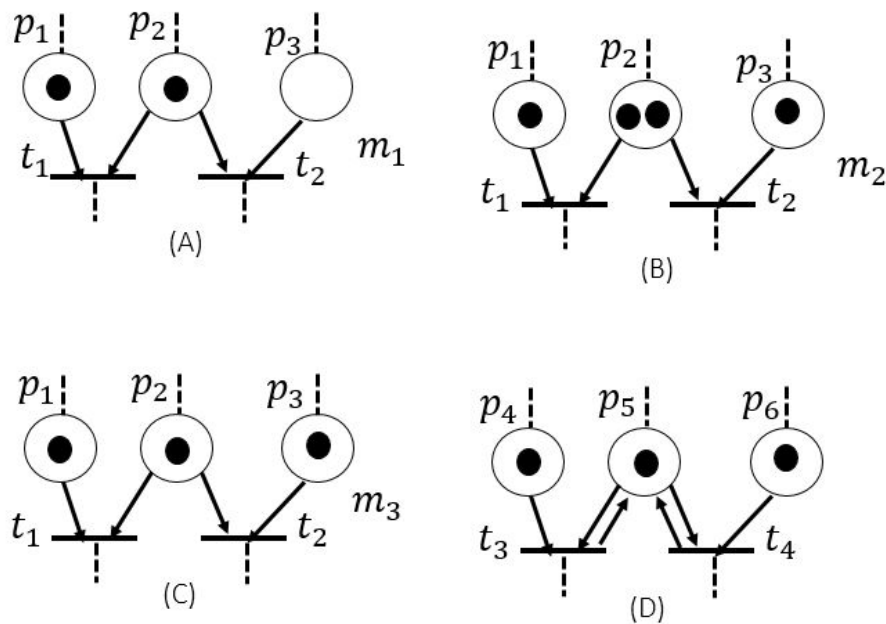


FIGURE 1.7: (A) et (B) Pas de conflits effectifs, (C) Conflit effectif, (D) Conflit effectif mais persistant.

Dans la Figure 1.7 (D), bien qu'il y a un conflit effectif, la structure est particulière. En effet, le marquage ne permet de tirer qu'une seule des transitions activées. Cependant, si l'une d'entre elles est franchie, par exemple  $t_3$ , la transition  $t_4$  peut ainsi être franchie puisqu'il y a encore un jeton dans la place  $p_5$ . L'ordre dans lequel  $t_3$  et  $t_4$  sont activées peut ne pas avoir d'importance, puisque les séquences  $t_3t_4$  et  $t_4t_3$  sont toutes les deux possibles.

### 1.2.5 Les graphes d'événements

**Définition 1.7** Un graphe d'événements  $G$  est un réseau de Petri ordinaire où chaque place possède exactement une transition d'entrée et une transition de sortie. Un graphe d'événement est dit temporisé (GET) si des temporisations sont associées aux places (P-Temporisé) et/ou transitions (T-Temporisé). Cela est illustré dans la Figure 1.8 et la Figure 1.9.

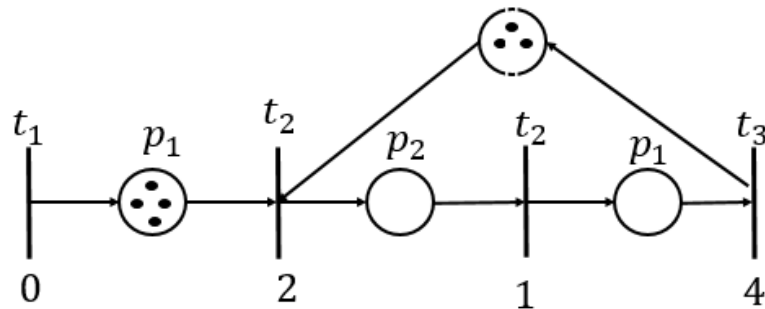


FIGURE 1.8: GET T-Temporisé

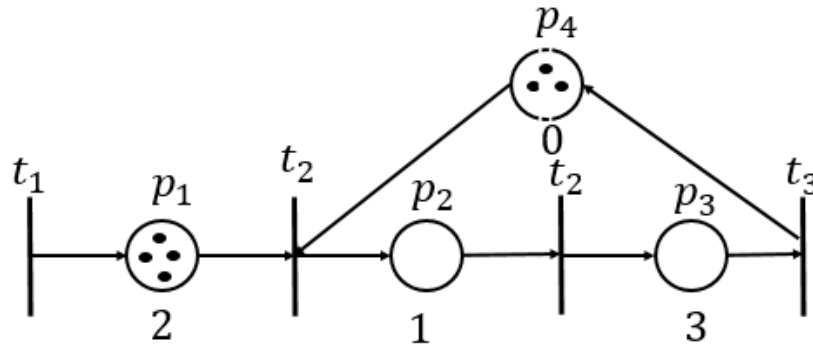


FIGURE 1.9: GET P-Temporisé

**Remarque 1.8** Dans la suite de ce manuscrit, on considère uniquement les graphes d'événement P-temporisé.

**Définition 1.9** Un réseau de Petri P-temporisé est un couple  $N = (R, \text{Tempo})$ , où :

- $R$  est un RdP marqué.
- $\text{Tempo}$  est une application de l'ensemble des places  $P$  à l'ensemble rationnels non négatifs  $Q^+$ ,  $\text{Tempo}(p_i) = \tau_i$ . Avec  $\tau_i$  est le délai associé à la place  $p_i$ .

**Définition 1.10** Un réseau de graphes d'événements temporisés avec conflit est un ensemble de GETs (serveur, machine, robot..etc) noté  $G_i = \{G_1, G_2, G_3, \dots, G_q\}$ , connectés à un ensemble de places de conflit (ressources partagées)  $\tilde{P} = \{\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, \dots, \tilde{p}_L\}$ . La temporisation associée à la place de conflit  $\tilde{p}_i$  est notée  $\tilde{\tau}_i$ . La Figure 1.10 illustre un exemple de tels graphes.

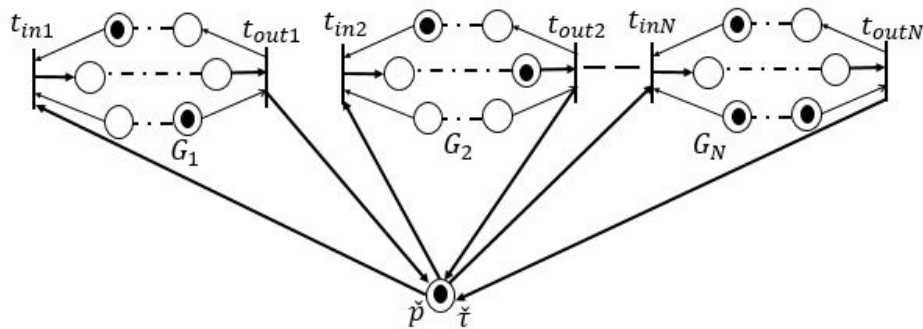


FIGURE 1.10: Exemple de RGETC

### 1.3 Algèbre des dioïdes

Les informations de cette sous-section sont basées sur les références suivantes : [20], [18], [30] et [31].

Un dioïde est un ensemble doté de deux opérations, appelées addition ( $\oplus$ ) et multiplication ( $\otimes$ ), qui satisfont certaines propriétés. Les éléments neutres de ces opérations sont respectivement  $\varepsilon = -\infty$  et  $e = 0$ . La loi de multiplication ( $\otimes$ ) est distributive par rapport à la loi d'addition ( $\oplus$ ), aussi bien à gauche qu'à droite. La loi d'addition ( $\oplus$ ) est idempotente et  $\varepsilon$  agit comme un élément absorbant pour la multiplication ( $\otimes$ ). Les éléments d'un dioïde peuvent être des nombres, des matrices, ou des fonctions. Un exemple de dioïde est le dioïde Max-plus (ou Min-plus), où l'addition ( $\oplus$ ) représente le maximum (ou minimum) et la multiplication ( $\otimes$ ) représente l'addition usuelle.

Les dioïdes sont des structures algébriques appropriées pour obtenir des modèles "linéaires" pour certains systèmes à événements discrets. Ils trouvent de nombreuses applications en informatique, en théorie des systèmes, en optimisation et en analyse numérique. Pour plus de détails,

nous invitons le lecteur à consulter les références suivantes : [29], [32]

### 1.3.1 Quelques définitions des Dioïde

**Définition 1.11** (*Dioïde*) *Un ensemble  $D$  équipé de deux opérations internes, notées respectivement  $\oplus$  pour l'addition et  $\otimes$  pour la multiplication, forme un dioïde lorsque les axiomes suivants sont satisfaits :*

1. *Associativité de  $\oplus$  et  $\otimes$*

$$\forall a, b, c \in D$$

$$(a \oplus b) \oplus c = a \oplus (b \oplus c) \text{ et } (a \otimes b) \otimes c = a \otimes (b \otimes c)$$

2. *Commutativité  $\oplus$*

$$\forall a, b \in D, a \oplus b = b \oplus a$$

3. *Distributivité de  $\otimes$  sur  $\oplus$*

$$\forall a, b, c \in D, (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$$

4. *Éléments neutres de  $\oplus$  ("zéro") et de  $\otimes$  ("identité")*

$$\exists \varepsilon \in D : \forall a \in D, a \oplus \varepsilon = a,$$

$$\exists e \in D : \forall a \in D, a \otimes e = e \otimes a = a$$

5. *Propriété d'absorption du zéro pour  $\otimes$*

$$\forall a \in D, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$$

6. *Idempotente de l'addition*

$$\forall a \in D, a \oplus a = a$$

**Définition 1.12** *Si la loi de multiplication, notée  $\otimes$ , d'un dioïde est commutative, on dit alors que ce dioïde est commutatif.*

**Définition 1.13** *Une relation  $T$  définit un ordre sur  $D$  si les trois axiomes suivants sont vérifiés :*

1.  $T$  est réflexive :  $a T a$

2.  $T$  est transitive : si  $a T b$  et  $a T c$  alors  $b T c$ .

3.  $T$  est antisymétrique : si  $a T b$  et  $a T c$  alors  $b = c$ .

Tel que  $\forall a, b, c \in D$ .

**Remarque 1.14** Dans la suite de ce manuscrit, nous adoptons la notation "." pour représenter la loi  $\otimes$ , dans le but de rendre l'écriture plus concise.

Dans un dioïde donné, la propriété d'idempotence de la loi additive  $\oplus$  donne lieu à une relation d'ordre, notée  $\preceq$ , définie comme suit : pour tout  $(x, y) \in D^2$ ,  $x \preceq y$  si et seulement si  $x \oplus y = y$ .

En outre, cette relation d'ordre est compatible avec les opérations du dioïde, ce qui signifie que les propriétés suivantes sont vérifiées :

Pour tout  $(x, y, z) \in D^3$ ,  $x \preceq y \Leftrightarrow x \oplus z \preceq y \oplus z$ .  $x \preceq y \Leftrightarrow x \otimes z \preceq y \otimes z$ . Cela signifie que la relation d'ordre  $\preceq$  conserve les propriétés de structure du dioïde lorsqu'elle est utilisée avec les opérations  $\oplus$  et  $\otimes$ .

**Remarque 1.15**

– L'application *max* correspond à l'ordre usuel  $\leq$  définie par la relation  $\preceq$ ,  $a \preceq b \Leftrightarrow \max(a, b) = b$ .

– L'application *min* correspond à l'inverse de ordre usuel  $\leq$  définie par la relation  $\preceq$ ,  $a \preceq b \Leftrightarrow \min(a, b) = b$ .

### ***Dioïdes matriciel***

Considérons  $(D, \oplus, \otimes)$  comme un dioïde. L'ensemble des matrices de dimensions  $n \times p$  avec des coefficients dans  $D$  forme un dioïde de matrices, noté  $D^{n \times n}$ . Les lois  $\oplus$  et  $\otimes$  de ce dioïde de matrices sont définies pour  $A, B \in D^{n \times n}$ , comme suit :

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

$$(A \cdot B)_{ij} = \bigoplus_{k=1}^n (A_{ik} \cdot B_{kj}).$$

Ainsi, les opérations  $\oplus$  et  $\otimes$  sont appliquées respectivement élément par élément pour l'addition de matrices et selon une règle spécifique pour la multiplication de matrices. Le résultat de la multiplication de matrices est une nouvelle matrice  $C$  de dimensions  $n \times s$  dans le dioïde  $D$ .

**Exemple 1.2** Considérons les matrices  $A$  et  $B$ , où :

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

Dans le dioïde Min-plus ( $\mathbb{R}_{min}$ ) :

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

$$A \cdot B = \begin{bmatrix} \min(1 + e, 3 + 1) & \min(1 + 2, 3 + 4) \\ \min(e + e, \varepsilon + 1) & \min(e + 2, \varepsilon + 4) \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ e & e \end{bmatrix}$$

Dans le dioïde Max-plus ( $\mathbb{R}_{max}$ ) :

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

$$A \otimes B = \begin{bmatrix} \max(1 + e, 3 + 1) & \max(1 + 2, 3 + 4) \\ \max(e + e, \varepsilon + 1) & \max(e + 2, \varepsilon + 6) \end{bmatrix} = \begin{bmatrix} 4 & 7 \\ e & 2 \end{bmatrix}$$

### 1.3.2 Résolution d'équations Linéaires dans les dioïdes $\mathbb{R}_{max}$ et $\mathbb{R}_{min}$

Les dioïdes complets offrent des solutions particulières extrêmes pour certaines équations. Des recherches ont été menées dans ce domaine afin de bénéficier de ces résultats pour la théorie des dioïdes [24, 25]. Ces solutions particulières extrêmes sont particulièrement utiles pour résoudre des équations de type point fixe de la forme  $x = a \otimes x \oplus b$ . Nous utiliserons ces résultats dans la suite de notre étude.

**Définition 1.16** *Dans le contexte du dioïde complet  $D$ , l'étoile de Kleene d'un élément  $a$ , 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  $(I_n(i, j) = e$  si  $i = j$  et  $\varepsilon$  sinon).*

La notation  $a^+$  correspond à l'équation :  $a^+ = \bigoplus_{k \geq 1} a^k$ .

**Propriété** Soit  $D$  un dioïde complet tel que  $a, b \in D$  et  $c$  est un entier strictement positif.

On a :

1.  $a^0 a^* = a^+$
2.  $(a^*) a^* = a^*$
3.  $a^+ a^* = a^*$
4.  $(a \oplus b)^* = (a^* b)^* a^* = b^* (a b^*)^*$
5.  $(a b^*)^+ = a (a \oplus b)^*$
6.  $(a b^*)^* = e \oplus a (a \oplus b)^*$
7.  $a^* = (a^0 \oplus \dots \oplus a^{c-1}) (a^c)^*$

On peut trouver les preuves de ces propositions à la page 30 de [30].

**Théorème 1.17** [29] *L'équation  $x = a \cdot x \oplus b$  et l'inéquation  $x \preceq a \cdot x \oplus b$ , dans un dioïde complet admet une plus petit solution qui est donnée par la quantité  $a^* \otimes b$  avec  $a \in \mathbb{R}_{max}^{n \times n}$ ,  $b \in \mathbb{R}_{max}^{n \times m}$  et  $a^*$  est l'étoile de Kleene de  $a$ .*

**Démonstration 1.1** *Vous trouverez la démonstration de ce Théorème dans [29].*

## 1.4 Représentation linéaire des Graphes d'événements temporisés

Le comportement d'un graphe d'événements temporisé (GET) peut être modélisé à l'aide de deux systèmes d'équations : le système d'équations Max-Plus et le système d'équations Min-Plus. Par exemple, le système d'équations Max-Plus utilise les opérations de maximum et de somme directe pour calculer les temps d'activation des transitions. Pour chaque arc du graphe, on définit une fonction d'activation qui indique le temps minimum auquel l'événement suivant peut se produire, compte tenu du délai spécifié sur les places. Ensuite, on construit un système d'équations à partir de ces fonctions d'activation, en utilisant les opérations de maximum et de somme directe pour calculer les temps d'activation des transitions. Dans cette partie, nous développons comment décrire le comportement d'un GET par un système d'équations Max-Plus ou Min-Plus linéaires.

Les définitions suivantes sont utiles pour la suite de cette thèse.

### 1.4.1 Définitions et notations

**Définition 1.18** Dans un RdP, un chemin qui relie deux transitions,  $t_j$  et  $t_i$ , est une séquence de places et de transitions reliées par des arcs.

**Définition 1.19** Dans le contexte d'un RdP, un circuit est un chemin qui commence et se termine au même sommet.

**Définition 1.20** Un chemin élémentaire est défini comme un chemin qui ne présente aucune répétition d'arc, c'est-à-dire qu'aucun arc n'est traversé plus d'une fois.

La représentation dynamique des GETs, peut être exprimée sous forme d'équations linéaires dans les dioïdes  $\mathbb{R}_{min}$  ou  $\mathbb{R}_{max}$ . Dans cette représentation, nous considérons  $\tau_i$  comme la temporisation associée à la place  $p_i$  avec un marquage initial  $m_{0i}$ . La plus grande temporisation dans le graphe d'événements est notée  $\tau^{max}$ , définie comme étant le maximum parmi toutes les valeurs de  $\tau_i$ . De même, le plus grand marquage initial parmi toutes les places du graphe d'événements est défini comme  $M^{max} = \max(m_{0i})$ .

### 1.4.2 Fonctions dateur, domaine événementiel

Dans ce contexte, à chaque transition  $t_i$  est associée une fonction appelée dateur  $\theta_i(k) \in \mathbb{R}_{max}^n$ , qui représente la date du  $k^{\text{ème}}$  franchissement de la transition  $t_i$ . Les dateurs des transitions sources, notés  $t_u$ , sont les composantes du vecteur  $u(k) \in \mathbb{R}_{max}^m$ . L'équation suivante traduit la dynamique d'un graphe d'événements temporisés :

$$\theta(k) \geq \bigoplus_{l=0}^{M^{max}} A_l \cdot \theta(k-l) \oplus B_l \cdot u(k-l) \cdot u(k). \quad (1.3)$$

Dans le cadre de la représentation Max-plus, la matrice  $A_l \in \mathbb{R}_{max}^{n \times n}$  contient les valeurs de temporisation des places. Chaque élément de cette matrice,  $A_{lij}$ , correspond à la temporisation de la place  $p_{ij}$ , notée  $\tau_{ij}$ . Si la place  $p_{ij}$  n'existe pas, le terme  $A_{lij}$  est défini comme  $\varepsilon$ . Par ailleurs, la matrice  $B_l \in \mathbb{R}_{max}^{n \times m}$  est utilisée pour représenter les temporisations des places associées aux transitions source (de commande)  $t_u$ .

Nous nous concentrons spécifiquement à l'évolution au plus tôt des graphes d'événements temporisés. Cette évolution correspond à la solution minimale dans  $\mathbb{R}_{max}$  de l'inéquation 1.3. Cette solution est déterminée par une équation linéaire.

$$\theta(k) = \bigoplus_{l=0}^{M^{max}} A_l \cdot \theta(k-l) \oplus B_l \cdot u(k-l). \quad (1.4)$$

Cette forme implicite peut s'écrire sous la forme explicite :

$$\theta(k) = \bigoplus_{l=0}^{M^{max}} A_l^* \cdot A_l \cdot \theta(k-l) \oplus A_l^* \cdot B_l \cdot u(k-l), \quad (1.5)$$

où  $A_l^*$  est la matrice obtenue en appliquant l'opération d'étoile de Kleene à la matrice  $A_l$ .

### Équation d'état dans $\mathbb{R}_{max}$

La méthode pour obtenir un modèle d'état consiste à ajouter des places et des transitions pour obtenir un graphe d'événements temporisés sauf. Nous décomposons chaque place  $p_i$  en  $m_{0_i}$  places marquées à 1, et nous ajoutons  $m_{0_i} - 1$  transitions intermédiaires.

Nous avons associé des dateurs à ces transitions intermédiaires, qui sont les composantes d'un vecteur  $\bar{\theta}(k) \in \mathbb{R}_{max}^{\bar{n}}$ , et nous notons par,  $x(k)$ , le vecteur d'état résultant :

$$x(k) = \begin{bmatrix} \theta(k) \\ \bar{\theta}(k) \end{bmatrix}, \quad (1.6)$$

qui appartient à  $\mathbb{R}_{max}^N$ , avec  $N = n + \bar{n}$ . Le comportement dynamique du graphe d'événements temporisés étendu est décrit par une équation de la forme implicite :

$$x(k) = A_0 \cdot x(k) \oplus A_1 \cdot x(k-1) \oplus B_0 \cdot u(k), \quad (1.7)$$

qui peut s'écrire sous la forme explicite :

$$x(k) = A \cdot x(k-1) \oplus B \cdot u(k), \quad (1.8)$$

avec  $A = A_0^* \cdot A_1$  et  $B = A_0^* \cdot B_0$ .

**Propriété 1.1** *Considérons un graphe d'événements temporisés dont le comportement est décrit par l'équation (1.8). Cette équation est équivalente à la formulation d'état suivante :*

$$x(k) = A^\phi \cdot x(k-\phi) \oplus \bigoplus_{l=0}^{\phi-1} A^l \cdot B \cdot u(k), \quad (1.9)$$

pour tout  $\phi$  entier tel que  $\phi \geq 1$ .

