

Commande sous contraintes temporelles des réseaux de graphes d'événements temporisés en conflit

Sofiane Aberkane¹, Redouane Kara¹, and Said Amari²

¹ Laboratoire de Conception et Conduite des systèmes de Production, Université Mouloud Mammeri de Tizi-Ouzou, Route de Hasnaoua BP 17, 15000, Algérie.

`sofiane.aberkane@ummo.dz`, `redouk@yahoo.fr`

² LURPA, ENS Cachan, Univ. Paris-Sud, Université Paris 13, Sorbonne Paris Cité, Université Paris-Saclay, France.

`samari@ens-paris-saclay.fr`

Abstract

Dans ce papier, nous abordons le problème de modélisation et de commande des systèmes à événements discrets avec des ressources partagées représentés par une classe particulière des réseaux de Petri temporisés. Précisément, nous considérons des Réseaux de Graphes d'Événements Temporisés en Conflit (RGETC) soumis à des contraintes temporelles strictes. Premièrement, une formalisation algébrique en termes de systèmes à commutation Max-Plus est proposée pour décrire le comportement dynamique des RGETCs. Deuxièmement, des lois de commande en boucle fermée sont calculées pour garantir le respect de ces contraintes de temps imposées à certaines places du réseau. Des conditions suffisantes pour l'existence de telles lois de commande ont été fournies. Finalement, nous appliquons les résultats théoriques développés précédemment pour contrôler un système ferroviaire de croisement de train à temps critique.

1 Introduction

Le vocable systèmes à événements discrets désigne les systèmes dont l'évolution dépend de l'occurrence d'événements. On rencontre ces systèmes dans plusieurs domaines d'application tels que les systèmes de production, les réseaux de transport et les réseaux de communication. Pour modéliser ce type de système, on retrouve des outils comme: les réseaux de Petri, les files d'attente, les automates [7] et les chaînes de Markov. Chaque outil permet de traiter un problème d'analyse et/ou de commande particulier. L'outil Réseau de Petri (RdP) permet de traiter avec le même formalisme des problèmes de conception, d'analyse et de conduite des systèmes à événements discrets. En outre, ils permettent de modéliser les phénomènes de synchronisation, parallélisme et de concurrence rencontrés lorsqu'on s'intéresse à cette classe de système.

Les Graphes d'Événements Temporisés (GETs) sont une classe des RdPs qui modélise uniquement les situations de synchronisation et de parallélisme. La dynamique d'un GET peut être modélisée par des équations linéaires dans les algèbres tropicales (Max-plus ou Min-plus)[5][6]. Notons que les GETs ne permettent pas de modéliser des systèmes avec partage de ressources. Ceci est dû au fait que, dans un GET, chaque place comporte au plus une transition d'entrée et une transition de sortie. Une classe plus générale que les GETs est celle des réseaux de graphes d'événements temporisés en conflit (RGETCs). Ces modèles graphiques permettent de représenter le fonctionnement de partage de ressources dans le contexte des systèmes à événements discrets non déterministes. Dans la littérature, plusieurs travaux sont dédiés à l'analyse et l'étude de cette catégorie de systèmes et des résultats intéressants sont proposés [10, 17, 16, 9, 1, 2].

Dans [10] les auteurs ont proposés une modélisation sous forme de GETs d’ateliers production avec concurrence pour ressources du type job-shop et flow-shop. En particulier, ils proposent de modéliser les séquences d’entrée sur les machines par des circuits de contrôle. Un algorithme d’optimisation et une allocation cyclique de ressources sont proposés. Les auteurs dans [16, 17] s’intéressent au système de fabrication flexible avec partage de ressources. La méthode développée repose sur la construction de GET à partir d’un modèle du réseau de Petri en dupliquant des transitions communes. Les auteurs ont proposé dans [9] une procédure de détermination basée sur la complétion de modèles de tas. Pour revenir à la RDP à partir de laquelle le modèle de pièces est construit, cela signifie qu’une représentation déterministe (Max-plus) peut être obtenue si deux partages sont partagés. Dans les travaux [1, 2], les auteurs se sont intéressés à représenter le comportement dynamique de RGETCs par des équations Max-Plus récurrentes.

La synthèse de contrôleur pour les *SEDs* doit répondre à des spécifications pouvant se présenter sous différentes formes (contraintes sur l’état, échéances, intervalles de temps, durée de validité, ...). Dans la littérature, plusieurs approches ont été proposées pour la commande des SEDs sous différentes contraintes et spécifications.

De nombreux auteurs ont traité de la construction de contrôleurs pour garantir la satisfaction d’exigences temporelles critiques imposées pour le comportement du système. La contribution dans [8] est basée sur l’utilisation des automates temporisés pour modéliser à la fois le système et les spécifications. Les auteurs ont étendu le cadre de la théorie de Ramadge et Wonham [14] et ont défini le concept de contrôlabilité pour les SEDs temporisés. Ces approches sont intéressantes et pertinentes mais le problème de l’explosion combinatoire limite leur domaine d’application. Pour surmonter ce problème, souvent rencontré lorsqu’il s’agit de modèles automates, des méthodes basées sur l’utilisation des Graphes d’Evénements Temporisés et l’algèbre des dioïdes sont proposées dans [3, 4, 11, 15, 12]. Les auteurs dans [12, 15] traitent des problèmes de synthèse de contrôleurs par retour d’état pour garantir que les systèmes considérés évoluent sans violer les restrictions de temps imposées à l’état. Les travaux dans [3] et [4] concernent la synthèse de loi de commande en boucle fermée pour les graphes d’événements temporisés soumis à des contraintes de temps strictes. Dans ces approches, les contraintes de temps sont exprimées dans l’algèbre Max-Plus par des inégalités linéaires sur le vecteur d’état.

Dans ce travail, nous proposons une méthodologie pour la modélisation et la commande sous contraintes de temps des RGETCs. Premièrement, nous définissons un nouveau cadre de représentation analytique pour décrire le comportement des RGETCs. Cette modélisation nous permet par la suite de calculer, dans un second temps, des lois de commande en boucle fermée qui garantissent la satisfaction de contraintes temporelles strictes.

L’article est organisé de la manière suivante: La section 2 rappelle certaines notions de base sur les RdPs et l’algèbre (Max, +). Dans la section 3, une modélisation des RGETCs basée sur les systèmes Max-plus linéaires à commutation est présentée. La section 4 est consacrée à la synthèse de loi de commande pour les RGETCs sous contraintes de temps. Nous appliquons la méthode proposée à un système de croisement ferroviaire à temps critique à la section 5. Enfin, une conclusion et des perspectives sont données à la section 6.

2 Généralités sur les réseaux de Petri

Un réseau de Petri est un graphe biparti composé de deux types de sommets : les places et les transitions. Les places sont représentées par des cercles et les transitions par des barres (ou rectangles). Des arcs orientés relient certaines places à certaines transitions ou certaines

transitions à certaines places. Un arc ne relie jamais deux sommets de même nature. Chaque place peut contenir une ou plusieurs marques, représentées par des points ou jetons. Le marquage d'un réseau de Petri est un vecteur à composantes entières positives ou nulles, et dont la dimension est égale au nombre de places. La $j^{\text{ème}}$ composante de ce vecteur représente le nombre de marques que contient la $j^{\text{ème}}$ place du réseau de Petri.

De manière plus formelle, un RdP est un cinq-uplet $R = (P, T, Pres, Post, M_0)$ avec :

- P est un ensemble fini de places, $P = \{p_1, p_2, \dots, |P|\}$, où $|\cdot|$ représente la cardinalité d'un ensemble.
- T est un ensemble fini de transitions, $T = \{t_1, t_2, \dots, |T|\}$.
- $Pre : P \times T \rightarrow \{0, 1\}$ est une application dite d'incidence avant, telle que : $Pre(p_i, t_j)$ est le poids de l'arc qui relie la transition t_j à la place p_i .
- $Post : P \times T \rightarrow \{0, 1\}$ est une application dite d'incidence arrière, telle que : $Post(p_i, t_j)$ est le poids de l'arc qui relie la place p_i à la transition t_j .
- $M = [M_1, M_2, \dots, M_{|P|}]^T$ est le vecteur marquage avec M_i correspond au nombre de jetons dans la place p_i . Nous notons par $M_0 = [M_{0_1}, M_{0_2}, \dots, M_{0_{|P|}}]^T$ le marquage initial du RdP.