**Démonstration 1.2** Nous allons démontrer cette propriété par récurrence. Pour un graphe d'événements temporisés dont le comportement est décrit par l'équation d'état, la vérification de cette propriété est évidente lorsque  $\phi = 1$ . Supposons maintenant qu'elle soit également vérifiée pour  $\phi = l$ , c'est-à-dire

$$x(k) = A^l \cdot x(k-l) \oplus \bigoplus_{l=0}^l A^{l-1} \cdot B \cdot u(k-l), \quad (1.10)$$

Étant donné que l'hypothèse de l'équation d'état est vérifiée, nous avons donc :

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

En remplaçant  $x(k-l)$  par son expression dans 1.10, nous obtenons alors :

$$x(k) = A^{l+1} \cdot x(k-(l+1)) \oplus \bigoplus_{l=0}^l A^l \cdot B \cdot u(k-l). \quad (1.12)$$

Nous montrons alors que la propriété est satisfaite pour  $\phi = l+1$ , donc elle est vraie pour tout  $\phi \geq 1$ . □

**Exemple 1.3** Considérons l'exemple illustré dans la Figure 1.11. Pour obtenir les équations d'état correspondantes, nous devons décomposer la place  $p_2$ , qui contient deux jetons, en deux places distinctes avec un marquage de 1 et une transition intermédiaire  $t_3$ , comme illustré dans la figure 1.12. À chaque transition  $t_i$  est associée une fonction dateur  $\theta_i(k)$ , tandis que  $u(k)$  représente la fonction dateur associée à la transition source. Dans le cas d'un franchissement au plus tôt, les fonctions dateurs doivent satisfaire les équations suivantes :

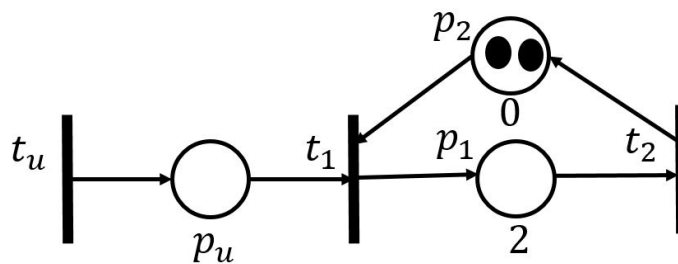
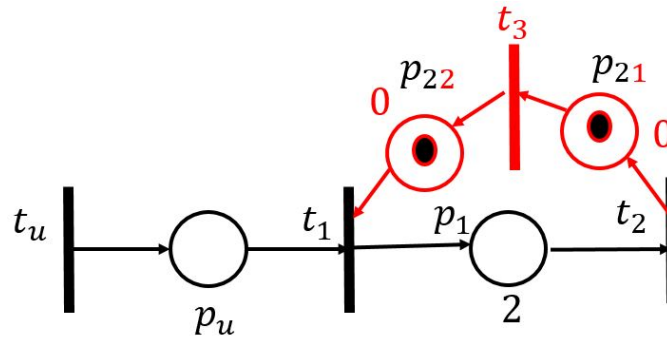


FIGURE 1.11: Modèle de GET

FIGURE 1.12: Décomposition de la place  $p_2$ 

$$\begin{cases} \theta_1(k) = \theta_3(k-1) \oplus u(k) \\ \theta_2(k) = 2 \cdot \theta_1(k) \\ \theta_3(k) = 1 \cdot \theta_2(k-1) \end{cases} \quad (1.13)$$

L'équation donne :

$$\theta(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot \theta(k-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix} \cdot u(k). \quad (1.14)$$

L'équation 1.14 peut être formulée de manière explicite de la façon suivante :

$$\theta(k) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & 2 & \varepsilon \end{bmatrix} \cdot \theta(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 2 \end{bmatrix} \cdot u(k), \quad (1.15)$$

### 1.4.3 Fonction compteur, domaine temporel

Chaque transition  $t_i$  du GET est associée à une fonction du temps  $t$ , qui représente le nombre cumulé de franchissements de cette transition jusqu'à l'instant  $t$ . Cette fonction est appelée compteur. Les transitions de commande sont désignées par la notation  $t_u$ , où le paramètre  $u$  est utilisé pour différencier les transitions de commande des autres transitions du graphe. Les fonctions compteurs correspondants aux transitions de commande sont les composantes du vecteur  $u(t) \in \mathbb{R}_{min}^m$ , tandis que les fonctions compteurs des autres transitions sont regroupés dans le vecteur  $\theta(t) \in \mathbb{R}_{min}^n$ . Le comportement d'un graphe d'événements temporisé est exprimé par

l'inéquation suivante :

$$\theta(t) \leq \bigoplus_{l=0}^{\tau^{max}} A_{\tau} \cdot \theta(t - \tau) \oplus B_{\tau} \cdot u(t - \tau) \quad (1.16)$$

$A_{\tau_{ij}}$  est la composante de la matrice  $A_{\tau} \in \mathbb{R}_{min}^{n \times n}$  est égal à  $\tau_{ij}$  qui correspond à la temporisation de la place  $p_{ij}$ . Si cette place n'existe pas le terme  $A_{\tau_{ij}}$  est égale à  $\varepsilon$ .  $B_{\tau} \in \mathbb{R}_{max}^{n \times m}$  correspond aux temporisations des places en aval des transitions de commande  $t_u$ . En général, on s'intéresse à l'évolution au plus tôt des graphes d'événements temporisés, c'est-à-dire qu'une transition est franchie dès qu'elle est franchissable. Cette évolution est caractérisée par la solution maximale de l'inéquation précédente dans l'ensemble  $\mathbb{R}_{min}$ . Cette solution est obtenue en résolvant l'équation :

$$\theta(t) = \bigoplus_{l=0}^{\tau^{max}} A_{\tau} \cdot \theta(t - \tau) \oplus B_{\tau}(t - \tau) \cdot u(t). \quad (1.17)$$

La forme implicite (1.17) peut être remplacée par la forme explicite suivante :

$$\theta(t) = \bigoplus_{l=0}^{\tau^{max}} A_0^* \cdot A_{\tau} \cdot \theta(t - \tau) \oplus A_0^* \cdot B_{\tau} \theta(t - \tau) \cdot u(t) \quad (1.18)$$

où  $A_0^*$  est l'étoile de Kleene de la matrice  $A_0$  définie précédemment.

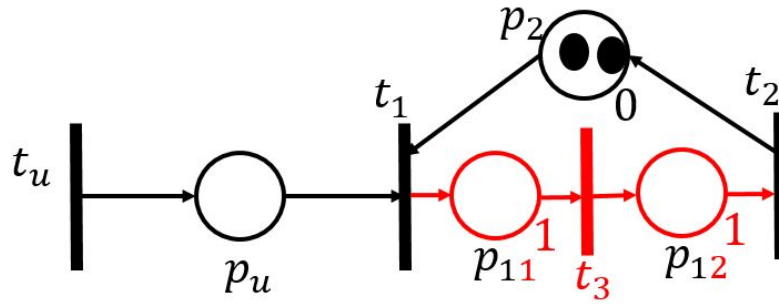
### Équation d'état dans $\mathbb{R}_{min}$

Afin de créer un modèle d'état, nous procédons à la décomposition de toutes les places dont la temporisation est supérieure à 1 (c'est-à-dire  $\tau > 1$ ) en  $\tau$  places marquées et temporisées à 1. Cela implique l'ajout de  $\tau - 1$  transitions intermédiaires au modèle. La somme des jetons de ces  $\tau$  places correspond au marquage initial de la place décomposée. Les transitions intermédiaires associées sont dotées de compteurs, représentés par les composantes d'un vecteur  $\bar{\theta}(t) \in \mathbb{R}_{min}^{\bar{n}}$ . Le vecteur résultant, noté  $x(t)$ , appartient à  $\mathbb{R}_{min}^{N \times N}$  et est défini comme suit, où  $N = n + \bar{n}$  :

$$x(t) = \begin{bmatrix} \theta(t) \\ \bar{\theta}(t) \end{bmatrix}. \quad (1.19)$$

Le comportement dynamique du graphe d'événements temporisé étendu est décrit par l'équation implicite.

$$x(t) = A_0 \cdot x(t) \oplus A_1 \cdot x(t - 1) \oplus B_0 \cdot u(t). \quad (1.20)$$

FIGURE 1.13: Décomposition de la place  $p_1$  de GET

Cette équation peut également être exprimée sous la forme explicite.

$$x(t) = A \cdot x(t-1) \oplus B \cdot u(t), \quad (1.21)$$

avec  $A = A_0^* \cdot A_1$  et  $B = A_0^* \cdot A_1$ .

**Propriété 1.2** *Considérons un graphe d'événements temporisé dont le comportement est décrit par l'équation (1.21). Cette équation est équivalente à la formulation d'état suivante :*

$$x(t) = A^\tau \cdot x(t-\tau) \oplus \bigoplus_{K=0}^{\tau-1} A^K \cdot B \cdot u(t) \quad (1.22)$$

La démonstration de cette propriété est similaire à celle de la propriété 1.1.

**Exemple 1.4** *Prenons le même exemple de la Figure 1.11 ; dans le but d'avoir des fonctions compteur sous forme d'état il faut décomposer la place  $p_1$  en deux places avec une temporisation égal à 1 selon la Figure 1.13.*

Chaque transition  $t_i$ , où  $i$  varie de 1 à 3, est associée à une fonction dateur  $\theta_i(k)$ , tandis que la fonction compteur associée à la transition de commande est notée  $u(k)$ . Pour un franchissement au plus tôt, ces fonctions compteurs doivent satisfaire les équations suivantes :

$$\begin{cases} \theta_1(k) = \theta_2(k-1) \oplus u(k) \\ \theta_2(k) = 1 \cdot \theta_1(k) \\ \theta_3(k) = 1 \cdot \theta_2(k) \end{cases} \quad (1.23)$$

L'équation (1.23) se traduit par :

$$\theta(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot \theta(k-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix}. \quad (1.24)$$

L'équation explicite s'exprime de la manière suivante :

$$\theta(k) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & 2 & \varepsilon \end{bmatrix} \cdot \theta(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 2 \end{bmatrix} \cdot u(k). \quad (1.25)$$

## 1.5 Conclusion

Dans ce premier chapitre, nous avons exposé les outils mathématiques utilisés dans cette étude. Nous avons défini certains concepts et présenté des propriétés relatives aux dioïdes. Ensuite, nous avons introduit les graphes d'événements temporisés en se référant brièvement aux réseaux de Petri. Par la suite, nous avons démontré comment obtenir des modèles linéaires dans les algèbres Max-plus et Min-plus à partir d'un graphe d'événements temporisés.

Dans le chapitre qui suit, nous aborderons la modélisation des graphes d'événements temporisés avec conflit.

## Chapitre 2

# Modélisations des graphes d'événements temporisés avec conflit

### 2.1 Introduction

*Les réseaux de Petri avec des ressources partagées sont étendant des RdPs classiques qui permettent de modéliser des systèmes où des ressources sont partagées entre plusieurs processus. Une ressource est représentée par un jeton, et plusieurs processus peuvent demander l'accès à cette ressource au même temps. Les réseaux de Petri avec partage de ressources peuvent être utilisés pour détecter les problèmes de planification des ressources et pour améliorer les performances du système. Une classe plus générale que les GETs est la classe des réseaux de graphes d'événements temporisés avec conflit (RGETCs). Cette classe permet de modéliser le partage de ressources avec un certain nombre d'hypothèses. De nombreux travaux dans la littérature se sont focalisés sur l'analyse des réseaux de Petri avec partage de ressources. Dans [33], les auteurs ont proposé un modèle sous forme de GET des systèmes de fabrication flexibles avec compétition pour le partage de ressources telles que les job-shops et les flow-shops. Une méthodologie de modélisation et d'évaluation des performances, en utilisant les réseaux de Petri, est proposée. Les auteurs dans [34] se concentrent sur les systèmes de fabrication flexibles qui impliquent le partage de ressources. Leur approche consiste à construire de GET à partir d'un modèle de réseaux de Petri en dupliquant les transitions communes. Le travail pionnier de [25] présente une procédure de modélisation basée sur des tas de pièces qui est utilisée à des fins d'évaluation de performance. L'article se termine par l'introduction des automates Max-plus qui sont actuellement utilisés pour*

*l'analyse et la commande des systèmes à événements discrets [35, 36]. Dans [37, 38, 39], une classe des Systèmes Linéaires Max-plus à Commutation (SLMC) à été introduite. Cette classe est constituée des systèmes à événements discrets qui peuvent basculer entre différents modes de fonctionnement. Dans chaque mode, le système est décrit par une équation d'état linéaire Max-plus avec des matrices d'états différentes pour chaque mode. La commutation entre les différents modes est assurée par une fonction de commutation qui peut dépendre des signaux d'entrée, des états précédents et/ou d'un signal externe. Dans [40], les auteurs ont proposé un modèle de RGETC en utilisant un système d'équations linéaires Max-plus variant dans le temps alors que la politique d'allocation de ressources est arbitraire. L'évaluation des performances en termes de temps de cycle de cette classe est abordée dans l'article [41]. Les auteurs dans [42, 43, 44] proposent une modélisation et une synthèse des lois de commande d'une classe de RdPs présentant des phénomènes de ressources partagées. La méthode est basée sur le produit de Hadmard et la théorie de la résiduation.*

## 2.2 Les RdPs avec partage de ressources

### 2.2.1 Partage de ressources

*Le modèle illustré dans la Figure 2.1 représente une machine qui comprend un stock d'entrée  $p$ -entrée, une section centrale de la machine entre les transitions  $t_{in}$  et  $t_{out}$ , et un stock de sortie  $p$ -sortie. Chaque pièce arrivant sur le site est représentée par un jeton. Au début du processus, la pièce est dans le stock d'entrée lorsque la transition  $t_{in}$  est franchie, le jeton est placé dans la place machine, symbolisant ainsi que la pièce est traitée par la machine. Lorsque l'opération de l'usinage est effectuée, la transition  $t_{out}$  est franchie et le jeton représentant la place machine passe dans la place  $p$ -sortie, la machine est alors libre. Ensuite, la validation de la transition  $t_{in}$  dépend de la disponibilité d'une ressource, telle qu'un robot utilisé pour approvisionner la machine [28]. Le modèle est donc étendu de la manière suivante :*

*La transition  $t_{out}$  est franchie à condition que le robot soit disponible (ce qui est le cas ici puisqu'il y a un jeton dans la place robot). Le robot est alors occupé par la machine et n'est alors pas disponible pour aucune autre machine. Lorsque la machine a fini son opération,  $t_{out}$  est franchie, un jeton est restitué dans la place robot, ce qui signifie qu'il est à nouveau disponible. Un autre*

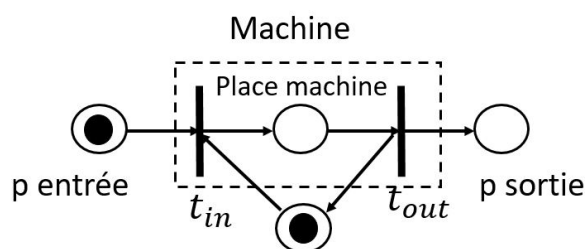


FIGURE 2.1: Une machine nécessitant une ressource

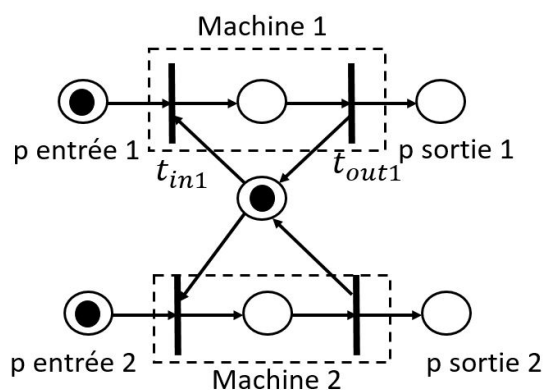


FIGURE 2.2: Deux machines en concurrence

jeton apparaît à la place  $p$ -sortie, représente une pièce qui sort de la machine. Le jeton situé dans la place machine disparaît. Afin de représenter maintenant la partage d'un robot par deux machines, le modèle de la Figure 2.2, est développé. Les deux machines sont représentées par la Figure 2.2, constitués toutes les deux par place d'entrée (stock d'entrée), place de sortie (stock de sortie) et place machine proprement dite.

D'autres configurations peuvent également être envisagées, par exemple une tâche nécessite l'utilisation simultanée de plusieurs ressources partagées. Les montages deviennent plus spécifiques mais sont toujours construits à l'aide des mêmes sous modèles élémentaires. Dans la Figure 2.3, par exemple, la machine 2 partage deux ressources avec deux autres machines.

### 2.3 Modélisation des RGETC

Dans ce qui suit, nous présenterons un aperçu des méthodes de modélisation existantes pour la modélisation de tels graphes.

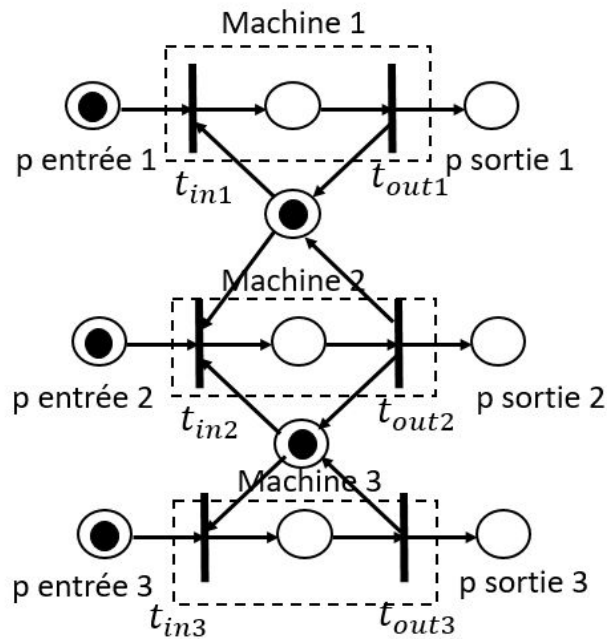


FIGURE 2.3: Machine à double concurrence

### 2.3.1 Méthode de Addad

Dans cette partie, nous présenterons la méthode suggérée par les auteurs dans leur publication [45].

Soit le réseau de graphes d'événements temporisé avec conflits représenté à la Figure 2.4 comprenant  $N$  graphes d'événements  $G_i$  ( $1 \leq i \leq N$ ) en concurrence pour consommer le jeton de la place de conflit  $\tilde{p}$  et dont la temporisation est  $\tilde{\tau}$ . Posons les notations suivantes :

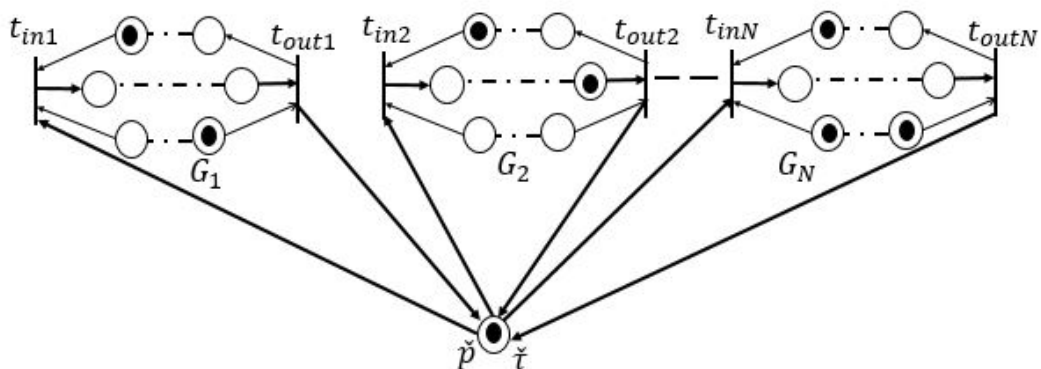


FIGURE 2.4: Réseau de Graphes d'Evénements Temporisés avec Conflits

- $t_{in_i}$  la transition unique située en amont de  $G_i$  et la transition située en aval de  $G_i$  notée  $t_{out_i}$ , comme le montre la figure 2.4.
- Chaque GET  $G_i$  est représenté par son équation d'état :

$$\theta_i(k+1) = A_i \cdot \theta(k) + B_i \cdot u(k+1), \quad (2.1)$$

**Remarque 2.1** Dans cette méthode, on considère que la place de conflit est une place sauf. La ressource ne peut être utilisée que par un seul utilisateur à la fois.

La ressource est géré par une séquence  $S$  par exemple ( $S = G_1 G_2 G_3 \cdots G_N$ ).

### Équations Max-plus récurrentes des RGETC

Chaque transition  $t_{in_i}$  et  $t_{out_i}$  sont associées aux dateurs  $\theta_{in}^i(k_i)$  et  $\theta_{out}^i(k_i)$ , en plus, du dateur associé à la place  $\tilde{p}$ . La date de disponibilité des jetons dans la place de conflit, pour la  $k^{\text{ème}}$  fois, est représentée par  $\psi(k)$ . Cette date indique quand les jetons entrant dans la place  $\tilde{p}$  est disponible. Supposons, sans perte de généralité, que la première composante du vecteur  $\theta_{in}^i$  soit le dateur associé à la transition  $t_{in_i}$ . Si le  $k^{\text{ème}}$  jeton entrant dans  $\tilde{p}$  est consommé par le GET  $G_i$  et que celui-ci a déjà consommé  $(k_i - 1)$  jetons de  $\tilde{p}$ , alors l'équation pour la  $i^{\text{ème}}$  composante  $\theta_{in}^i(k_i + 1)$ , est donnée par :

$$\theta_{in}^i(k_i + 1) = \bigoplus_{\delta=1}^N A_{1\delta}^N \cdot \theta_{\delta}(k_i) \oplus \bigoplus_{\delta=1}^Z B_{1\delta} \cdot u(k_i + 1) \oplus \psi(k + 1) \quad (2.2)$$

Puisque la date de franchissement des autres transitions  $\theta_{out}^i(k_i)$  ne dépend pas des jetons dans  $\tilde{p}$  (ces transitions n'étant pas en aval de  $\tilde{p}$ ), donc la composante  $\theta_{out}^i(k_i)$  ne dépend pas de la date  $\psi(k)$ , elle s'écrit comme suit :

$$\theta_{out}^i(k_i + 1) = \bigoplus_{\delta=1}^N A_{n\delta}^N \cdot \theta_{\delta}(k_i) \oplus \bigoplus_{\delta=1}^Z B_{n\delta} \cdot u(k_i + 1) \quad (2.3)$$

La combinaison entre (2.2) et (2.3), donne la forme matricielle suivante :

$$\theta_{out}^i(k_i + 1) = A_i \cdot \theta_{\delta}(k_i) \oplus (B)_i \cdot u(k_i + 1) \oplus F_i \cdot \psi(k + 1) \quad (2.4)$$

Où la matrice est  $F_i$  de dimension  $n_i \times 1$ .