Une place p_i est dite place d'entrée (de sortie) d'une transition t_j s'il existe un arc orienté reliant p_i à t_j (resp. t_j à p_i).

Un RdP P-temporisé est un couple $N = (R, Tempo)$, ou :

- R est le RdP marqué.
- Tempo est une application de l'ensemble des places P vers l'ensemble des rationnels non négatifs Q_+ , $Tempo(p_i) = \tau_i$. Avec τ_i la temporisation associée à la place p_i .

Un Graphe d'Événements Temporisé (GET) est un RdP temporisé où chaque place possède au plus une transition en amont et une transition en aval.

Une place p_i est dite binaire (marquage sauf) si quelque soit l'évolution du RdP son marquage n'excède pas un jeton (i.e. $M_i \leq 1$).

Un conflit structurel correspond à l'existence d'une place qui possède au moins deux transitions de sortie. On désigne un conflit structurel par la paire formée d'une place et de l'ensemble de ses transitions de sortie $K = \{\tilde{p}, \{t_1, t_2, \dots\}\}$. Un conflit effectif noté $K^E = \{\tilde{p}, \{t_1, t_2, \dots\}, M_{\tilde{p}}\}$ est un conflit structurel avec un nombre de jetons $M_{\tilde{p}}$ dans la place \tilde{p} inférieur au nombre de ses transitions de sortie.

2.1 Modélisation Max-Plus linéaire des GETs

Un dioïde D est un ensemble muni de deux lois internes notées \oplus et \otimes telles que : (D, \oplus) est un semi-groupe commutatif et (D, \otimes) un semi-groupe. Les éléments neutres sont respectivement ε et e . La loi \otimes est distributive par rapport à la loi additive \oplus à gauche et à droite. \oplus est idempotente et ε est absorbant pour \otimes . Si de plus, \otimes est commutative alors le dioïde est dit commutatif.

L'ensemble des réels \mathcal{R} muni du max noté additivement et de la somme usuelle notée multiplicativement est un dioïde noté \mathcal{R}_{max} . Les deux opérations \oplus et \otimes sont définies comme suit : Pour $x, y \in \mathcal{R}_{max}$: $x \oplus y = \max(x, y)$ et $x \otimes y = x + y$. L'élément neutre de \oplus est $\varepsilon = -\infty$ et l'élément neutre de \otimes est $e = 0$. Dans le cas matriciel, on note par $\mathcal{R}_{max}^{n \times n}$ le dioïde des matrices à coefficients dans \mathcal{R}_{max} . Soient $A, B \in \mathcal{R}_{max}^{n \times n}$, nous avons :

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

$$(A \otimes B)_{ij} = \bigoplus_k (A_{ik} \otimes B_{kj})$$

La matrice nulle est représentée par ε , tous ses éléments sont égaux à ε . La matrice identité, notée I possède des éléments diagonaux égaux à e et les autres à ε .

Pour représenter le comportement dynamique d'un GET dans l'algèbre Max-plus, on associe à chaque transition du réseau considéré une fonction dateur. Pour distinguer les transitions d'entrées des autres transitions, on désigne par $u_j(k)$ la fonction dateur d'une transition d'entrée tu_j et par $x_i(k)$ la fonction dateur d'une transition interne t_i .

La représentation d'état Max-Plus des GETs (éventuellement après duplication des places contenant plus d'un jeton), est donnée par [6] :

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

avec, $A \in \mathcal{R}_{max}^{n \times n}$, $B \in \mathcal{R}_{max}^{n \times m}$. Les vecteurs $x(k)$ et $u(k)$ représentent respectivement le vecteur d'état (fonctions dateurs des transitions internes) et le vecteur de commande.

Théorème 1. [6] *La solution minimale de l'équation $x = A \otimes x \oplus B$, où ($A \in \mathcal{R}_{max}^{n \times n}$, $B \in \mathcal{R}_{max}^{n \times m}$), est donnée par $x = A^* \otimes B$, où la matrice $A^* = \bigoplus_{i \in \mathbb{N}} A^i$ est l'étoile de Kleene de A .*

3 Modélisation des RGETCs par des systèmes linéaires à commutation dans l'algèbre Max-Plus

Le comportement dynamique des graphes d'événements temporisés est décrit par des équations linéaires dans l'algèbre des dioïdes. Cependant, les GETs ne permettent pas de représenter les phénomènes de concurrences ou de choix multiples qui caractérisent les systèmes avec partage de ressources. Pour prétendre à l'analyse et à la commande de systèmes plus complexes une modélisation des conflits est nécessaire. Dans cette partie, les RGETCs sont représentés par des systèmes Max-plus linéaire à commutation. Cette modélisation a été introduite récemment par [13], en utilisant la notion de dateur généralisé et de chemins de jetons dans les places de conflit. Dans ce présent papier, la modélisation repose 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 commutateurs est basée sur le routage des jetons des places de conflit. Pour chaque position des commutateurs un modèle Max-plus linéaire est obtenu.

Définition 1. [2] *Un Réseau de Graphes d'événements Temporisés en Conflit (RGETC) est un ensemble de GETs (serveur, machine, robot..etc) noté $G = \{g_1, g_2, g_3, \dots, g_q\}$, connecté 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 \tilde{p}_i est notée $\tilde{\tau}_i$.*

D'un autre côté, chaque place de conflit \tilde{p}_i possède, au plus, une transition de sortie et une transition d'entrée appartenant à un GET.
 Un exemple de tels graphes est donné par la figure 1.

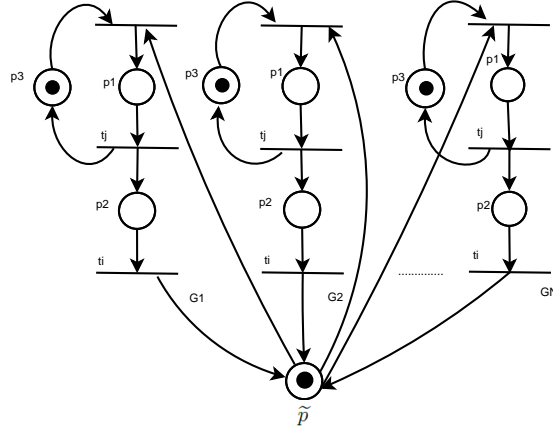


Figure 1: Exemple de RGETCs

Pour pouvoir décrire le comportement des RGETCs, on introduit deux types d'objets qui permettent de faciliter la modélisation.

Définition 2. Un commutateur arrière **B** est défini par une place de conflit vide \tilde{p} qui possède plusieurs transitions d'entrées t_j^{in} et une unique transition de sortie t^{out} . On dira que le couple (t_j^{in}, t^{out}) détermine la dynamique d'un réseau de Petri \mathcal{N} si le tir de la transition t^{out} est une conséquence du tir de la transition t_j^{in} , figure 2.

Définition 3. Un commutateur avant **F** est déterminé par une place de conflit vide \tilde{p} qui contient une unique transition en amont t^{in} et plusieurs transitions en aval t_j^{out} . on dénote que le paire (t^{in}, t_j^{out}) détermine la dynamique d'un réseau de Petri \mathcal{N} si le tir de la transition d'entrée t^{in} est suivi du tir de la transition de sortie t_j^{out} , figure 2.

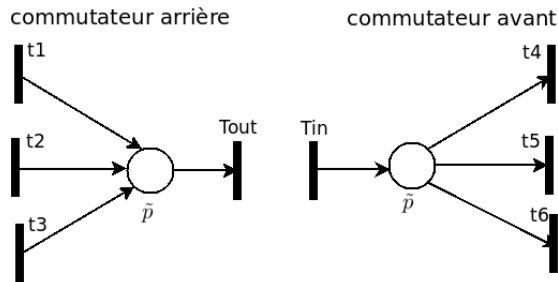


Figure 2: Commutateurs

Définition 4. Une configuration C_i d'un réseau de Petri \mathcal{N} correspond à une position donnée des commutateurs.