De plus, lorsqu'un jeton de la place est consommé par le GET  $G_i$ , le premier jeton de  $\tilde{p}$  est généré par franchissant de la transition  $t_{in_i}$  pour la  $k^{\text{ème}}$  fois. Supposons que la dernière

composante du vecteur  $\theta_{out}^i$  soit le dateur associé à  $t_{out_i}$  (c'est-à-dire  $\theta_{out}^i$ ). Ainsi, l'expression pour le dateur  $\psi(k+1)$  est la suivante :

$$\psi(k+1) = \tilde{\tau} \cdot \theta_{out}^i(k_i) \quad (2.5)$$

Ceci peut se traduire sous forme matricielle :

$$\psi(k+1) = G_i \cdot \theta_{out}^i(k_i) \quad (2.6)$$

Où  $G_i$  est une matrice de dimensions  $1 \times n_i$ . Les équations Max-plus récurrentes suivantes sont vérifiées pour  $1 \leq i \leq N$  et  $k = \sum_{i=1}^N (k_i - 1)$  :

$$\begin{aligned} \theta_i(k_i+1) &= A_i \cdot \theta_\delta(k_i) \oplus (B)_i \cdot u(k_i+1) \oplus F_i \cdot \psi(k+1) \\ \psi(k+1) &= \tilde{\tau} \cdot \theta_{out}^i(k_i) \end{aligned} \quad (2.7)$$

Afin d'illustrer la méthode, nous pouvons prendre l'exemple de la Figure 2.5

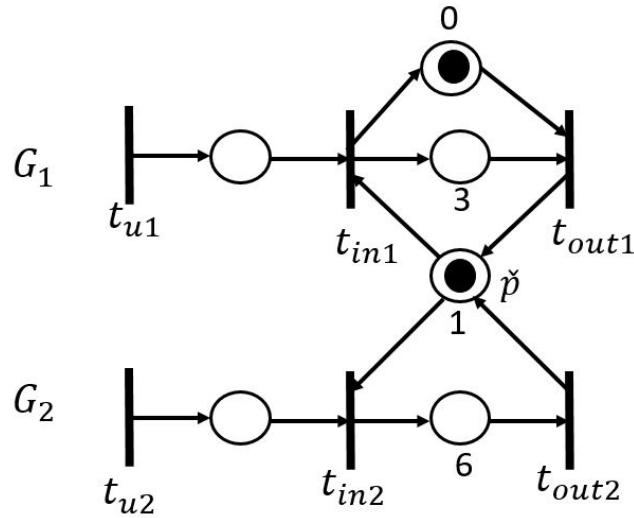


FIGURE 2.5: Exemple de RGETC pour illustrer la méthode de Addad

Nous associons à chaque transition  $t_{in1}, t_{in2}, t_{out1}, t_{out2}$  successivement les dateurs  $\theta_{in}^1, \theta_{in}^2, \theta_{out}^1, \theta_{out}^2$ , les deux GETs sont représentés par :

$$G1 : \begin{cases} \theta_{in}^1(k_1+1) = u_1(k_1) \oplus \theta_{out}^1(k_1) \\ \theta_{out}^1(k_1+1) = 3 \cdot \theta_{in}^1(k_1+1) \end{cases} \quad (2.8)$$

$$G2 : \begin{cases} \theta_{in}^2(k_2+1) = u_2(k_2) \\ \theta_{out}^2(k_2+1) = 6 \cdot \theta_{in}^2(k_2+1) \end{cases} \quad (2.9)$$

Si on prend la séquence de fonctionnement  $S = G_1 G_2$ , donc le jeton de conflit est alloué d'une manière périodique. Si le  $k_i^{\text{ème}}$  jeton de  $\tilde{p}$  est consommé par le GET  $G_1$  et que ce dernier a déjà consommé  $(k_i - 1)^{\text{ème}}$  par GET  $G_2$ , nous avons :

$$G1 : \begin{cases} \theta_{in}^1(k_1 + 1) = u_1(k_1) \oplus \theta_{out}^1(k_1) \oplus \psi(k_1 + 1) \\ \theta_{out}^1(k_1 + 1) = 3 \cdot \theta_{out}^1(k_1) \oplus 3 \cdot u_1(k_1) \oplus 3 \cdot \psi(k_1 + 1) \\ \psi(k_1 + 1) = 1 \times \theta_{out}^2(k_2) \end{cases} \quad (2.10)$$

$$G2 : \begin{cases} \theta_{in}^1(k_2 + 1) = u_2(k_2) \oplus \psi(k_2 + 1) \\ \theta_{out}^2(k_2 + 1) = 6 \cdot u_2(k_2) \oplus 6 \cdot \psi(k_2 + 1) \\ \psi(k_2 + 1) = 1 \times \theta_{out}^1(k_1) \end{cases} \quad (2.11)$$

Le système d'équations (2.10) et (2.11) peut être réécrit sous la forme suivante :

$$A_1 = \begin{bmatrix} e & \varepsilon \\ 3 & \varepsilon \end{bmatrix}; B_1 = \begin{bmatrix} e \\ 3 \end{bmatrix}; F_1 = \begin{bmatrix} e \\ 3 \end{bmatrix}; G_1 = \begin{bmatrix} \varepsilon & 1 \end{bmatrix}.$$

$$A_2 = \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}; B_2 = \begin{bmatrix} e \\ 6 \end{bmatrix}; F_2 = \begin{bmatrix} e \\ 6 \end{bmatrix}; G_2 = \begin{bmatrix} 1 & \varepsilon \end{bmatrix}.$$

### 2.3.2 Méthode de Moradi

Dans cette partie, nous allons présenter la modélisation des RdPs avec des ressources partagées par la méthode proposée par les auteurs dans [42].

En considérant un système dans lequel différents utilisateurs (sous-systèmes) partagent un nombre limité de ressources (égales  $N$ ). Pour des raisons de simplicité, les auteurs ont supposé que chaque sous-système est un GET, noté  $G_i$ , comme le montre la Figure 2.6 .

- $N$  représente le nombre des GETs qui partage avec une place de conflit  $\tilde{p}$  avec  $m_{\tilde{p}}$  ressources (jetons).
- $\tilde{\tau}$  représente le temps de séjour dans la place de conflit.
- L'allocation d'une ressource par un sous-système  $G_i$  est modélisé par le franchissement de la transition  $t_{in_i}$ .
- A chaque transition  $t_{in_i}, t_{out_i}, t_{y_i}$  est associée, successivement, une fonction compteur  $\theta_{in}^K(t), \theta_{out}^K(t), u^K(t)$ .

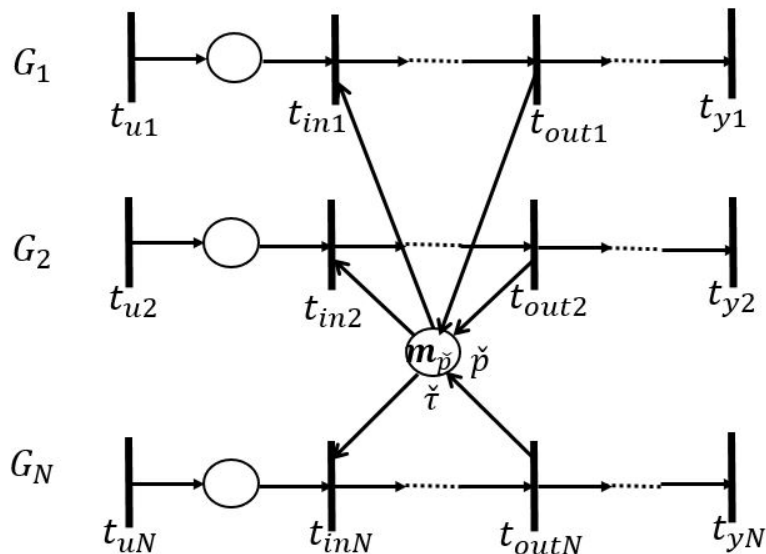


FIGURE 2.6: Un RdP avec des ressources partagées

Négliger le partage des ressources (ce qui signifie que pour chaque sous-système, on ignore tous les arcs venant d'autres sous-systèmes et se terminent dans la place partagée et ignorer tous les arcs qui commencent dans la place partagée et se termine dans d'autres sous-systèmes) et en considérant la règle du tir, le  $K^{\text{ème}}$  sous-système peut être modélisé dans l'algèbre Min plus comme suit :

$$G_k : \begin{cases} \theta^K(t) = A_1^K \theta^K(t-1) \oplus \dots \oplus A_T^K \theta^K(t-T) \oplus B_0^K u^K(t) \oplus \dots \oplus A_H^K u^K(t-H) \\ y^K(t) = C^K \theta^K(t) \end{cases} \quad (2.12)$$

### Modélisation du partage de ressource avec la politique prioritaire

**Propriété 2.1** La propriété d'activation de sous système  $G_K$  est supérieure à celle de  $G_{K+1}$ , pour chaque  $K = 0 \dots N$ .

La vérification de cette propriété donne l'équation récursive suivante qui détermine le nombre de ressources disponibles pour  $G_k$  au moment  $t$  :

$$\begin{cases} a^K(t) = a^{K-1}(t) \cdot (\theta_{in}^{K-1}(t-1) - \theta_{in}^{K-1}(t)) \text{ pour } K = 2, \dots, N \\ a^1(t) = [\otimes_{K=1}^N \theta_{out}^K(t - \tilde{\tau}) - \theta_{in}^1(t-1)] \cdot n \end{cases} \quad (2.13)$$

$a^1$  représente le nombre des ressources disponibles pour le sous-système  $G_1$  à l'instant  $t$ , qui est égal au nombre initial de ressources  $n$  moins le nombre de ressources déjà utilisées et donc

non disponibles.

La quantité  $\theta_{out}^K(t - \tilde{\tau}) - \theta_{in}^K(t - 1)$  représente le nombre de ressources libérées par  $G_k$  jusqu'au temps  $t - \tilde{\tau}$  moins le nombre de ressources allouées par  $G_k$  jusqu'au temps  $t - 1$ .  $a^K(t)$ , pour  $k = 2, \dots, N$  représente le nombre de ressources disponibles pour  $G_K$  à l'instant  $t$ . Il est donné par  $a^{K-1}(t)$  moins le nombre de ressources nouvellement allouées par  $G_{K-1}$  à l'instant  $t$ . La description de la dynamique de chaque sous système dépend du signal  $a^K(t)$ . Donc, l'évolution du sous système  $G_k$  est donnée par :

$$G_k : \begin{cases} \theta^K(t) = (A_1^K \oplus A_{a(t)}^K) \theta^K(t-1) \oplus \dots \oplus A_T^K \theta^K(t-T) \oplus B_0^K u^K(t) \oplus \dots \oplus A_H^K u^K(t-H) \\ y^K(t) = C^K \theta^K(t) \end{cases} \quad (2.14)$$

Où

$$(A_{a(t)}^K)_{i,j} = \begin{cases} a^K(t) & \text{si } i = j \\ \varepsilon & \text{sinon} \end{cases} \quad (2.15)$$

pour illustrer la méthode présentée on considère le système de la Figure 2.7 sous la politique de priorité. Nous associons à chaque transition  $t_{in_i}, t_{out_i}$  et  $t_{y_i}$  les fonctions compteurs  $\theta_1^i(t)$ ,  $\theta_2^i(t)$  et  $y^i(t)$ . En appliquant la méthode de modélisation présentée, nous obtenons les systèmes d'équations (2.16), (2.17) et (2.18) suivant :

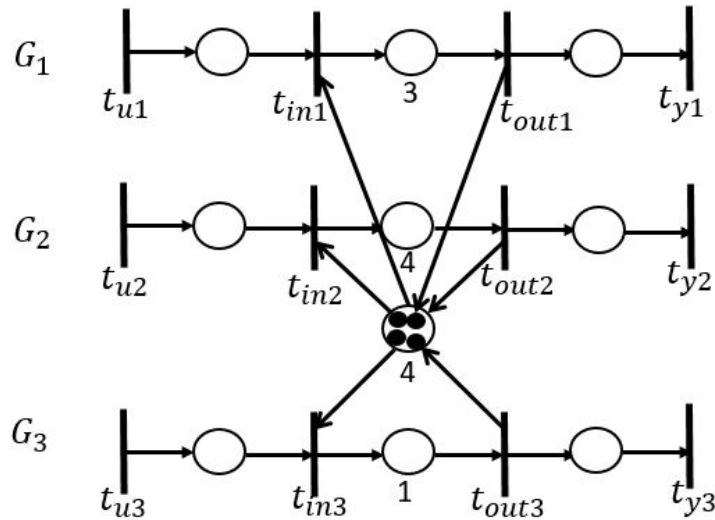


FIGURE 2.7: Un RDP avec partage de ressources pour illustrer la méthode de Moradi

$$G_1 : \begin{cases} \theta_1^1(t) = u_1(t) \oplus a_1(t) \\ \theta_2^1(t) = \theta_1^1(t - 3) \\ y^1(t) = \theta_2^1(t) \end{cases} \quad (2.16)$$

$$G_2 : \begin{cases} \theta_1^2(t) = u_2(t) \oplus a_2(t) \\ \theta_2^2(t) = \theta_1^2(t - 4) \\ y^2(t) = \theta_2^2(t) \end{cases} \quad (2.17)$$

$$G_3 : \begin{cases} \theta_1^3(t) = u_3(t) \oplus a_3(t) \\ \theta_2^3(t) = \theta_1^3(t - 1) \\ y^3(t) = \theta_2^3(t) \end{cases} \quad (2.18)$$

Où :

$$\begin{cases} a_1(t) = 4 \cdot (\theta_2^1(t - 4) - \theta_1^1(t - 4)) \cdot \\ (\theta_2^2(t - 4) - \theta_1^2(t - 4)) \cdot (\theta_2^3(t - 4) - \theta_1^3(t - 4)) \\ a_2(t) = a_1(t) \cdot (\theta_1^1(t - 1) - \theta_1^1(t)) \\ a_3(t) = a_2(t) \cdot (\theta_1^2(t - 1) - \theta_1^2(t)) \end{cases} \quad (2.19)$$

### 2.3.3 Modélisation des RGETCs par les systèmes linéaires Max-plus à commutation (méthode proposée)

Dans cette partie, nous proposons une méthode de modélisation basée sur les modèles SLMC qui sera pertinente pour la suite de cette thèse : [21, 22].

Pour pouvoir modéliser les RGETCs, on introduit deux types d'objets qui permettent de faciliter la modélisation.

**Définition 2.2** [22] Un commutateur arrière ( $B$ ) est défini par une place de conflit vide  $\tilde{p}$  avec plusieurs transitions d'entrée  $t_j^{in}$  et une seule transition de sortie  $t^{out}$ . Si la transition de sortie  $t^{out}$  est une conséquence du franchissement de la transition d'entrée  $t_j^{in}$ , on dit que le couple  $(t_j^{in}, t^{out})$  détermine la dynamique du réseau de Petri, si le tir de la transition  $t^{out}$  est une conséquence du tir de la transition  $t_j^{in}$  (Figure 2.8).

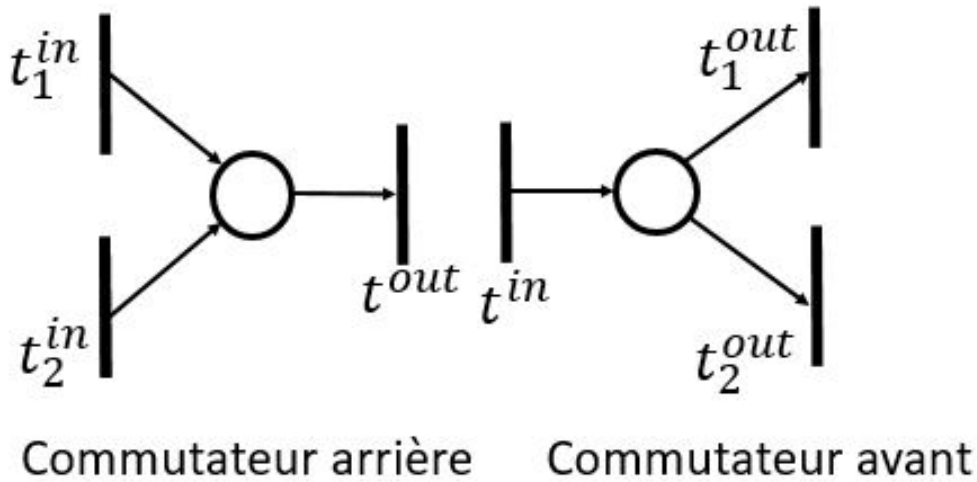


FIGURE 2.8: Comutateur avant et arrière

**Remarque 2.3** Un commutateur arrière exerce une fonction semblable à celle d'un multiplexeur dans les circuits numériques.

**Définition 2.4** [22] Un commutateur avant est défini par une place de conflit vide  $\tilde{p}$  qui possède une unique transition d'entrée  $t^{in}$  et plusieurs transitions de sortie  $t_j^{out}$ . On dira que le couple  $(t^{in}, t_j^{out})$  détermine la dynamique d'un réseau de Petri, si le tir de la transition d'entrée  $t^{in}$  est suivi du tir de la transition de sortie  $t_j^{out}$  (Figure 2.8).

**Remarque 2.5** Un commutateur avant exerce une fonction semblable à celle un décodeur dans les circuits numérique.

**Définition 2.6** [22] Une configuration  $C_i$  d'un réseau de Petri  $\mathcal{N}$  correspond à une position donnée des commutateurs.

Soit  $\mathcal{C} = \{C_1, C_2, \dots\}$  l'ensemble des configurations possibles des commutateurs (**B** et **F**) d'un RdP donné. On peut toujours représenter une place de conflit par la mise en série d'un commutateur (**B** et **F**), comme le montre la Figure 2.9.

En plus de la notion de commutateur, nous présentons les concepts de dateur généralisé et de système linéaire Max-plus à commutations. Ces concepts revêtent une importance capitale pour la compréhension des résultats exposés par la suite dans ce mémoire de thèse.

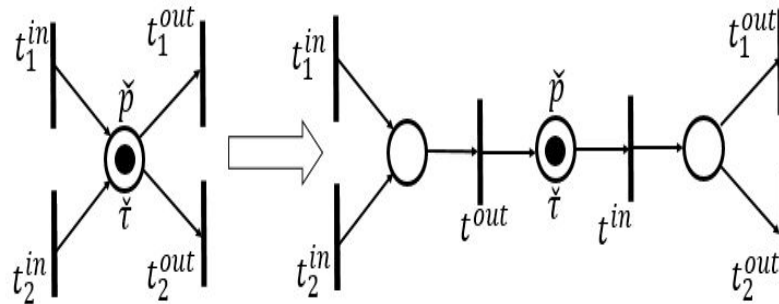


FIGURE 2.9: Exemple de configuration

**Définition 2.7** Un dateur généralisé  $\theta_i(k)$  est une fonction associée à chaque transition  $t_i$  du RGETC non décroissante représentant la date du dernier franchissement de la transition  $t_i$  lors de la  $k^{\text{ème}}$  exécution du graphe.

**Remarque 2.8** La notion de dateur généralisé a été définie par les auteurs mentionnés dans [46].

L'exemple de la Figure 2.10 illustre cette notion de dateur généralisé.

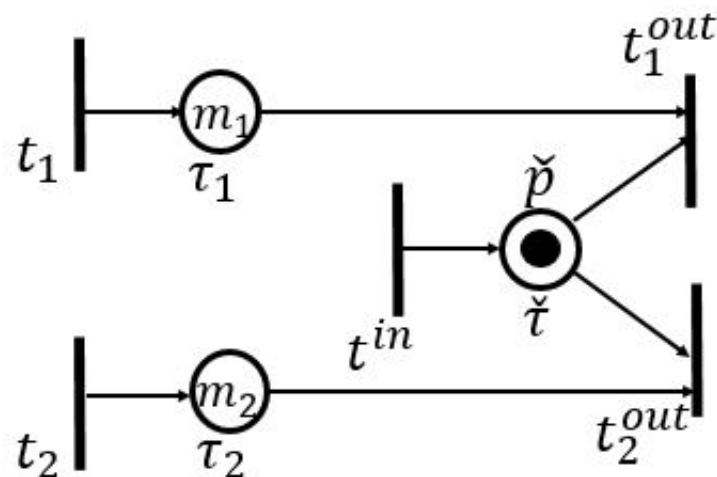


FIGURE 2.10: Dateur généralisé

L'expression du dateur généralisé associée à la transition  $t_1^{\text{out}}$  est donné par :

$$\theta_1^{out}(k+1) = \begin{cases} \tau_1 \cdot \theta_1(k - m_1) \oplus \tilde{\tau} \cdot \theta^{in}(k), & \text{si le jeton est alloué à } t_1^{out} \\ \theta_1^{out}(k), & \text{autrement} \end{cases} \quad (2.20)$$

**Définition 2.9** [37] *Les systèmes Max-plus linéaires à commutations sont des systèmes à événements discrets qui peuvent passer d'un mode de fonctionnement à un autre. Dans chaque mode, le système est décrit par un système linéaire dans l'algèbre Max-plus.*

$$\boldsymbol{\theta}(k+1) = A^{\gamma(k)} \cdot \boldsymbol{\theta}(k) \oplus B^{\gamma(k)} \cdot \mathbf{u}(k+1) \quad (2.21)$$

Dans le Théorème qui suit, nous avons démontré que le comportement dynamique des RGETCs se modélise par des systèmes Max-plus linéaires à commutations.

**Théorème 2.10** [22] *Le comportement dynamique d'un RGETC est un système Max-plus linéaire à commutations. Chaque configuration  $\mathcal{C}_i$  des commutateurs définit un mode de fonctionnement. La dynamique dans chaque mode est représentée par l'équation Max-plus linéaire ci-dessous :*

$$\boldsymbol{\theta}(k+1) = A_0^{\gamma(k)} \cdot \boldsymbol{\theta}(k+1) \oplus A_1^{\gamma(k)} \cdot \boldsymbol{\theta}(k) \oplus B_0^{\gamma(k)} \cdot \mathbf{u}(k+1), \quad (2.22)$$

où :

$\gamma(k) \in \{1, \dots, N\}$  est une fonction qui donne l'indice du mode ;

$N$  est le nombre total des modes ;

$\boldsymbol{\theta}(k)$  est le vecteur d'état composé des dateurs généralisés  $\theta_i(k)$  ;

$\mathbf{u}(k)$  est le vecteur de commande ;

$A_0^{\gamma(k)}, A_1^{\gamma(k)} \in \bar{R}_{max}^{\bar{n} \times \bar{n}}$  ;

$B_0^{\gamma(k)} \in \bar{R}_{max}^{\bar{n} \times \bar{m}}$ .

**Démonstration 2.1** *La démonstration de ce Théorème est simple, elle est basée sur la définition des dateurs généralisés et de la notion de configuration.*

**Remarque 2.11** *L'équation 2.22 peut être écrite sous forme explicite 2.21 en utilisant le Théorème de Kleene 1.17.*

**Remarque 2.12** Il convient de souligner que la configuration spécifique des commutateurs ne garantit pas automatiquement un fonctionnement correct des RGETCs . Pour assurer un fonctionnement approprié, il est essentiel de connaître une séquence de transitions réalisables qui permettra d'obtenir une séquence de configurations correctes.

**Exemple 2.1** Prenons l'exemple suivant représenté par la Figure 2.11. Cet exemple contient 2 GETs ( $G_1$  et  $G_2$ ) interconnectés par 2 ressources partagées. Pour cela, nous pouvons définir 4 configurations. Les configurations sont caractérisées par les séquences d'activations par GET,  $G_iG_j$ , et par les séquences de franchissement des transitions,  $t_k t_s$ .

1. ( $G_1G_1$ ) défini par la configuration  $C_2=(t_1t_1;t_2t_2)$ .

Dans la configuration  $C_1$ , signifie que le franchissement de la transition  $t_1$  est suivi du franchissement de la transition  $t_1$ , et le franchissement de la transition  $t_2$  est suivi du franchissement de la transition  $t_2$ . Donc, après l'activation de  $G_1$ , nous aurons ensuite l'activation de  $G_2$ .

2. ( $G_2G_1$ ) défini par la configuration  $C_2=(t_3t_1;t_4t_2)$ .

3. ( $G_2G_2$ ) défini par la configuration  $C_3=(t_3t_3;t_4t_4)$ .

4. ( $G_1G_2$ ) défini par la configuration  $C_4=(t_1t_3;t_2t_4)$ .

Les 4 configurations donnent les modèles Max-plus linaires suivants :

$$C_1 : \begin{cases} \theta_1(k+1) = 1 \cdot \theta_1(k) \oplus u_1(t) \\ \theta_2(k+1) = 4 \cdot \theta_1(k) \oplus 1 \cdot \theta_2(k) \oplus 3 \cdot u_1(t) \\ \theta_3(k+1) = e \cdot \theta_3(k) \\ \theta_4(k+1) = e \cdot \theta_4(k) \end{cases} \quad (2.23)$$

$$C_2 : \begin{cases} \theta_1(k+1) = 1 \cdot \theta_3(k) \oplus u_1(t) \\ \theta_2(k+1) = 4 \cdot \theta_3(k) \oplus 1 \cdot \theta_4(k) \oplus 3 \cdot u_1(t) \\ \theta_3(k+1) = e \cdot \theta_3(k) \\ \theta_4(k+1) = e \cdot \theta_4(k) \end{cases} \quad (2.24)$$

$$C_3 : \begin{cases} \theta_1(k+1) = e \cdot \theta_1(k) \\ \theta_2(k+1) = e \cdot \theta_2(k) \\ \theta_3(k+1) = 1 \cdot \theta_3(k) \oplus u_2(k) \\ \theta_4(k+1) = 5 \cdot \theta_3(k) \oplus 1 \cdot \theta_4(k) \oplus 4 \cdot u_2(k) \end{cases} \quad (2.25)$$

$$C_4 : \begin{cases} \theta_1(k+1) = e \cdot \theta_1(k) \\ \theta_2(k+1) = e \cdot \theta_2(k) \\ \theta_3(k+1) = 1 \cdot \theta_1(k) \oplus u_2(k) \\ \theta_4(k+1) = 5 \cdot \theta_1(k) \oplus 1 \cdot \theta_2(k) \oplus 4 \cdot u_2(k) \end{cases} \quad (2.26)$$

Les configurations s'écrivent sous la forme 2.21 où :

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

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

La dynamique d'un RGETC modélisé par des systèmes linéaires Max-plus à commutation peut être représentée par un automate. De sorte que les états de l'automate correspondent aux Modes de configurations et l'ensemble de l'alphabet de l'automate correspondent aux positions des commutateurs, ce qui est illustré par la Figure 2.12 :

**Exemple 2.2** [21] Examinons le réseau du graphe d'événements temporisés avec conflit illustré dans la Figure 2.13. Il est composé de deux GET reliés par une place de conflit sauf. Les

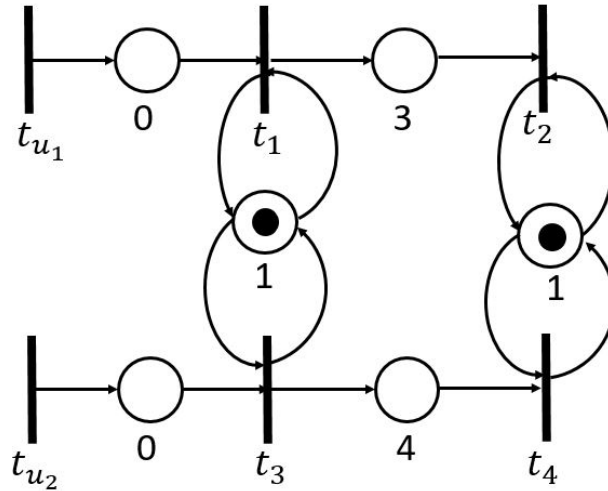


FIGURE 2.11: Un modèle de RGETC

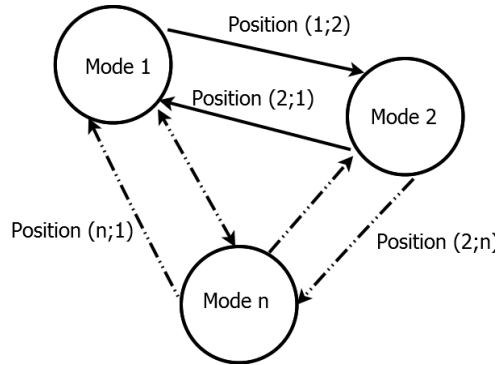


FIGURE 2.12: Automate à états finis des modes du RGETC

transitions sont notées  $t_i$  (où  $0 \leq i \leq 4$ ) et les places sont notées  $p_j$  (où  $0 \leq j \leq 4$ ). Les deux transitions de contrôles sont notées  $tu_1$  et  $tu_2$ .

On dénote par  $\theta_i(k)$  la fonction dateur associée à la transition  $t_i$  et  $\tau_i$  la temporisation associée à  $p_{ij}$  tel que ( $0 \leq i, j \leq 4$ ). Le fonctionnement de RGETC est représenté par l'automate illustré dans la Figure 2.14, qui se compose de quatre états  $Q_i$  où chaque état correspond à l'un des modes de configuration. Chaque configuration peut se modéliser par des équations à commutation Max-plus :

–MODE 1 : L'état  $Q1$  de l'automate, qui est généré par la configuration  $C_{41}$  (voir la Figure 2.14), correspond à une situation où le jeton de conflit est consommé par le  $GET_2$  lors du  $(\mathbf{k})^{\text{ème}}$  franchissement, puis par le  $GET_1$  lors du  $(\mathbf{k} + 1)^{\text{ème}}$  franchissement. Cela est dû au

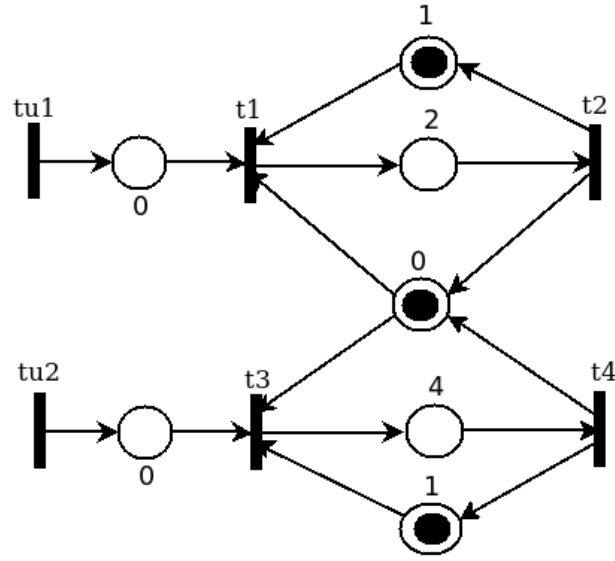


FIGURE 2.13: Exemple de RGETC

modèle *Max-plus* à commutation, qui est représenté par les équations suivantes :

$$\begin{cases} \theta_1(k+1) = 1 \cdot \theta_2(k) \oplus \theta_4(k) \oplus u(k+1) \\ \theta_2(k+1) = 3 \cdot \theta_2(k) \oplus 2 \cdot \theta_4(k) \oplus 2 \cdot u(k+1) \\ \theta_3(k+1) = \theta_3(k) \\ \theta_4(k+1) = \theta_4(k) \end{cases} \quad (2.27)$$

Sa forme matricielle est donnée par :

$$\theta(k+1) = \begin{bmatrix} \varepsilon & 1 & \varepsilon & e \\ \varepsilon & 3 & \varepsilon & 2 \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} e & \varepsilon \\ 2 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} \cdot u(k+1) \quad (2.28)$$

–*MODE 2* : L'état  $Q_2$  de l'automate, qui est généré par la configuration  $\mathcal{C}_{21}$  (voir la Figure 2.14), correspond à une situation où le jeton de conflit est consommé par le  $GET_1$  lors du  $(\mathbf{k})^{\text{ème}}$  franchissement, puis par le  $GET_1$  lors du  $(\mathbf{k}+1)^{\text{ème}}$  franchissement. Cela est dû au modèle *Max-plus* à commutation, qui est représenté par les équations suivantes :

$$\begin{cases} \theta_1(k+1) = 1 \cdot \theta_2(k) \oplus u(k+1) \\ \theta_2(k+1) = 3 \cdot \theta_2(k) \oplus 2 \cdot u(k+1) \\ \theta_3(k+1) = \theta_3(k) \\ \theta_4(k+1) = \theta_4(k) \end{cases} \quad (2.29)$$

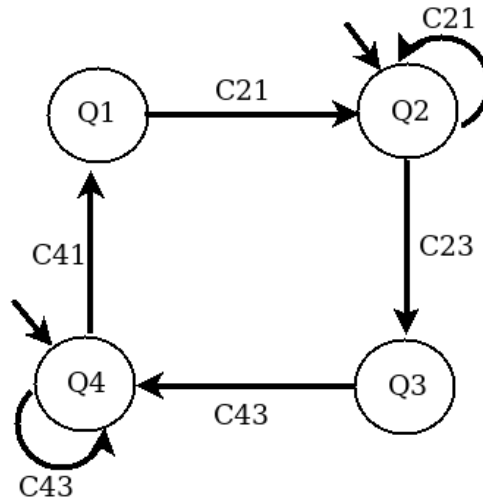


FIGURE 2.14: Représentations du système à commutation comme Automate

Sa forme matricielle est donnée par :

$$\theta(k+1) = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & 3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} e & \varepsilon \\ 2 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} \cdot u(k+1) \quad (2.30)$$

– Les modes 3 et 4 ont été successivement générés par les configurations  $C_{23}$  et  $C_{43}$ , respectivement. Ces configurations sont caractérisées par les équations d'état suivantes :

$$\theta(k+1) = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & 1 \\ \varepsilon & 4 & \varepsilon & 5 \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & 4 \end{bmatrix} \cdot u(k+1) \quad (2.31)$$

$$\theta(k+1) = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & 5 \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & 4 \end{bmatrix} \cdot u(k+1) \quad (2.32)$$

## 2.4 Comparaison des trois méthodes présentées

- La méthode de Addad [45] est une méthode qui se base sur l'allocation périodique des jetons de conflit, les auteurs ont proposé une approche algébrique Max-plus pour la modélisation de réseaux de graphes d'événements temporisés avec conflits. Des formes semblables à la représentation d'état des GET sont données indépendamment de la politique d'arbitrage de conflits appliquée. Pour une politique d'arbitrage périodique, la forme standard a été obtenue systématiquement sans passer par un GET équivalent.
- La méthode de Moradi [42] est une modélisation d'une classe de réseaux de Petri temporisés avec des problèmes de partage de ressources dans le dioïde Min-plus. En particulier, plusieurs sous-systèmes, décrits par des graphes d'événements temporisé (GET), se partagent  $n$  ressources modélisées par des jetons dans une place commune. Les auteurs ont adopté une politique de priorité préétablie et ont utilisé une modélisation basée sur l'algèbre Min-plus pour représenter le système résultant. Cette modélisation se concentre sur les sous-systèmes (GET).
- La méthode que nous avons proposée dans [21, 22] est une méthodologie pour la modélisation des RGETCs par des systèmes Max-plus linéaires à commutation. Nous avons présenté une modélisation des RGETCs par des systèmes linéaires Max-plus à commutation en utilisant la notion de configuration et le dateur généralisé. La modélisation est basée sur l'introduction de la notion de commutateurs. Plus précisément, pour chaque place en conflit, on associe un ensemble de commutateurs. La connaissance de la position des switchers est basée sur le routage des jetons des places en conflit. Pour chaque position de switchers, un modèle linéaire Max-plus est obtenu. La modélisation proposée dans notre travail représente le système global par rapport aux deux autres modèles qui représentent les sous systèmes (GETs) et des fonctions intermédiaires déduisent soit par rapport la politique de partage ou sur une politique de priorisation.

## 2.5 Conclusion

Dans ce chapitre, nous nous sommes intéressé à une classe de RdP temporisés avec conflit (RGETC) et quelques notions sur le partage des ressources. Ensuite, nous avons présenté deux méthodes qui ont été proposées par Addad et al [40] et Moradi et al [42] pour la modélisation du partage de ressources dans les RdPs en utilisant la théorie des dioïdes. Chaque méthode est illus-

*trée par un exemple. Dans la seconde partie de ce chapitre, nous avons donné quelques définitions qui nous ont permis de définir le modèle dynamique des RGETCs. Finalement, une comparaison a été effectuée entre les trois méthodes de modélisation de RGETC afin de mettre en évidence les distinctions entre ces approches.*

## Chapitre 3

# Commande sous contraintes temporelles des RGETCs

### 3.1 Introduction

*Dans ce chapitre, nous nous intéressons à la commande des systèmes à événements discrets temporisés avec des spécifications temporelles. Plus précisément, nous abordons le problème des contraintes de temps qui s'appliquent à certaines places ou, de manière plus générale, à certains chemins. Notre objectif est de développer une approche pour la synthèse de lois de commande qui fonctionnent en boucle fermée avec le système considéré, garantissant ainsi le respect des spécifications temporelles imposées sur le système, modélisé par des réseaux de graphes d'événements avec conflits. Lors de la synthèse de ces lois de commande pour les SEDs, il est crucial qu'elles répondent aux différentes exigences pouvant prendre diverses formes, telles que des contraintes d'état, des intervalles de temps, des durées de validité, et bien d'autres encore. La littérature propose plusieurs approches pour la commande des SEDs en prenant en compte différentes contraintes et spécifications.*

*De nombreux chercheurs se consacrent à la synthèse des lois de commande dans le but de satisfaire des exigences temporelles telles que des échéances, des intervalles de temps et des durées de validité. La contribution de l'article [47] repose sur l'utilisation d'automates temporisés pour modéliser à la fois le système et ces spécifications. Les auteurs ont étendu le cadre de la théorie de Ramadge et Wonham [48] et ont introduit le concept de contrôlabilité pour les SEDs temporisés. Ces approches sont intéressantes et pertinentes, cependant, le problème de l'explosion*

combinatoire limite leur applicabilité dans certains domaines. Pour surmonter cet inconvénient souvent rencontré lors de l'utilisation de modèles d'automates à états finis, plusieurs méthodes basées sur l'utilisation de graphes d'événements temporisés et l'algèbre des dioïdes sont proposées dans des travaux [12, 14, 15, 19]. Dans [12], les auteurs se focalisent sur la conception de lois de commande par retour d'état afin d'assurer que les systèmes considérés évoluent en respectant les contraintes de temps imposées. Les travaux présentés dans [14] et [15] portent sur la synthèse de lois de commande en boucle fermée pour les graphes d'événements temporisés soumis à des contraintes de temps strictes. Dans ces approches, les contraintes temporelles sont exprimées à l'aide de l'algèbre Max-plus, par des inégalités linéaires sur le vecteur d'état.

## 3.2 Synthèse de lois de commande pour RGETC sous contraintes temporelles

Dans cette section, notre objectif est d'étendre les travaux [14, 15] en vue de la synthèse de lois de commande pour les RGETCs soumis à des contraintes temporelles. Différentes approches ont été développées pour la commande des GETs en tenant compte des contraintes temporelles. Ces méthodes reposent sur l'utilisation des GETs et de l'algèbre des dioïdes.

### 3.2.1 Contraintes temporelles

On considère une place  $p_i$ , soumise à une contrainte temporelle stricte, un intervalle de temps  $[\tau_i^{min}, \tau_i^{max}]$  est associé à cette place comme illustré à la Figure 3.1 .

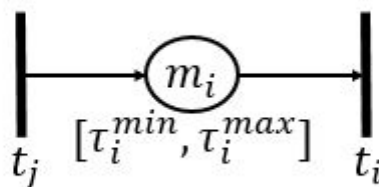


FIGURE 3.1: Contrainte temporelle

Cette contrainte se traduit, dans l'algèbre  $R_{max}$ , par l'inéquation suivante :

$$\tau_i^{\min} \cdot \theta_j(k - m_i) \leq \theta_i(t) \leq \tau_i^{\max} \cdot \theta_j(k - m_i) \quad (3.1)$$

L'inéquation de droite est satisfaite en tenant compte de la définition d'un graphe d'événements temporisé et d'un modèle Max-Plus linéaire associé à son comportement. La deuxième inéquation donnée par l'inégalité (3.2) :

$$\theta_i(k) \leq \tau_i^{\max} \cdot \theta_j(k - m_i) \quad (3.2)$$

dont le produit est défini dans  $R_{\max}$ , est une contrainte supplémentaire à satisfaire, la condition impose que le temps de séjour d'un jeton dans la place  $p_i$  ne doit pas dépasser  $\tau_i^{\max}$ . On souhaite calculer des lois de commande  $u(k)$  qui permettent de garantir le respect de cette contrainte du temps.

**Définition 3.1** Une transition est considérée comme observable lorsqu'elle est mesurable.

**Définition 3.2** RGETC est considéré comme complètement observable si toutes ses transitions sont observables. En revanche, s'il possède au moins une transition non observable, il est considéré comme partiellement observable.

### 3.3 Cas des RGETCs complètement observables

Dans cette partie, on suppose que les transitions des RGETCs considérés sont complètement observables, autrement dit, les événements associés aux transitions sont mesurables par des captures.

#### 3.3.1 Synthèse de commande pour les RGETCs à temps critique

Dans ce travail, notre objectif est de déterminer des lois de commande sous la forme (3.3) pour résoudre le problème des contraintes de temps (3.2) imposées aux RGETCs. Ces lois de commande en boucle fermée seront associées aux dateurs des transitions sources  $u^{\gamma(k)}(k+1)$  afin de prévenir la violation de ces contraintes temporelles.

$$u^{\gamma(k)}(k+1) = F^{\gamma(k)} \cdot \theta(k) \quad (3.3)$$

Les vecteurs  $F^{\gamma(k)}$  sont des gains de retour d'état à calculer pour satisfaire les contraintes temporelles (3.2).

L'expression du comportement dynamique d'un RGETC pour chaque mode est donnée :

$$\theta(k+1) = A^{\gamma(k)} \cdot \theta(k) \oplus B^{\gamma(k)} \cdot u^{\gamma(k)}(k+1) \quad (3.4)$$

La  $i^{\text{ème}}$  composante de l'équation (3.4) est donnée par :

$$\theta_i(k+1) = \bigoplus_{\delta=1}^{\gamma(k)} A_{i\delta}^N \cdot \theta_\delta(k) \oplus \bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{i\delta} \cdot u^{\gamma(k)}(k+1) \quad (3.5)$$

$Z$  est le nombre des transitions source (commande) connectées au RGETC.

Les contraintes temporelles imposées sur chaque GET, peut se traduire sous la forme de l'inéquation suivante :

$$\theta_i(k+1) \leq \tau_i^{\max} \cdot \theta_j(k+1) \quad (3.6)$$

Dans la suite de ce travail, nous considérons les hypothèses suivantes :

**Hypothèse 3.1** Toutes les transitions de RGETC sont commandables (il existe un chemin de transition source à chaque transition interne du GET).

**Hypothèse 3.2** Il existe dans chaque GET de ce RGETC au-moins un chemin vide  $\rho_i$  qui relie la transition source  $tu^{\gamma(k)}$  à la transition d'entrée  $t_j$  de la place sous contrainte de temps.

**Hypothèse 3.3** Chaque transition du RGETC est franchie dès qu'elle est franchissable (franchissement au plus tôt).

### 3.3.2 Une transition source unique dans chaque GET

#### 1. Une seule contrainte par GET

Dans cette partie, on considère un RGETC complètement observable avec une transition source pour chaque GET et soumis à une seule contrainte temporelle. Les hypothèses 3.1, 3.2 et 3.3 sont supposées vérifiées. La commande qui permet de satisfaire les contraintes temporelles du RGETC considéré est proposée dans le Théorème suivant.

**Théorème 3.3** [22] Nous considérons un RGETC complètement observable soumis à une contrainte temporelle dont le comportement dynamique est donné par les équations 3.4. la lois de commande

est donnée par l'équation :

$$u^{\gamma(k)}(k+1) = F^{\gamma(k)} \cdot \theta(k), \quad (3.7)$$

satisfait la contrainte de temps 3.6 si la condition suivante est vérifiée

$$\bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{i\delta} \leq \bigoplus_{\delta=1}^Z (\tau_i^{max} \cdot (B^{\gamma(k)})_{j\delta}), \quad (3.8)$$

$$\text{avec } F^{\gamma(k)} = \bigoplus_{\delta=1}^N ((A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{i\delta}^{\gamma(k)}).$$

**Démonstration 3.1** Du fait qu'il existe un chemin  $\rho$  qui relie la transition  $tu^{\gamma(k)}$  à la transition  $t_j$ , et à partir de la définition de la fonction dateur pour chaque transition  $t_j$ . La relation entre la transition de contrôle et la transition d'entrée est donnée par :

$$\theta_j(k+1) \geq \bigoplus_{\delta=1}^Z ((A^{\gamma(k)})^{m_\rho} \cdot B^l)_{j\delta} \cdot u^{\gamma(k)}(k+1 - m_\rho) \quad (3.9)$$

Dans ce travail, nous avons considéré que le marquage initial du chemin  $\rho$  est nul i.e.  $m_\rho = 0$ . L'équation (3.9) devient comme suit :

$$\theta_j(k+1) \geq \bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{j\delta} \cdot u^{\gamma(k)}(k+1) \quad (3.10)$$

Remplaçons l'équation (3.5) par son équation dans (3.6), on obtient alors :

$$\bigoplus_{\delta=1}^N (A_{i\delta}^{\gamma(k)}) \cdot \theta_\delta(k) \oplus \bigoplus_{\delta=1}^Z ((B^{\gamma(k)})_{i\delta}) \cdot u^{\gamma(k)}(k+1) \leq \tau_i^{max} \cdot \theta_j(k+1) \quad (3.11)$$

En prenant en considération **l'hypothèse 3.3** et en remplaçant l'équation (3.10) par son expression dans (3.11), on obtient :

$$\bigoplus_{\delta=1}^N (A_{i\delta}^{\gamma(k)}) \cdot \theta_\delta(k) \oplus \bigoplus_{\delta=1}^Z ((B^{\gamma(k)})_{i\delta}) \cdot u^{\gamma(k)}(k+1) \leq \bigoplus_{\delta=1}^Z (\tau_i^{max} \cdot B_{j\delta}^{\gamma(k)}) \cdot u^{\gamma(k)}(k+1) \quad (3.12)$$

Ainsi, la contrainte temporelle est satisfaite si les deux inégalités suivantes sont vérifiées :

$$\bigoplus_{\delta=1}^N (A_{i\delta}^{\gamma(k)}) \cdot \theta_\delta(k) \leq \bigoplus_{\delta=1}^Z (\tau_i^{max} \cdot (B^{\gamma(k)})_{j\delta}) \cdot u^{\gamma(k)}(k+1) \quad (3.13)$$

$$\bigoplus_{\delta=1}^Z ((B^{\gamma(k)})_{i\delta}) \cdot u^{\gamma(k)}(k+1) \leq \bigoplus_{\delta=1}^Z (\tau_i^{max} \cdot (B^{\gamma(k)})_{j\delta}) \cdot u^{\gamma(k)}(k+1) \quad (3.14)$$

Les deux conditions 3.13 et 3.14 peuvent s'écrire comme suite :

$$u^{\gamma(k)}(k+1) \geq \bigoplus_{\delta=1}^N \left[ (A^{\gamma(k)})_{i\delta} - \tau_i^{\max} - B_{j\delta}^{\gamma(k)} \right] \cdot \theta_\delta(k) \quad (3.15)$$

$$\bigoplus_{\delta=1}^Z ((B^{\gamma(k)})_{i\delta} \cdot u^{\gamma(k)}(k+1)) \leq \bigoplus_{\delta=1}^Z (\tau_i^{\max} \cdot (B^{\gamma(k)})_{j\delta}) \cdot u^{\gamma(k)}(k+1) \quad (3.16)$$

Si on prend en compte l'hypothèse 3.3, l'inéquation 3.15 représente la lois de commande et l'inégalité 3.16 correspond à la condition à vérifier.

□

## 2. Plusieurs contraintes dans chaque GET

On considère  $S$  places soumises à des contraintes de temps, notées  $p_s$ , pour  $s \in \{1, \dots, S\}$ . Pour chaque place  $p_s$ , on note respectivement par  $M_{p_s}$ ,  $\tau_s^{\max}$  le marquage initial et le délai maximal associé à cette place.

Par définition de la fonction dateur  $\theta_s(k)$ , les inégalités suivantes sont les contraintes temporelles imposées de RGETC pour tout  $s$  où,  $s \in \{1, \dots, S\}$ .

$$\theta_s(k+1) \leq \tau_s^{\max} \cdot \theta_s(k) \quad (3.17)$$

Nous supposons qu'il existe au moins un chemin vide, noté  $\alpha_s$ , qui relie la transition  $t_s$  et la transition de contrôle  $tu_i^{\gamma(k)}$  pour  $s = 1$  à  $S$ . Puisque toutes les transitions sont contrôlables, nous pouvons appliquer le résultat du Théorème 3.3 pour chaque configuration du réseau à chacune des contraintes temporelles. Si les conditions suffisantes du Théorème 3.3 sont vérifiées pour chaque contrainte  $s$ , alors la commande  $u_s^{\gamma(k)}$  calculée par le Théorème 3.3 satisfait la  $s^{\text{ème}}$  contrainte temporelle (3.17).

Les lois de commande  $u_s^{\gamma(k)}$  sont données par l'équation suivante :

$$u_s^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N ((A^{\gamma(k)})_{s\delta} - \tau_s^{\max} - B_{s'\delta}^{\gamma(k)}) \cdot \theta_\delta(k) \quad (3.18)$$

Le feed-back correspondant au maximum des lois de commande  $u_s^{\gamma(k)}(k+1)$ , pour  $s = 1, \dots, S$  garantit le respect des  $S$  contraintes temporelles.

**Théorème 3.4** [23] Soit un RGETC complètement observable avec une seule transition source pour chaque GET et dont le comportement dynamique est donné par les équations (3.4) et soumis à plusieurs contraintes de la forme (3.17). la lois de commande donnée par :

$$u^{\gamma(k)}(k+1) = \bigoplus_{s=1}^S \bigoplus_{\delta=1}^N ((A^{\gamma(k)})_{s\delta} - \tau_s^{max} - B_{s'\delta}^{\gamma(k)}) \cdot \theta_\delta(k)$$

assure le respect des contraintes temporelles (3.17) si les conditions suivantes sont vérifiées.

$$\bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{s\delta} \leq \bigoplus_{\delta=1}^Z (\tau_s^{max} \cdot (B^{\gamma(k)})_{s'\delta}). \quad (3.19)$$

**Démonstration 3.2** Selon l'énoncé du Théorème 3.3, ces conditions sont vérifiées pour chacune des contraintes (3.17), et en tenant compte du Théorème 3.3, on obtient l'égalité suivante

$$u_s^{\gamma(k)}(k+1) \leq \bigoplus_{\delta=1}^N ((A^{\gamma(k)})_{s\delta} - \tau_s^{max} - B_{s'\delta}^{\gamma(k)}) \cdot \theta_\delta(k)$$

la lois de commande qui répond à la  $s^{\text{ème}}$  contraintes est déterminée en prenant le maximum des lois de commande  $u_s^{\gamma(k)}(k+1)$ , et elle est donnée par l'inégalité suivante :

$$u_s^{\gamma(k)}(k+1) \leq \bigoplus_{s=1}^S u_s^{\gamma(k)}(k+1) \quad (3.20)$$

C'est claire que  $u_s^{\gamma(k)}(k+1) = \bigoplus_{s=1}^S u_s^{\gamma(k)}(k+1)$  satisfait les  $S$  contraintes temporelles (3.17).

□

### 3.3.3 Plusieurs transitions sources pour chaque GET

Dans cette partie, nous considérons des RGETCs avec plusieurs transitions sources dans chaque GET. Le comportement dynamique de ce graphe est présenté par l'équation linéaire (3.4).

#### 1. Une contrainte dans chaque GET

Nous considérons qu'une seule contrainte au maximum est imposée dans chaque GET du RGETC. Elle peut être exprimée par :

$$\theta_i(k+1) \leq \tau_i^{max} \cdot \theta_j(k+1) \quad (3.21)$$

Le plus court chemin qui relie l'une des transitions sources  $t_{u_i}^{\gamma(k)}$  à la transition d'entrée  $t_j$  de la place sous contrainte qui vérifie l'inégalité suivante :

$$\theta_j(k+1) \geq \bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{j\delta} \cdot u_c^{\gamma(k)}(k+1) \quad (3.22)$$

**Théorème 3.5** [23] Soit un REGTC complètement observable avec plusieurs transitions sources dans chaque GET et dont le comportement dynamique est donné par les équations 3.4. Le REGTC est soumis à une contrainte de la forme 3.21. Les lois de commande :

$$u_c^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N ((A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)}) \theta_\delta(k), \quad (3.23)$$

garantissent la validité de la contrainte 3.21 si les conditions suivantes sont vérifiées :

$$\bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{i\delta} \leq \bigoplus_{\delta=1}^Z (\tau_i^{max} \cdot (B^{\gamma(k)})_{c\delta}). \quad (3.24)$$

**Démonstration 3.3** La procédure de calcul de la loi de commande  $u_c^{\gamma(k)}$  est similaire à celle donnée par la preuve du théorème 3.5. Ainsi, la loi de commande  $u_c^{\gamma(k)}$  obtenue satisfaisant la contrainte (3.21) si les conditions 3.24 sont vérifiées.

□

## 2. Plusieurs contraintes dans chaque GET

Maintenant, nous généralisons le Théorème 3.5 en considérant plusieurs contraintes et plusieurs transitions sources.

$$\theta_s(k+1) \leq \tau_s^{max} \cdot \theta_{s'}(k) \quad (3.25)$$

**Théorème 3.6** [23] Soit un RGETC complètement observable avec plusieurs transitions sources dans chaque GET avec le comportement dynamique donné par les équations 3.4 et soumis à plusieurs contraintes de la forme (3.25). La commande donnée par :

$$u^{\gamma(k)}(k) = \bigoplus_{s=1}^S u_{sc}(k), \quad (3.26)$$

garantit le respect des contraintes (3.25), si les conditions suivantes sont vérifiées :

$$\bigoplus_{\delta=1}^Z (B^{\gamma(k)})_{s\delta} \leq \bigoplus_{\delta=1}^Z (\tau_s^{max} \cdot (B^{\gamma(k)})_{s'\delta}). \quad (3.27)$$

**Démonstration 3.4** La commande  $u^{\gamma(k)}$  est donnée par l'équation (3.23), si nous avons une seule contrainte. Nous aurons donc  $u_{sc}^{\gamma(k)}$  commandes, une commande pour chaque contrainte. Il est clair que la commande donnée par les équations (3.27) satisfait toutes les contraintes (3.25). □

**Remarque 3.7** Lorsqu'un GET est désactivé, il faut toujours garantir que le chemin entre la transition source et la transition en entrée de la place sous contrainte soit vide (Hypothèse 3.2). Pour garantir cette Hypothèse nous introduisons une place vide en entrée de transition source de ce dernier.

### 3.4 Exemple d'application : Gestion d'un croisement de lignes ferroviaires

Dans cette section, nous abordons un cas d'étude réaliste qui consiste en un système de croisement des rails ferroviaires. Ce cas d'étude est extrait de nos publications dans ([21] et [22]). Il s'agit également du même exemple traité pour la première fois [49] en utilisant une approche basé sur l'évaluation de performances des SEDs. La Figure 3.2 illustre la représentation graphique de ce système. Le réseau de transport comprend trois lignes unidirectionnelles : L1 (a à b), L2 (c d e) et L3 (f f' g). Il y a neuf feux de signalisation associés à ces lignes : a, á, b, c, d, e, f, f' et g. Chaque feu de signalisation a une durée spécifique : a (2 unités de temps), á (1 unité de temps), b (2 unités de temps), c (2 unités de temps), d (1 unité de temps), e (1 unité de temps), f (1 unité de temps), f' (1 unité de temps) et g (2 unités de temps) . Les feux de signalisation permettent de contrôler le passage des trains et de leur permettre de parcourir les tronçons jusqu'au feu suivant. De plus, les feux de signalisation sont utilisés pour garantir la présence d'un seul train sur les tronçons des lignes. Plus précisément, le feu a (ou f) autorise le passage d'un train à condition qu'aucun autre train ne soit présent sur les tronçons [aá] et [áb] (ou [ff'] et [fg]). Sur la ligne L2, seul un train est autorisé par tronçon. La gestion du croisement des trains est représentée par le graphe d'événements temporisés avec conflit, comme illustré dans la Figure 3 (Figure 3.3).

#### 3.4.1 Modèles linéaires Max-Plus à commutations du réseau ferroviaire

En se référant à la Figure 3.3, on peut observer la présence de deux places de conflit,  $\tilde{p}_1$  et  $\tilde{p}_2$ . Cela entraîne un total de 16 configurations de fonctionnement possibles. Dans le contexte de cette

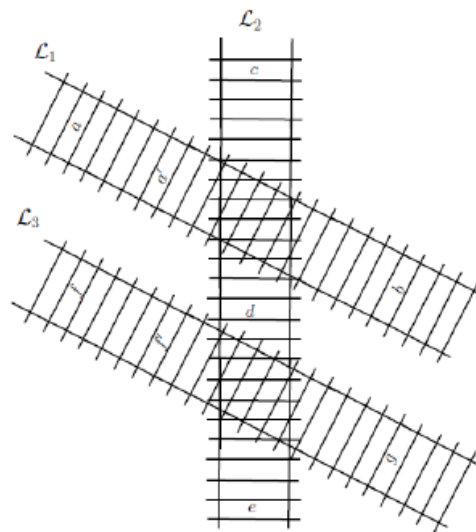


FIGURE 3.2: Croisement de trois lignes ferroviaires

*application, nous considérons uniquement deux modes. Dans le premier mode de configuration du réseau (mode 1), les jetons situés dans les places de conflit  $\tilde{p}_1$  et  $\tilde{p}_2$  sont consommés successivement par l'activation de GET1 et l'activation de GET3. Dans la deuxième configuration (mode 2), les jetons des places de conflit  $\tilde{p}_1$  et  $\tilde{p}_2$  sont consommés par l'activation de GET2.*

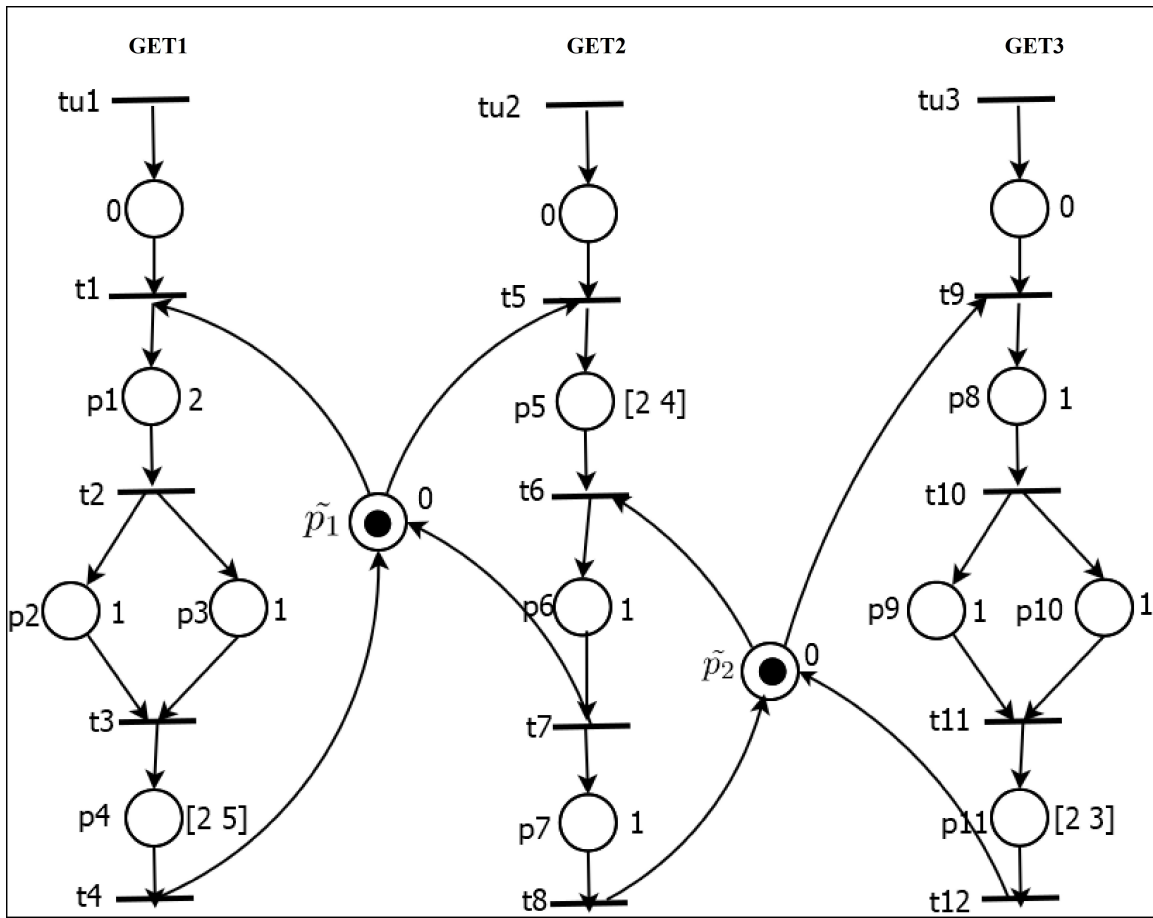


FIGURE 3.3: RGETC de croisement de trois lignes ferroviaires

### Modélisation du RGETC selon les configurations 1 et 2

En appliquant le théorème 2.10 , nous obtenons le système d'équations pour la configuration 1 suivant :

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

Le modèle peut s'écrire sous la forme d'état suivant :

$$x(k+1) = A^1 \cdot x(k) \oplus B^1 \cdot u^1(k) \quad (3.28)$$

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

Une approche similaire est adoptée pour la configuration 2, ce qui donne les équations Max-plus suivantes :

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

Le modèle peut s'écrire sous la forme d'état suivant :

$$x(k+1) = A^2 \cdot x(k) \oplus B^2 \cdot u^2(k) \quad (3.29)$$

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

### 3.4.2 Commande par retour d'état du réseau ferroviaire à temps critique

Cette section présente la synthèse d'une loi de commande visant à garantir que les contraintes temporelles imposées aux places  $p_2$ ,  $p_5$  et  $p_9$  pour les GETs du RGETC ( $GET_1$ ,  $GET_2$  et  $GET_3$ ) représenté dans la Figure 3.3 sont satisfaites. L'objectif principal est de calculer une loi de commande  $u^{\gamma(k)}(k+1)$  pour chaque mode, de manière à satisfaire les contraintes spécifiées. Étant donné que le système comporte trois transitions sources (chaque GET possède une transition source), nous appliquons le Théorème 3.3 pour garantir le respect des contraintes dans le système de croisement ferroviaire.

Les places  $p_4$ ,  $p_5$  et  $p_{11}$  des  $GET_1$ ,  $GET_2$  et  $GET_3$  sont soumises à des contraintes temporelles. Les contraintes à respecter sont exprimées successivement par les inégalités suivantes :

$$x_4(k) \leq 5 \otimes x_3(k) \quad (3.30)$$

$$x_6(k) \leq 4 \otimes x_5(k) \quad (3.31)$$

$$x_{12}(k) \leq 3 \otimes x_{11}(k) \quad (3.32)$$

### Une commande pour la configuration 1

Pour déterminer les lois de commande qui satisfont simultanément les contraintes imposées aux places  $p_4, p_5$  et  $p_{11}$ , nous vérifions les conditions et les hypothèses de ces contraintes.

Étant donné qu'il existe un chemin vide  $\rho_1$  reliant la transition  $tu_1$  à la transition  $t_3$ , nous aurons :

$$B_{3\delta}^1 = \begin{bmatrix} 3 & \varepsilon & \varepsilon \end{bmatrix}, B_{4\delta}^1 = \begin{bmatrix} 5 & \varepsilon & \varepsilon \end{bmatrix}, A_{4\delta}^1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

Les conditions,

$$\bigoplus_{\delta=1}^N B_{4\delta}^1 \leq \bigoplus_{\delta=1}^N \tau_{p_4}^{max} \otimes B_{3\delta}^1 \quad (3.33)$$

de **Théorème 3.3** sont vérifiées. Il existe alors une loi de commande qui satisfait la contrainte imposée à  $GET_1$ . Cette lois est donnée par l'équation suivante :

$$u_1^1(k+1) = \bigoplus_{\delta=1}^N (A_{4\delta}^1 - \tau_{p_4}^{max} - B_{3\delta}^1) \otimes x_\delta(k) \quad (3.34)$$

En remplaçant  $B_{3\delta}^1$  et  $\tau_{p_4}^{max}$  par leurs expressions, on obtient :

$$u_1^1(k+1) = \bigoplus_{\delta=1}^N (A_{4\delta}^1 - 5 - 3) \otimes x_\delta(k) \quad (3.35)$$

Comme l'expression  $(A_{4\delta}^1 - 5 - 3) \leq 0$ , nous pouvons la remplacer par l'élément neutre  $e$ . Ainsi, la lois de commande est défini comme suit :

$$u_1^1(k+1) = e \otimes x_7(k) \quad (3.36)$$

En suivant la même méthode pour  $u_3^1(k+1)$ , nous trouvons une lois de commande pour satisfaire la contrainte de temps de  $GET_3$ . Cette loi de commande est donnée comme suit :

$$u_3^1(k+1) = e \otimes x_8(k) \quad (3.37)$$

### Une commande pour la configuration 2

Nous procédons de la même manière pour la lois de commande de la configuration 2.

– Il est à noter qu’il existe également un chemin vide  $\rho_2$  reliant la transition  $tu_2$  à la transition  $t_5$ . Par conséquent, nous obtenons :

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

Nous le vérifions :

$$\bigoplus_{\delta=1}^N B_{6\delta}^2 \leq \bigoplus_{\delta=1}^N \tau p_5^{max} \otimes B_{5\delta}^2 \quad (3.38)$$

La condition du Théorème 3.3 est satisfaite, ce qui signifie qu’il existe une loi de commande qui respecte la contrainte imposée sur la place  $p_5$ . Cette loi de commande est donnée par l’équation suivante :

$$u_2^2(k+1) = \bigoplus_{\delta=1}^N (A_{6\delta}^2 - \tau p_5^{max} - B_{5\delta}^2) \otimes x_\delta(k) \quad (3.39)$$

Ainsi, on peut exprimer cela de la manière suivante :

$$u_2^2(k+1) = \bigoplus_{\delta=1}^N (A_{6\delta}^2 - 4 - B_{5\delta}^2) \otimes x_\delta(k) \quad (3.40)$$

la lois de commande pour la configuration 2 est donc donnée par :

$$u_2^2(k+1) = e \otimes x_4(k) \bigoplus e \otimes x_{12}(k) \quad (3.41)$$

Pour résumer, les lois de commande pour contrôler le réseau de croisement ferroviaires est donnée par le vecteur suivant :

$$u^{\gamma(k)}(k+1) = \begin{bmatrix} u^1(k+1) \\ u^2(k+1) \end{bmatrix} = \begin{bmatrix} u_1^1(k+1) \\ u_3^1(k+1) \\ u_2^2(k+1) \end{bmatrix} = \begin{bmatrix} e \cdot x_7(k) \\ e \cdot x_8(k) \\ e \cdot x_4(k) \bigoplus e \cdot x_{12}(k) \end{bmatrix} \quad (3.42)$$

Les lois de commande sont représentées par des places de commande marquées, temporisées et connectées au modèle RGETC initial. Ces places de commande, illustrées par des cercles rouges dans la Figure 3.4, jouent un rôle essentiel dans le respect des contraintes temporelles tout au long du fonctionnement du système.

## Simulations du RGETC contrôlé

Dans cette sous-section, nous déterminons l’évolution des marquages atteignables des RGETCs pour montrer l’évolution du système après la synthèse des places de commande. Nous simulons également l’évolution du système en boucle fermée (Figures 3.5, 3.6 et 3.7), afin de vérifier le respect des contraintes temporelles et l’efficacité des lois de commande calculées.

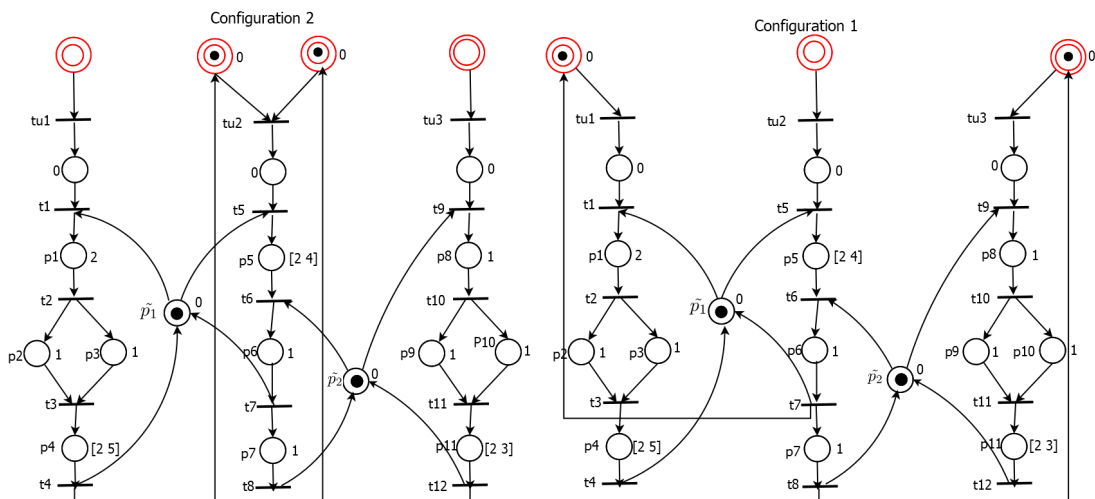


FIGURE 3.4: RGETCs commandés du croisement du réseau ferroviaire

– *Évolution des marquages pour la configuration 1 :*

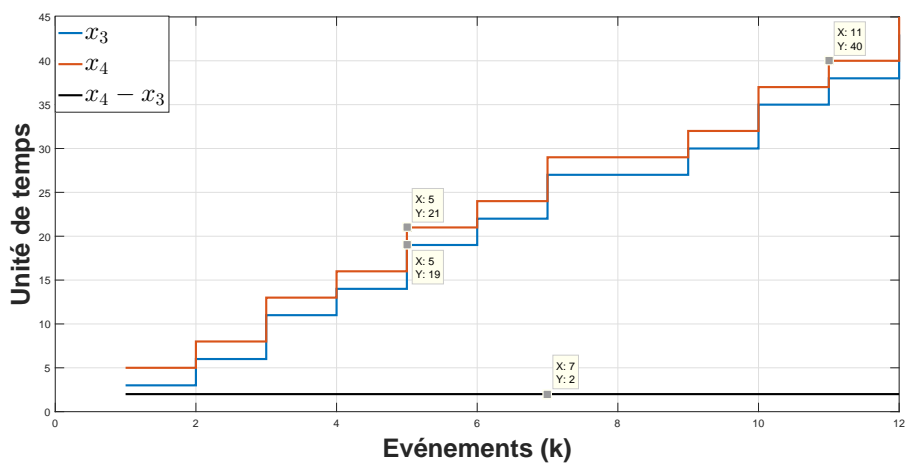
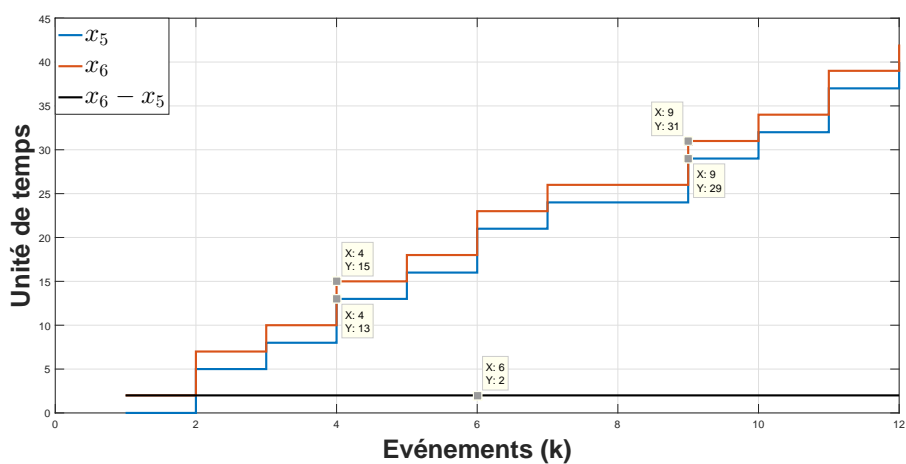
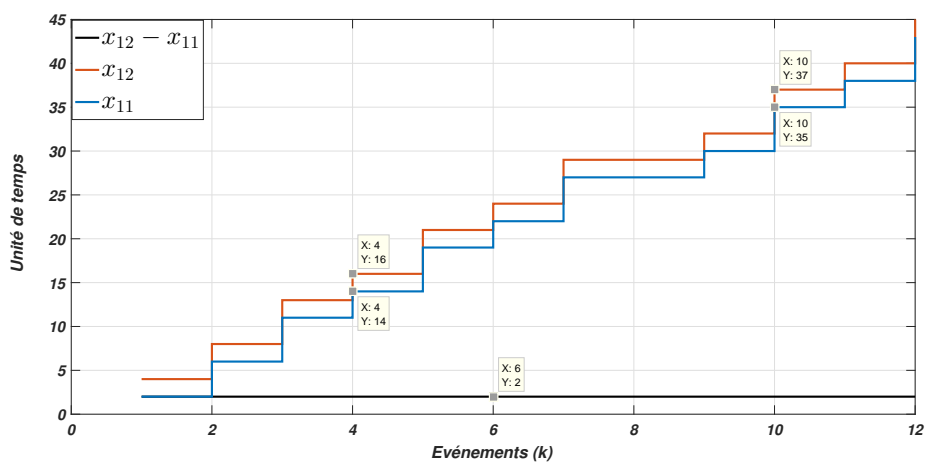
$$\begin{array}{c} \left[ \begin{array}{c} M \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \\ m_9 \\ m_{10} \\ m_{11} \\ m_{\bar{p}_1} \\ m_{\bar{p}_2} \end{array} \right] = \left[ \begin{array}{c} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \right] \xRightarrow{tu_1, t_1; tu_3, t_9/0; 0} \left[ \begin{array}{c} M_1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \xRightarrow{t_2; t_{10}/2; 1} \left[ \begin{array}{c} M_2 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right] \xRightarrow{t_3; t_{11}/1; 1} \left[ \begin{array}{c} M_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{array} \right] \xRightarrow{t_4; t_{12}/2; 2} \left[ \begin{array}{c} M_4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \right] \end{array}$$

– *Évolution des marquages pour la configuration 2 :*

$$\begin{array}{c} \left[ \begin{array}{c} M \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \\ m_9 \\ m_{10} \\ m_{11} \\ m_{\bar{p}_1} \\ m_{\bar{p}_2} \end{array} \right] = \left[ \begin{array}{c} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \right] \xRightarrow{tu_2, t_5/0} \left[ \begin{array}{c} M_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{array} \right] \xRightarrow{t_6/2} \left[ \begin{array}{c} M_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \xRightarrow{t_7/1} \left[ \begin{array}{c} M_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{array} \right] \xRightarrow{t_8/1} \left[ \begin{array}{c} M_4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{array} \right] \end{array}$$

– *Évolution des fonctions du dateur des transitions  $t_3, t_4, t_5, t_6, t_{11}$  et  $t_{12}$*

– *Interprétation des résultats de simulation*

FIGURE 3.5: Évolution des dateurs  $t_3$  et  $t_4$ FIGURE 3.6: Évolution des dateurs de  $t_5$  et  $t_6$ .FIGURE 3.7: Évolution des dateurs  $t_{11}$  et  $t_{12}$

– Les transitions  $t_4$  et  $t_{12}$  seront tirées après 2 unités de temps après les tirs des transitions  $t_3$  et  $t_{11}$ . Puisque les jetons des places de conflit  $\tilde{p}_1$  et  $\tilde{p}_2$  sont disponibles durant cette période, alors les contraintes soumises aux places  $p_4$  et  $p_{11}$  sont satisfaites. La transition  $t_6$  sera tirée après 2 unités de temps après le tir de  $t_5$ . Parce que le jeton de conflit de la place  $\tilde{p}_2$  est disponible durant cette période, alors la contrainte soumise à la place  $p_5$  est satisfaite.

Les Figures 3.5, 3.6 et 3.7 montrent l'évolution des dateurs des transitions d'entrée et de sortie des places sous contraintes de temps. Les transitions d'entrée et de sortie des places sous contraintes de temps, qui sont représentées par des lignes bleues et rouges. La ligne noire représente la différence entre deux événements successifs afin de montrer la satisfaction des contraintes temporelles imposées. Nous pouvons clairement voir sur la Les Figures 3.5, 3.6 et 3.7 que les différences temporelles entre l'occurrence des événements pour les dateurs des transitions ( $t_3$ ,  $t_4$ ,  $t_5$ ,  $t_6$ ,  $t_{11}$  et  $t_{12}$ ) et nous pouvons facilement montrer la satisfaction des contraintes de temps imposées après la synthèse des lois de commande et l'insertion de places de commandes.

### 3.5 RGETC partiellement observable

Dans cette section, nous considérons des RGETCs fortement connexes partiellement observables où les transitions ne sont pas toutes observables. Nous désignons par  $T_o$  l'ensemble des transitions observables et par  $T_{no}$  l'ensemble des transitions non observables, tel que  $T_o \cup T_{no} = T$  et  $T_o \cap T_{no} = \emptyset$ .

**Hypothèse 3.4** Dans cette partie, nous partons du principe que les RGETCs sont fortement connexes.

**Proposition 3.1** Pour toute transition non observable  $t_i \in T_{no}$ , il existe un chemin élémentaire  $\gamma$  reliant cette transition à une transition observable  $t_j \in T_o$  (Hypothèse 3.4). On peut exprimer le dateur associé à la transition  $t_i$  par l'expression suivante :

$$\theta_i(k+1) \leq \tau_o \cdot \theta_j(k+1 - m_o),$$

où  $\tau_o$  représente le délai du chemin  $\gamma_o$  et  $m_o$  représente le nombre des jetons dans ce chemin.

Notre but est de déterminer des lois de commande qui dépendent exclusivement des transitions observables.

### 3.5.1 Une transition source dans chaque GET

#### 1. Une contrainte dans chaque GET

Nous considérons dans cette partie un RGETC partiellement observable qui possède dans chaque GET une transition source et une contrainte temporelle. L'objectif est de trouver une loi de commande qui dépend uniquement des transitions observables et qui satisfait les contraintes temporelles imposées. En utilisant les résultats de la section précédente 3.3 et la propriété 3.1, nous obtenons le résultat suivant :

**Théorème 3.8** [23] *Étant donné un RGETC partiellement observable fortement connexe avec une transition source et une contrainte de temps dans chaque GET. Le comportement dynamique du RGETC considéré est donné par l'équation 3.4 et est soumis à une contrainte de la forme 3.6. la lois de commande est donnée par*

$$u^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N \left[ \bigoplus_{\delta \in T_o} \tau_o \cdot (A_{i\delta}^{\gamma(k)} - \tau_i^{max} - B_{j\delta}^{\gamma(k)}) \cdot \theta(k - m_o) \right] \\ \oplus \left[ \bigoplus_{\delta \in T_o} (A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)} \right] \cdot \theta(k)$$

où  $\tau_o$  et  $m_o$  représente successivement le délai et le marquage du chemin  $\gamma_o$  reliant la transition observable  $t_j$  à la transition non observable  $t_i$ .

la lois de commande satisfait à la contrainte 3.6 si la condition 3.8 est vérifiée.

**Démonstration 3.5** *Prenons le résultat du Théorème 3.3, si la condition 3.8 est satisfaite la lois de commande donnée par*

$$u^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N (A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)} \cdot \theta(k) \quad (3.43)$$

Cette loi de commande est obtenue pour toutes les transitions, quelles soient observables ou non, qui ne sont pas valides. Pour cela, nous développons l'équation 3.43 pour obtenir une loi de commande en termes de transitions observables. La lois de commande 3.43 peut être réécrite de la manière suivante.

$$u^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N \left[ \bigoplus_{\delta \in T_{no}} (A_{i\delta}^{v(k)} - \tau_i^{max} - B_{j\delta}^{\gamma(k)}) \cdot \theta(k) \right] \\ \oplus \left[ \bigoplus_{\delta \in T_o} (A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)} \right] \cdot \theta(k) \quad (3.44)$$

où  $|T_o|$  représente le nombre de transitions observables et  $|T_{no}|$  le nombre de transitions non observables et  $N$  le nombre de transitions ( $N = |T_{no}| + |T_o|$ ).

En considérant la propriété 3.1 qui stipule que, dans tout RGETC fortement connexe, il existe un chemin " $\gamma_o$ " connectant la transition non observables à la transition observable.

L'équation (3.44) peut être décrite par ce qui suit :

$$u^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N \left[ \bigoplus_{\delta \in T_o} (\tau_o \cdot ((A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)}) \cdot \theta(k - m_o)) \right] \oplus \left[ \bigoplus_{\delta \in T_o} (A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)} \cdot \theta(k) \right] \quad (3.45)$$

□

## 2. Plusieurs contraintes dans chaque GET

Nous considérons un RGETC partiellement observable soumis à  $S$  contraintes de temps exprimées par :

$$\theta_s(k+1) \leq \tau_s^{max} \cdot \theta_{s'}(k+1) \quad (3.46)$$

La procédure de synthèse de la lois de commande pour satisfaire les  $S$  contraintes de temps, est explicitement représentée par le résultat suivant.

**Théorème 3.9** [23] Etant donné un RGETC partiellement observable fortement connexe possédant une transition source unique pour chaque GET et soumis à plusieurs contraintes de la forme 3.46. Le comportement dynamique du RGETC considéré est exprimé par l'équation 3.4. la lois de commande donnée par

$$u^{\gamma(k)}(k+1) = \bigoplus_{s=1}^S \bigoplus_{\delta=1}^N \left[ \bigoplus_{\delta \in T_o} (\tau_o) \cdot (A_{s\delta}^{\gamma(k)} - \tau_s^{max} - B_{s'\delta}^{\gamma(k)}) \cdot \theta(k - m_o) \right] \oplus \left[ \bigoplus_{\delta \in T_o} (A^{\gamma(k)})_{s\delta} - \tau_s^{max} - B_{s'\delta}^{\gamma(k)} \cdot \theta(k) \right] \quad (3.47)$$

satisfait les spécifications temporelles 3.46 si les conditions 3.19 sont vérifiées.

**Démonstration 3.6** En considérant les résultats du Théorème précédent, il est facile de voir que la lois de commande définie dans ce théorème assure le respect de toutes les contraintes représentées par l'inégalité 3.46.

□

### 3.5.2 plusieurs transitions sources dans chaque GET

#### 1. Une contrainte dans chaque GET

Dans cette partie, nous considérons un RGETC partiellement observable avec plusieurs transitions sources et soumis à plus d'une contrainte dans chaque GET exprimée par 3.6, un chemin le plus court qui relie l'une des transitions sources  $t_{u_i}^{\gamma(k)}$  à la transition d'entrée de la contrainte  $t_j$ .

**Théorème 3.10** [23] *Étant donné un RGETC partiellement observable fortement connexe avec plusieurs transitions sources et soumis à des contraintes de la forme 3.17 pour chaque GET. Le comportement dynamique est donné par l'équation 3.4. La lois de commande*

$$u_c^{\gamma(k)}(k+1) = \bigoplus_{\delta=1}^N \left[ \bigoplus_{\delta \in T_o} (\tau_o) \cdot (A_{i\delta}^{\gamma(k)} - \tau_i^{max} - B_{j\delta}^{\gamma(k)}) \cdot \theta(k - m_o) \right] \\ \oplus \left[ \bigoplus_{\delta \in T_o} (A^{\gamma(k)})_{i\delta} - \tau_i^{max} - B_{j\delta}^{\gamma(k)} \cdot \theta(k) \right]$$

satisfait les spécifications temporelles 3.17 si les conditions 3.19 sont vérifiées.

**Démonstration 3.7** *On prend le résultat des Théorèmes 3.5 et 3.8. Il est facile de voir que la lois de commande donnée par le Théorème satisfait les spécifications temporelles (3.17) si les conditions (3.19) sont vérifiées.*

□

#### 2. Plusieurs contraintes dans chaque GET

Dans cette section, nous avons généralisé les résultats au cas des RGETC partiellement observable avec plusieurs transitions sources et plusieurs contraintes données par l'inégalité 3.48 dans chaque GET.

$$\theta_s(k+1) \leq \tau_s^{max} \cdot \theta_{s'}(k+1) \quad (3.48)$$

**Théorème 3.11** [23] *Étant donné un RGETC partiellement observable fortement connexe avec plusieurs transitions sources et soumis à plusieurs contraintes de la forme 3.48 pour chaque GET,*

dont le comportement dynamique est donné par l'équation 3.4 . La lois de commande donnée par :

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

satisfait les spécifications temporelles 3.48, où  $u^{\gamma(k)}(k+1)$  est la lois de commande se calcule par le Théorème 3.9 en supposant que la condition est satisfaite pour les  $S$  contraintes .

**Démonstration 3.8** Selon les résultats des Théorèmes 3.9 et 3.10 , la lois de commande définie par le Théorème 3.11 assure le respect de toutes les contraintes représentées par l'inégalité (3.46).

□

## 3.6 Exemple d'illustration

L'exemple ci-dessous est issu de notre publication dans ([23]). Nous considérons le réseau de graphe d'événements temporisés avec conflit de la Figure 3.8. Ce graphe modélise un système de fabrication (Figure 3.8). Il contient 3 transitions sources  $tu_1, tu_2$  et  $tu_3$ . Cet ensemble contient 3 lignes de productions et 2 ressources partagées qui interconnectent et alimentent les machines 2 et 3. Chaque ligne contient une zone de production modélisée respectivement par  $p_{12}, p_{45}$  et  $p_{78}$ . Le temps de traitement dans les zones est fixé à  $\tau_{12} = 2, \tau_{45} = 1$  et  $\tau_{78} = 2$  unité de temps. Les zones de production sont suivies des zones de vérification  $p_{23}, p_{53}$  et  $p_{89}$  sont fixées  $\tau_{23} = 1, \tau_{53} = 1$  et  $\tau_{89} = 1$  unité de temps. En terminant par la zone d'assemblage  $p_{35}$  qui se fixe à  $\tau_{35} = 1$  unité de temps.

### 3.6.1 Premier cas : toutes les transitions sont considérées observables

Dans ce cas, nous considérons que toutes les transitions de RGETC sont observables. Ce système de fabrication est soumis à des contraintes de temps imposées sur trois chemins.

- $\mu_1$  relie les transitions  $t_2$  à  $t_6$ .
- $\mu_2$  relie les transitions  $t_4$  à  $t_6$ .
- $\mu_3$  relie les transitions  $t_8$  à  $t_{10}$

**Remarque 3.12** Nous attribuons à chaque transition  $t_i$  de RGETC de l'exemple de la Figure 3.8 une fonction dateur  $z_i(k)$ .

Ces contraintes sont respectivement représentées par les inégalités suivantes :

$$z_6(k+1) \leq 2 \cdot z_2(k+1) \tag{3.49}$$

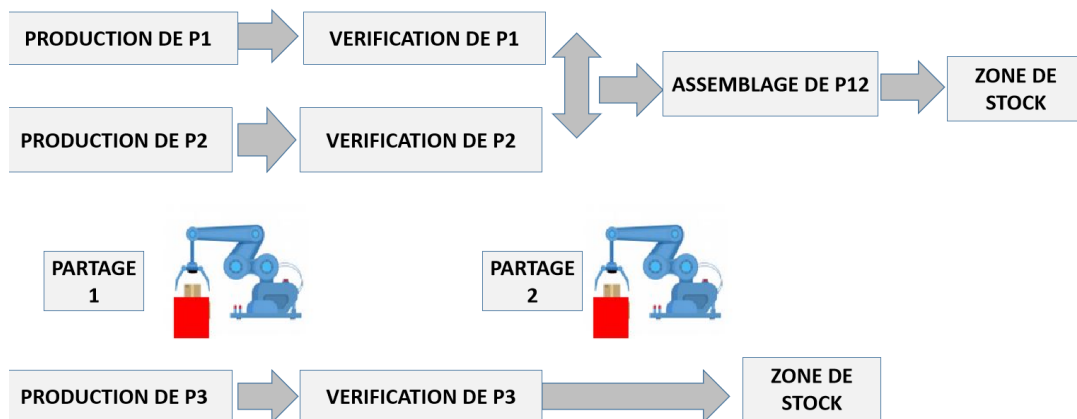


FIGURE 3.8: Zone de production

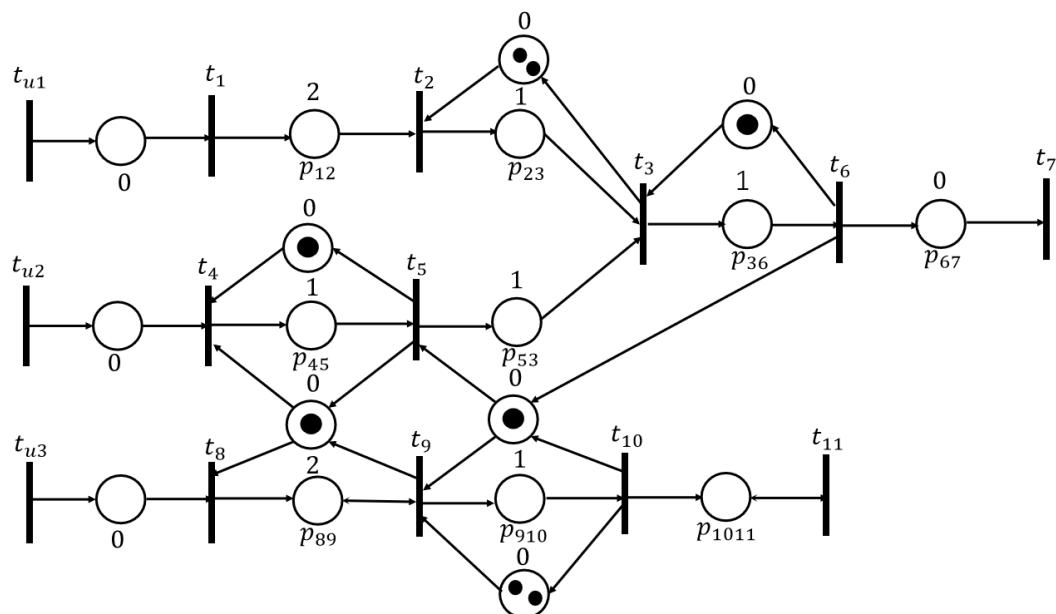


FIGURE 3.9: RGETC correspondant à la zone de production

$$z_6(k+1) \leq 3 \cdot z_4(k+1) \quad (3.50)$$

$$z_{10}(k+1) \leq 3 \cdot z_8(k+1) \quad (3.51)$$

L'objet de ce problème est de calculer la lois de commande en boucle fermée qui satisfait les trois contraintes en temps réels dans certaines situations du RGETC appliquées sur les chemins  $\mu_1, \mu_2$  et  $\mu_3$ . Dans notre cas, nous considérons deux situations de fonctionnement de RGETC  $C_1$  et  $C_2$ , où la configuration 1 :  $C_1 = (t_9, t_4; t_{10}, t_5)$  et la configuration 2 :  $C_2 = (t_9, t_8; t_6, t_9)$ .

Nous appliquons d'abord le Théorème 3.3 pour calculer les équations d'état des deux configurations, qui sont données successivement par le système d'équations suivant :

$$z(k+1) = A^{\gamma(k)} \cdot z(k) \oplus B^{\gamma(k)} \cdot u(k+1) \quad (3.52)$$

Où,

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

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

Dans la première configuration C1, les contraintes sont appliquées aux chemins  $\mu_1$  et  $\mu_2$ . En appliquant le Théorème 3.3, on calcule les lois de commande  $u_1(k+1)^1$  et  $u_2(k+1)^1$ . Nous obtenons les équations suivantes :

$$\begin{cases} u_1(k+1)^1 = \bigoplus_{\delta=1}^{12} ((A^1)_{6\delta} - 2 - B_{2\delta}^{\gamma(k)}) \cdot z(k) \\ u_1(k+1)^1 = e \cdot z_5(k) \oplus e \cdot z_6(k) \oplus e \cdot z_{12}(k) \oplus e \cdot z_9(k) \oplus e \cdot z_{10}(k) \end{cases} \quad (3.53)$$

$$\begin{cases} u_2(k+1)^1 = \bigoplus_{\delta=1}^{12} ((A^1)_{6\delta} - 3 - B_{4\delta}^{\gamma(k)}) \cdot z(k) \\ u_2(k+1)^1 = e \cdot z_5(k) \oplus e \cdot z_6(k) \oplus e \cdot z_{12}(k) \oplus e \cdot z_9(k) \oplus e \cdot z_{10}(k) \end{cases} \quad (3.54)$$

Puisque la date de tir de la transition  $t_6$  est supérieure aux dates de tir  $t_3, t_2$  et  $t_5$ , la lois de commande peut être réécrite sous une forme compacte comme suit :

$$u_1(k+1)^1 = u_2(k+1)^1 = e \cdot z_6(k) \oplus e \cdot z_{10}(k) \oplus e \cdot z_9(k) \quad (3.55)$$

Ensuite, nous suivons la même approche pour calculer les lois de commande  $u_3(k+1)^2$  de la configuration 2 :

$$\begin{cases} u_3(k+1)^2 = \bigoplus_{\delta=1}^{12} ((A^2)_{10\delta} - 3 - B_{8\delta}^{\gamma(k)}) \cdot z(k) \\ u_3(k+1)^2 = e \cdot z_7(k) \oplus e \cdot z_9(k) \oplus e \cdot z_{10}(k) \end{cases} \quad (3.56)$$

Les contraintes de temps imposées aux chemins  $\mu_1$  et  $\mu_2$  sont satisfaites après avoir appliqué la commande car la transition  $t_2$  ne se franchit qu'une seule fois et les jetons des places de conflit

sont consommés par les transitions  $t_4$  et  $t_5$ . Ce qui assure un temps de séjour de 4 unités de temps dans les chemins  $\mu_1$  et de 3 unités de temps. Il en est de même pour la deuxième configuration. La contrainte temps soumise au chemin  $\mu_3$  est satisfaite car les lois de commande, la transition  $t_4$  ne se franchit qu'une seule fois et les places de conflits sont consommées par les transitions  $t_4$  et  $t_5$  ce qui assure la satisfaction de la contrainte.

### 3.6.2 Deuxième cas : Les transitions ne sont pas toutes considérées comme observables.

Dans cette partie, l'ensemble des transitions observables est donné par  $T_o = \{t_2, t_5, t_9\}$  et l'ensemble des transitions non observables est  $T_{no} = \{t_1, t_3, t_4, t_6, t_7, t_8, t_{10}, t_{11}\}$ . Notre objectif est de calculer des lois de commande en boucle fermée qui ne dépendent que de l'ensemble des transitions observables. Les lois de commande pour la configuration 1 sont calculées en utilisant le Théorème 3.10 :

$$\begin{cases} u_1(k+1)^1 = \bigoplus_{\delta=1}^{12} \left( \bigoplus_{\delta \in T_o} (\tau_\sigma \cdot (((A^2)_{6\delta} - 2 - B_{2\delta}^{\gamma(k)}) \cdot z(k - m_\sigma)) \right) \\ \quad \bigoplus \left( \bigoplus_{\delta \in T_{no}} ((A^2)_{6\delta} - 2 - B_{2\delta}^{\gamma(k)}) \cdot z(k) \right) \\ u_1(k+1)^1 = 2 \cdot z_2(k) \oplus 2 \cdot z_5(k) \oplus 1 \cdot z_9(k) \end{cases} \quad (3.57)$$

$$u_2(k+1)^1 = 2 \cdot z_2(k) \oplus 2 \cdot z_5(k) \oplus 1 \cdot z_9(k) \quad (3.58)$$

En utilisant la même procédure que précédemment, nous obtenons la lois de commande pour la configuration  $C_2$ .

$$u_3(k+1)^2 = 2 \cdot z_2(k) \oplus 2 \cdot z_5(k) \oplus 1 \cdot z_9(k) \quad (3.59)$$

Ces lois de commande sont représentées par des places marqués connectés au graphe initial de RGETC (Figures 3.10 et 3.11), dans le but d'assurer la satisfaction des contraintes de temps imposées dans les chemins  $\mu_1, \mu_2, \mu_3$ .

### Évolution du marquage et simulations des RGETCs contrôlés

Dans cette partie, la satisfaction des contraintes temporelles des RGETCs considérés après la synthèse des commandes est vérifiée par l'évolution des marquages atteignables. Nous simulons également l'évolution du système en boucle fermée pour nous assurer que les contraintes de temps sont respectées et que les lois de commande calculées sont connecter (voir les Figures 3.12, 3.13 et 3.14).

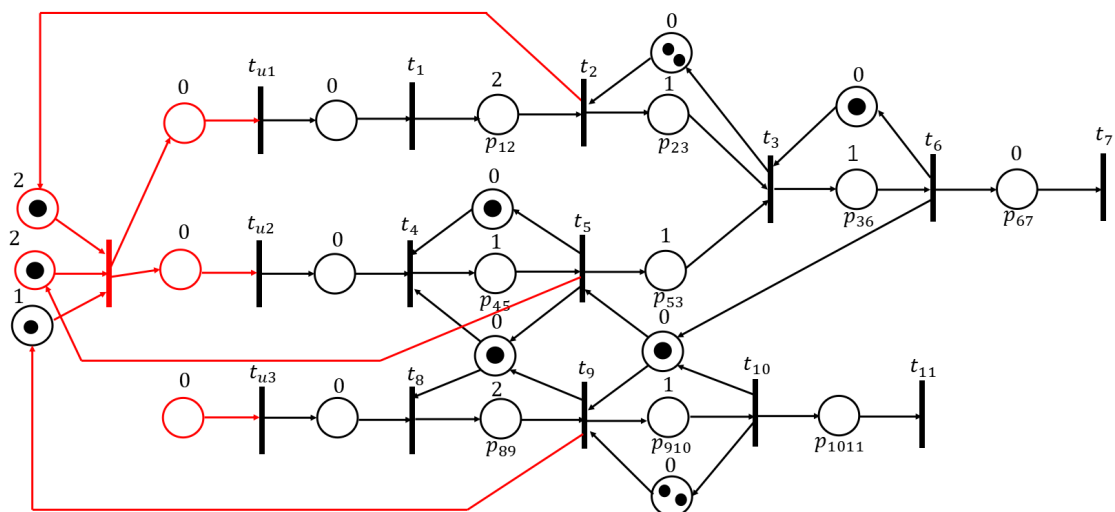


FIGURE 3.10: Loi de commande pour la configuration 1

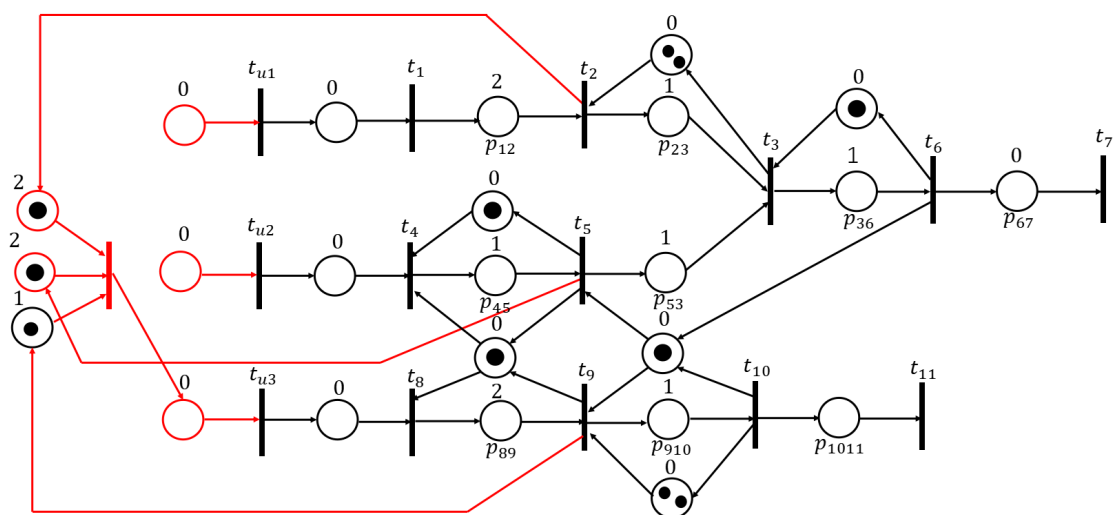


FIGURE 3.11: Loi de commande pour la configuration 2

– *Évolution du marquage pour la configuration 1 :*

$$\begin{array}{c}
 \left[ \begin{array}{c} M \\ m_{12} \\ m_{23} \\ m_{45} \\ m_{53} \\ m_{36} \\ m_{67} \\ m_{89} \\ m_{910} \\ m_{1112} \\ m_{32} \\ m_{54} \\ m_{63} \\ m_{109} \\ m_{p_{c1}} \\ m_{p_{c2}} \end{array} \right] = \left[ \begin{array}{c} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} \right] \xRightarrow{tu_1, t_1; tu_2, t_4/0; 0} \left[ \begin{array}{c} M_1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 1 \\ 2 \\ 0 \\ 1 \end{array} \right] \xRightarrow{1.t_2; t_5/2; 1} \left[ \begin{array}{c} M_2 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \\ 0 \end{array} \right] \xRightarrow{t_3/1} \left[ \begin{array}{c} M_3 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \\ 2 \\ 1 \\ 1 \end{array} \right]
 \end{array}$$

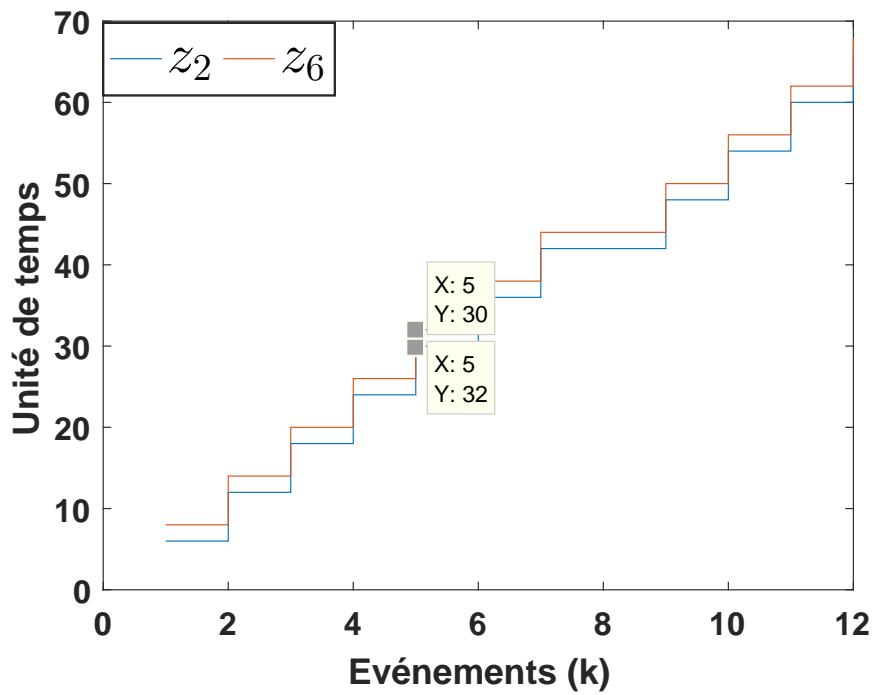
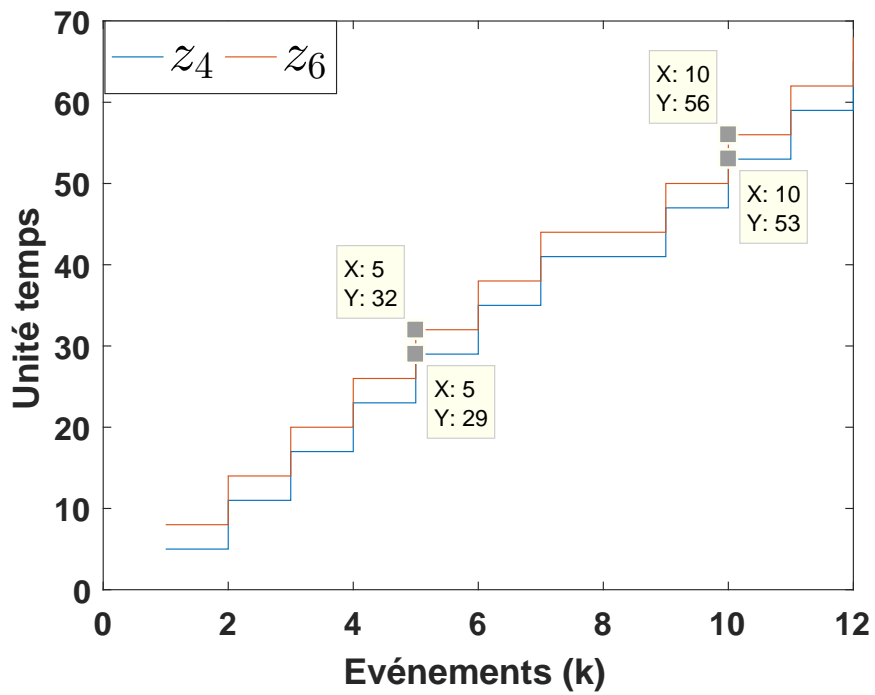
$$\xRightarrow{t_6/1} \left[ \begin{array}{c} M_4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} \right] \xRightarrow{t_7/0} \left[ \begin{array}{c} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} \right] \tag{3.60}$$

– *Évolution du marquage pour la configuration 2 :*

$$\begin{array}{c}
 \left[ \begin{array}{c} M \\ m_{12} \\ m_{23} \\ m_{45} \\ m_{53} \\ m_{36} \\ m_{67} \\ m_{89} \\ m_{910} \\ m_{1011} \\ m_{32} \\ m_{54} \\ m_{63} \\ m_{109} \\ m_{p_{c1}} \\ m_{p_{c2}} \end{array} \right] = \begin{array}{c} \left[ \begin{array}{c} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} \right] \xrightarrow{tu_3, t_8/0} \left[ \begin{array}{c} M_5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 0 \\ 1 \end{array} \right] \xrightarrow{1.t_9/2} \left[ \begin{array}{c} M_6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{array} \right] \xrightarrow{t_{10}/1} \left[ \begin{array}{c} M_7 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right] \xrightarrow{t_{11}/1} \left[ \begin{array}{c} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} \right]
 \end{array} \quad (3.61)$$

### Interprétation des résultats des simulations et de l'évolution des marquages

En appliquant les lois de commande  $u_1(k+1)^1$  et  $u_2(k+1)^1$  (Configuration 1), les contraintes temporelles imposées aux chemins  $\mu_1$  et  $\mu_2$  sont satisfaites car les transitions  $t_2$  et  $t_4$  ne sont déclenchées qu'une seule fois et les jetons des places de conflit sont nécessairement consommés par les transitions  $t_4$  et  $t_5$ . Cela garantit un temps de maintien de 2 et 3 unités de temps dans les chemins  $\mu_1$  et  $\mu_2$  respectivement (voir l'équation 3.60). Nous utilisons les mêmes arguments pour la deuxième configuration, la contrainte temporelle imposée au chemin  $\mu_3$  est satisfaite grâce à la lois de commande  $u_3(k+1)^2$  (Configuration 2) où, la transition  $t_8$  n'est déclenchée qu'une seule fois et les jetons des places de conflit sont nécessairement consommés par les transitions  $t_8$  et  $t_9$  ce qui assure la satisfaction de la contrainte (voir l'équation 3.61). Les Figures 3.12, 3.13 et 3.14 montre l'évolution des données des transitions d'entrée et de sortie des places sous contraintes temporelles après application des lois de commande, qui sont représentées respectivement par

FIGURE 3.12: Évolution des dateurs  $t_2$  et  $t_6$ FIGURE 3.13: Évolution des dateurs  $t_4$  et  $t_6$

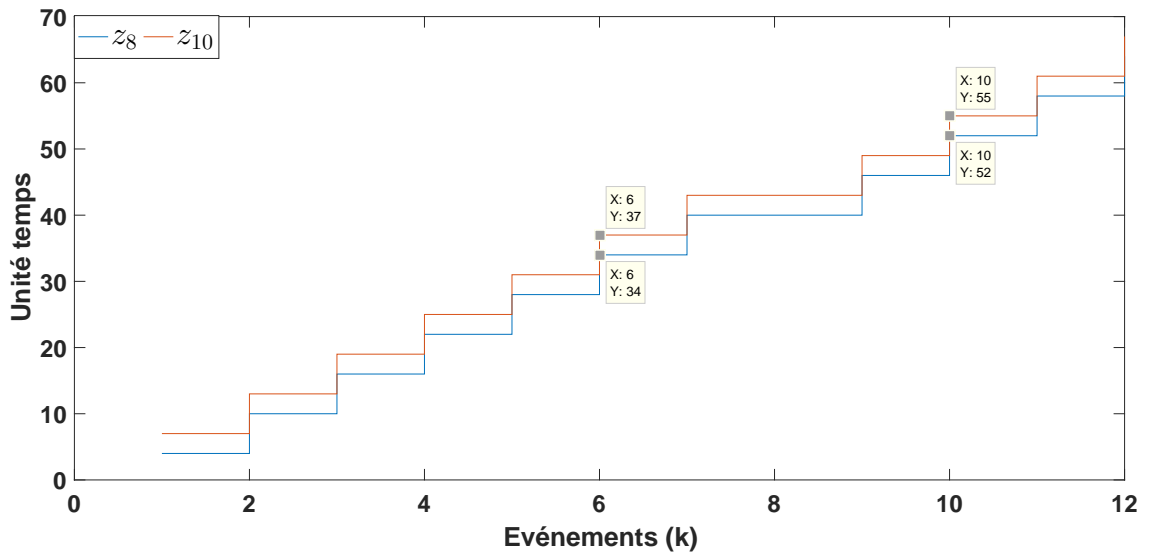


FIGURE 3.14: Évolution dateurs  $t_8$  et  $t_{10}$

des lignes bleues et rouges après avoir synthétisé les lois de commande. Nous pouvons facilement montrer que les contraintes de temps imposées sont satisfaites en observant la différence de temps entre les occurrences des événements.

### 3.7 Conclusion

Dans cette partie, nous avons proposé une approche de commande des systèmes à événements discrets, modélisés par des réseaux graphes d'événements temporisés avec conflit soumis à des contraintes de temps. Notre approche est basée sur l'utilisation d'une modélisation algébrique pour décrire le comportement des RGETCs basée sur des systèmes Max-plus linéaires à commutation. Il s'agit d'une extension du formalisme de l'algèbre Max-plus pour représenter la dynamique d'une classe de systèmes à événements discrets. Une approche formelle pour la conception de lois de commande est proposée afin de garantir des contraintes de temps strictes dans les RGETCs observable et partiellement observable en utilisant l'algèbre Max-plus. Des conditions suffisantes sont proposées pour l'existence des lois de commande qui assurent la satisfactions des contraintes de temps strictes imposées à certaines chemins des RGETC considérés. Les lois de commande calculées sont représentées par des places de commande connectées au RGETC initial qui permettent d'éviter la violation des contraintes de temps.



# Conclusion Générale

*Nous avons présenté dans cette thèse des méthodologies de synthèse de lois de commande pour les systèmes à événements discrets modélisés par des graphes d'événements temporisés avec conflits. Nous avons développé une approche de commande et de modélisation des réseaux des graphes d'événements temporisés avec conflits (RGETCs) soumis à des spécifications (contraintes) de temps imposées aux places ou bien aux chemins de ces graphes. Tout d'abord, nous avons établi un cadre de modélisation algébrique pour décrire le comportement des RGETCs en commutant des équations linéaires Max-plus. Il s'agit d'une extension du formalisme de l'algèbre Max-plus pour représenter la dynamique d'une classe de système à événements discrets avec des ressources partagées. Deuxièmement, nous avons généralisé les méthodes existantes de synthèse de commande pour garantir les contraintes temporelles pour les systèmes RGETCs. Une première approche est introduite pour calculer les lois de commande en boucle fermée afin de satisfaire des contraintes temporelles strictes des RGETCs avec des transitions toutes observables. Une extension pour les RGETCs partiellement observables est proposée et une procédure pour synthétiser les lois de commande est développée pour garantir les contraintes temporelles. Des conditions suffisantes sont proposées pour l'existence de lois de commande qui assurent le respect des contraintes de temps strictes imposées à certaines places d'un RGETC. Les lois de commande calculées sont représentées par des places de commandes connectés au RGETC initial qui permettent d'éviter la violation des contraintes de temps. Deux applications qui illustrent des résultats théoriques obtenus sont présentés. Il s'agit, d'un réseau de transport à temps critique et une zone de fabrication sont présentés pour montrer l'efficacité de la méthode développée. Une question importante pour les recherches futures, nous souhaitons généraliser notre approche aux RdPs avec conflits pas nécessairement les RGETCs. En ce qui concerne le problème de commande, il serait intéressant de développer des outils pour faciliter les calculs et le traitement de cas réels. Dans un futur proche, nous prévoyons de proposer un algorithme itératif qui détermine toutes les configurations des*

*RGETCs et calcule automatiquement les lois de commande.*

# Bibliographie

- [1] T. Moor, K. Schmidt, S. Perk, *Applied supervisory control for a flexible manufacturing system*, *IFAC Proceedings Volumes* 43 (12) (2010) 253–258. doi: 10.3182/20100830-3-de-4013.00043.
- [2] S. Lauzon, A. Ma, J. Mills, B. Benhabib, *Application of discrete-event-system theory to flexible manufacturing*, *IEEE Control Systems Magazine* 16 (1) (1996) 41–48. doi: 10.1109/37.482149.
- [3] K. C. Lee, S. Lee, *Performance evaluation of switched ethernet for real-time industrial communications*, *Computer Standards & Interfaces* 24 (5) (2002) 411–423. doi: 10.1016/s0920-5489(02)00070-3.
- [4] M. Dotoli, M. P. Fanti, A. M. Mangini, *Real time identification of discrete event systems using petri nets*, *Automatica* 44 (5) (2008) 1209–1219. doi: 10.1016/j.automatica.2007.10.014.
- [5] P. Spacek, M.-A. Manier, A. E. Moudni, *Control of an electroplating line in the max and min algebras*, *International Journal of Systems Science* 30 (7) (1999) 759–778. doi: 10.1080/002077299292065.
- [6] L. Houssin, S. Lahaye, J.-L. Boimond, *Just in time control of constrained  $(\max,+)$ -linear systems*, *Discrete Event Dynamic Systems* 17 (2) (2007) 159–178. doi: 10.1007/s10626-006-0009-5.
- [7] K. Outafraout, A. Nait-Sidi-Moh, E. H. C. E. Alaoui, *A control approach based on colored hybrid petri nets and  $(\max,+)$  algebra : Application to multimodal transportation systems*, *IEEE Transactions on Automation Science and Engineering* 17 (3) (2020) 1208–1220. doi: 10.1109/tase.2020.2973996.

- [8] A. Mifdaoui, F. Frances, C. Fraboul, *Real-time communication over switched ethernet for military applications*, in : *Proceedings of the 2005 ACM conference on Emerging network experiment and technology - CoNEXT05*, ACM Press, 2005. doi: 10.1145/1095921.1095973.
- [9] R. Cruz, *A calculus for network delay. i. network elements in isolation*, *IEEE Transactions on Information Theory* 37 (1) (1991) 114–131. doi: 10.1109/18.61109.
- [10] C. Kim, T.-E. Lee, *Feedback control of cluster tools for regulating wafer delays*, *IEEE Transactions on Automation Science and Engineering* 13 (2) (2016) 1189–1199. doi: 10.1109/tase.2015.2404921.
- [11] S. Balemi, G. Hoffmann, P. Gyugyi, H. Wong-Toi, G. Franklin, *Supervisory control of a rapid thermal multiprocessor*, *IEEE Transactions on Automatic Control* 38 (7) (1993) 1040–1059. doi: 10.1109/9.231459.
- [12] C. Maia, C. Andrade, L. Hardouin, *On the control of max-plus linear system subject to state restriction*, *Automatica* 47 (5) (2011) 988–992. doi: 10.1016/j.automatica.2011.01.047.
- [13] K. Tebani, S. Amari, R. Kara, *Closed-loop control of constrained discrete event systems : application to a networked automation system*, *The International Journal of Advanced Manufacturing Technology* 90 (5-8) (2017) 1295–1306. doi: 10.1007/s00170-016-9466-7.
- [14] A. M. Atto, C. Martinez, S. Amari, *Control of discrete event systems with respect to strict duration : Supervision of an industrial manufacturing plant*, *Computers & Industrial Engineering* 61 (4) (2011) 1149–1159. doi: 10.1016/j.cie.2011.07.004.
- [15] S. Amari, I. Demongodin, J. J. Loiseau, C. Martinez, *Max-plus control design for temporal constraints meeting in timed event graphs*, *IEEE Transactions on Automatic Control* 57 (2) (2012) 462–467. doi: 10.1109/tac.2011.2164735.
- [16] S. Amari, *Feedback control for a class of discrete event systems with critical time*, *International Journal of Control* 88 (10) (2015) 1974–1983. doi: 10.1080/00207179.2015.1025430.
- [17] K. Tebani, S. Amari, R. Kara, *Control of petri nets subject to strict temporal constraints using max-plus algebra*, *International Journal of Systems Science* 49 (6) (2018) 1332–1344. doi: 10.1080/00207721.2018.1445311.

- 
- [18] S. Amari, *Commande des graphes d'événements temporisés sous contraintes temporelles*, Ph.D. thesis, Nantes (2005).
- [19] L. Houssin, S. Lahaye, J.-L. Boimond, Control of  $(\max, +)$ -linear systems minimizing delays, *Discrete Event Dynamic Systems* 23 (3) (2012) 261–276. doi: 10.1007/s10626-012-0143-1.
- [20] K. Tebani, *Commande temps réel d'une classe de systèmes à événements discrets sous contraintes par l'algèbre des dioïdes*, Ph.D. thesis, Université Mouloud Mammeri (2017).
- [21] S. Aberkane, R. Kara, S. Amari, *Commande sous contraintes temporelles des réseaux de graphes d'événements temporisés en conflit*, in : MSR 2019-12ème Colloque sur la Modélisation des Systèmes Réactifs, Nov 2019, Angers, France, 2019.
- [22] S. Aberkane, R. Kara, S. Amari, Modelling and feedback control for a class of petri nets with shared resources subject to strict time constraints using max-plus algebra, *International Journal of Systems Science* 52 (14) (2021) 3060–3075. doi: 10.1080/00207721.2021.1919937.
- [23] S. Aberkane, R. Kara, S. Amari, Algebraic approaches for designing control laws of time-constrained networked conflicting timed event graphs, *European Journal of Control* 67 (2022) 100714. doi: <https://doi.org/10.1016/j.ejcon.2022.100714>.
- [24] S. Gaubert, J. Mairesse, Asymptotic analysis of heaps of pieces and application to timed petri nets, in : *Proceedings 8th International Workshop on Petri Nets and Performance Models (Cat. No.PR00331)*, IEEE Comput. Soc, 1999. doi: 10.1109/pnpm.1999.796562.
- [25] S. Gaubert, J. Mairesse, Modeling and analysis of timed petri nets using heaps of pieces, *IEEE Transactions on Automatic Control* 44 (4) (1999) 683–697. doi: 10.1109/9.754807.
- [26] M. Silva, R. Valette, *Petri nets and flexible manufacturing*, in : *European Workshop on Applications and Theory in Petri Nets*, Springer, 1988, pp. 374–417.
- [27] R. David, H. Alla, *Discrete, continuous, and hybrid Petri nets*, Vol. 1, Springer, 2010.
- [28] M. BOURCERIE, *Elaboration pour les systemes de production*, Ellipses, Février 2011.
- [29] F. Baccelli, G. Cohen, G. J. Olsder, J.-P. Quadrat, *Synchronization and linearity : an algebra for discrete event systems*.

- [30] S. Gaubert, *Introduction aux systèmes à événements discrets*, Notes de cours, Inria Rocquencourt (1999).  
URL <http://perso.ens-lyon.fr/eric.thierry/Graphes2010/gaubert1999.pdf>
- [31] M. Lhommeau, *Etude de systèmes à événements discrets dans l'algèbre  $(\max, +)$  : synthèse de correcteurs robustes dans un dioïde d'intervalles : synthèse de correcteurs en présence de perturbations*, Ph.D. thesis, Angers (2003).
- [32] F. Baccelli, G. Cohen, B. Gaujal, *Recursive equations and basic properties of timed petri nets*, Ph.D. thesis, INRIA (1991).
- [33] H. Hillion, J.-M. Proth, *Performance evaluation of job-shop systems using timed event-graphs*, *IEEE Transactions on Automatic Control* 34 (1) (1989) 3–9. doi: 10.1109/9.8644.
- [34] B. Trouillet, O. Korbaa, J.-C. Gentina, *Formal approach of FMS cyclic scheduling*, *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37 (1) (2007) 126–137. doi: 10.1109/tsmcc.2006.876063.
- [35] S. Lahaye, J. Komenda, J.-L. Boimond, *Compositions of  $(\max, +)$  automata*, *IFAC Proceedings Volumes* 45 (29) (2012) 61–66. doi: 10.3182/20121003-3-mx-4033.00013.
- [36] S. Lahaye, J. Komenda, J.-L. Boimond, *Compositions of  $(\max, +)$  automata*, *Discrete Event Dynamic Systems* 25 (1-2) (2014) 323–344. doi: 10.1007/s10626-014-0186-6.
- [37] T. J. Van den Boom, B. De Schutter, *Modelling and control of discrete event systems using switching max-plus-linear systems*, *Control Engineering Practice* 14 (10) 1199–1211. doi: 10.1016/j.conengprac.2006.02.006.
- [38] T. J. Van den Boom, B. De Schutter, *Modeling and control of switching max-plus-linear systems with random and deterministic switching*, *Discrete Event Dynamic Systems* 22 (3) (2011) 293–332. doi: 10.1007/s10626-011-0123-x.
- [39] B. Kersbergen, J. Rudan, T. van den Boom, B. D. Schutter, *Towards railway traffic management using switching max-plus-linear systems*, *Discrete Event Dynamic Systems* 26 (2) (2014) 183–223. doi: 10.1007/s10626-014-0205-7.
- [40] B. Addad, S. Amari, J.-J. Lesage, *Networked conflicting timed event graphs representation in  $(\max, +)$  algebra*, *Discrete Event Dynamic Systems* 22 (4) (2012) 429–449. doi: 10.1007/s10626-012-0136-0.

- 
- [41] W. M. Boussahel, S. Amari, R. Kara, *Analytic evaluation of the cycle time on networked conflicting timed event graphs in the  $(\max,+)$  algebra*, *Discrete Event Dynamic Systems* 26 (4) (2016) 561–581. doi: 10.1007/s10626-015-0220-3.
- [42] S. Moradi, L. Hardouin, J. Raisch, *Modeling and control of resource sharing problems in dioids*, in : *2016 13th International Workshop on Discrete Event Systems (WODES)*, IEEE, 2016. doi: 10.1109/wodes.2016.7497881.
- [43] G. Schafaschek, L. Hardouin, J. Raisch, *Optimal control of timed event graphs with resource sharing and output-reference update*, at - *Automatisierungstechnik* 68 (7) (2020) 512–528. doi: 10.1515/auto-2020-0051.
- [44] S. Moradi, L. Hardouin, J. Raisch, *Optimal control of a class of timed discrete event systems with shared resources, an approach based on the hadamard product of series in dioids*, in : *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017. doi: 10.1109/cdc.2017.8264373.
- [45] B. Addad, S. Amari, J.-J. Lesage, *Modélisation de réseaux de graphes d'événements temporisés avec conflits dans l'algèbre  $(\max,+)$* , in : *Conference Internationale Francophone en Automatique*, 2010, p. 22.
- [46] M. Lisa, R. kara, A. Said, *Modelling and analysis of discrete event systems with shared resources in  $(\max,+)$  algebra*, *Technical reports*, University of Mouloud Mammeri (2016).
- [47] B. Brandin, W. Wonham, *Supervisory control of timed discrete-event systems*, *IEEE Transactions on Automatic Control* 39 (2) (1994) 329–342. doi: 10.1109/9.272327.
- [48] P. J. Ramadge, W. M. Wonham, *Supervisory control of a class of discrete event processes*, *SIAM Journal on Control and Optimization* 25 (1) (1987) 206–230. doi: 10.1137/0325013.
- [49] R. Boukra, *Contributions à la modélisation, l'évaluation de performances et la commande des systèmes à événements discrets*, *Ph.D. thesis*, Université d'Angers (2013).

## Annexe 1. Les automates

### *Les autmates*

Les automates à états finis (AEF) sont des modèles mathématiques qui décrivent le comportement d'un système en termes d'états et de transitions entre ces états. Un AEF est constitué d'un ensemble fini d'états, d'un ensemble fini d'entrées ou de symboles d'entrée, d'un ensemble fini de sorties ou de symboles de sortie, et d'une fonction de transition qui définit les règles de passage d'un état à un autre en fonction des entrées. Les AEF peuvent être utilisés pour modéliser des systèmes simples ou complexes. Les systèmes simples sont souvent représentés par des AEF déterministes, où chaque transition est associée à une entrée unique et l'état suivant est déterminé de manière unique. Les systèmes complexes peuvent être représentés par des AEF non déterministes, où plusieurs transitions sont possibles pour une même entrée et l'état suivant est choisi de manière non déterministe. Les AEF sont également utilisés dans la vérification de modèles, où ils sont utilisés pour vérifier si un système satisfait certaines propriétés spécifiées. Des techniques telles que la vérification de modèles basée sur les automates à états finis ont été développées pour vérifier des propriétés telles que la sûreté, la vivacité et la conformité d'un système. Dans ce qui suit, nous introduisons quelques définitions et propriétés importantes des automates à états finis qui seront utiles dans le chapitre suivant.

### **Définition 3.13** *Alphabet chaîne :*

Un alphabet est un ensemble fini de symboles. Un mot sur un alphabet  $\Sigma$  est une séquence possiblement vide de symboles. La séquence vide est notée  $(\varepsilon)$ . Les autres séquences sont notées par la juxtaposition des symboles qui les composent.

### **Définition 3.14** *Automate fini :*

Un automate fini est un quintuplet  $B = (\Sigma, Q, \gamma, r, T)$

- $\Sigma$  est un ensemble fini de symbole appelé alphabet.
- $Q$  est un ensemble fini dont les éléments sont appelés états.
- $\gamma$  est une relation de  $\Sigma \times Q \times \Sigma$  appelé transition ou un ensemble de transitions.
- $r$  est un état de  $Q$  appelé état initial.
- $T$  est un sous-ensemble de  $Q$  appelé ensemble des états finis de  $A$ .

L'ensemble des transitions  $\gamma$  est une relation, c'est-à-dire un ensemble de triplets. Cet ensemble est nécessairement fini puisque  $Q$  et  $\Sigma$  sont finis. Un automate fini est constitué de composants

qui sont tous finis  $(\Sigma, Q, \gamma, T)$ , d'où le qualificatif finit

Soit l'exemple de la Figure 3.15 représentant un automate fini  $\beta$  défini par :

- $\Sigma = a, b$  C'est l'ensemble de symbole.
- $Q = q_0, q_1$  C'est l'ensemble fini des états.
- $\gamma = a, aa, aab, aaaab...$  c'est ensemble de transitions.
- $q_0$  est un état initial de  $\beta$ .
- $q_0$  est un états final de  $\beta$ .

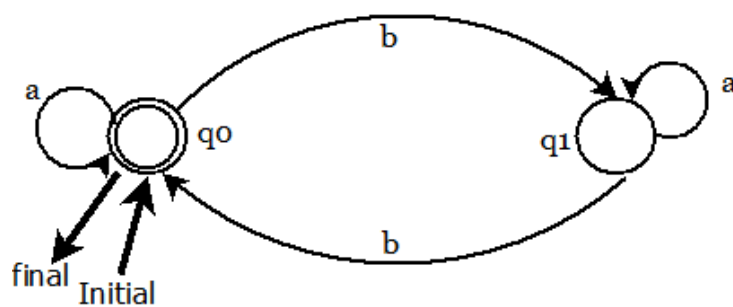


FIGURE 3.15: Automate fini

## Résumé :

L'objectif de cette thèse consiste à modéliser et proposer des lois de commande par retour d'état pour les systèmes à événements discrets avec des ressources partagées, en utilisant l'algèbre Max-plus. Nous considérons la classe des Graphes d'événements temporisés avec conflit (RGETC), qui est une classe de réseaux de Petri, soumis à des contraintes temporelles strictes. Tout d'abord, une formalisation algébrique en termes de systèmes linéaires Max-plus à commutation est proposée pour décrire le comportement dynamique d'un RGETC. Des conditions suffisantes pour l'existence de lois de commande garantissant ces contraintes temporelles sont établies pour les deux cas de RGETC : les RGETCs complètement observables et les RGETCs partiellement observables. Les lois de commande calculées sont des gains de retour qui peuvent être représentés par des places de commande qui supervisent le franchissement du système afin d'empêcher la violation des contraintes temporelles. Afin d'illustrer l'approche suggérée et montrer l'efficacité des lois de commande développées, deux exemples d'application sont présentés.

**Mots-clés :** Système à événements discret, Réseaux de Petri, Réseaux de Graphes d'Événements Temporisés avec Conflit , Systèmes Linéaires Max-plus à Commutation.

## Abstract :

The objective of this thesis is to study a modelling and a state feedback control problem of discrete event systems with shared resources using Max-plus algebra. We consider the class of Networked Conflicting Timed Event Graphs (NCTEGs), which are timed Petri nets with conflicts, subjected to strict temporal constraints. Firstly, an algebraic formalisation in terms of switching Max-plus linear systems is proposed to describe the dynamic behaviour of a NCTEG. Sufficient conditions for the existence of control laws to guarantee these strict time constraints are established for both cases of NCTEGs; fully observable and partially observable NCTEGs. The computed control laws are feedback gains that can be represented by control places which supervise the system to prevent from violating of temporal constraints. In order to illustrate the proposed approach and to demonstrate the effectiveness of the control laws, two application examples are presented.

**Keywords :** Discrete event systems, Petri nets, Networked conflicting timed event graphs, Switching Max-plus linear systems .

## ملخص :

الهدف من هذه الأطروحة هو نمذجة و اقتراح قوانين التحكم من خلال ملاحظات حالة أنظمة الأحداث المنفصلة ذات الموارد المشتركة باستخدام الجبر ماكس بلوس ، نحن نأخذ بعين الاعتبار فئة الرسوم البيانية للأحداث المتضاربة في الوقت المناسب ، و هي فئة من شبكات بتري ، تخضع لقيود زمنية صارمة ، أولاً ، تم اقتراح صياغة جبرية من حيث تبديل أنظمة ماكس بلوس الخطية لوصف السلوك الديناميكي للرسوم البيانية الشبكية للأحداث الموقوتة المتعارضة ، تم وضع شروط كافية لوجود قوانين رقابة لضمان هذه القيود الزمنية الصارمة لكلتا حالتها الرسوم البيانية الشبكية للأحداث الموقوتة المتعارضة : التي يمكن ملاحظتها كلياً و التي يمكن ملاحظتها جزئياً . قوانين الرقابة المحسوبة هي مكاسب تغذية مرتدة يمكن تمثيلها بأماكن رقابة تشرف على عبور النظام من أجل منع انتهاك القيود الزمنية . من أجل توضيح النهج المقترح وإثبات فعالية قوانين الرقابة ، تم تقديم مثالين تطبيقيين

**الكلمات المفتاحية :** أنظمة الأحداث المنفصلة ، وشبكات بتري ، والرسوم البيانية الشبكية للأحداث الموقوتة المتعارضة ، وتبديل أنظمة ماكس بلوس الخطية.