Soit $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_N\}$ est l'ensemble des configurations possibles des commutateurs (**B** et **F**) d'un réseau \mathcal{N} donné, avec N représente le nombre de configurations possibles. On peut toujours représenter une place de conflit par une composition constituée par deux commutateurs en série (**B** et **F**), comme montré à la figure 3.

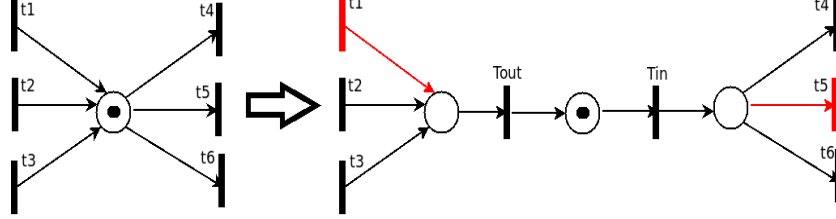


Figure 3: Exemple de configuration (t_1, t_5)

En plus de la notion de commutateur, on définit la notion de dateur généralisé et de système linéaire Max-plus à commutation qui vont être utiles pour la compréhension des résultats établis dans la suite de ce document.

Définition 5. [13] Nous attribuons à chaque transition t_i du RGETC G , un dateur généralisé $\theta_i(k)$ qui est une fonction non décroissante qui indique la date du dernier franchissement de la transition t_i pour la $k^{\text{ème}}$ exécution du graphe.

Cette notion de dateur généralisé est explicitée par l'exemple suivant :

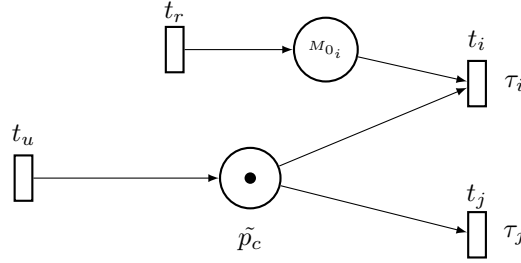


Figure 4: Exemple d'une situation de conflit

L'expression du dateur généralisé associée à la transition t_i est donnée par :

$$\theta_i(k+1) = \begin{cases} \tau_i \otimes \theta_u(k) \oplus \tau_i \otimes \theta_r(k - M_{0p_i}), & \text{si le jeton est alloué à } t_i \\ \theta_i(k), & \text{autrement} \end{cases} \quad (2)$$

Dans le théorème qui suit, on exprime l'évolution dynamique des RGETCs par des systèmes Max-plus linéaires à commutations introduits par [18]. Un système Max-Plus linéaire à commutation commute entre différents modes de fonctionnement. Dans chaque mode, la dynamique du système est donnée par un système linéaire dans l'algèbre $(max, +)$.

Théorème 2. Le comportement dynamique d'un RGETC est un système Max-Plus linéaire à commutation. Chaque configuration \mathcal{C}_i des commutateurs définie par un mode de fonction-

nement. La dynamique dans chaque mode est représentée par l'équation Max-Plus linéaire ci-dessous :

$$\boldsymbol{\theta}(k+1) = A_0^{q(k)} \otimes \boldsymbol{\theta}(k+1) \oplus A_1^{q(k)} \otimes \boldsymbol{\theta}(k) \oplus B_0^{q(k)} \otimes \mathbf{u}(k+1), \quad (3)$$

où:

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

N est le nombre total des modes;

k est le compteur d'événements;

$\boldsymbol{\theta}(k)$ est le vecteur d'état;

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

$A_0^{q(k)}, A_1^{q(k)} \in R_{max}^{\bar{n} \times \bar{n}}$;

$B_0^{q(k)} \in R_{max}^{n \times \bar{n}}$.

Preuve. La preuve de ce théorème est intuitive à partir de la définition de dateur généralisé et de la notion de configuration.

Remarque 1. L'équation 3 peut être écrite sous forme d'état en utilisant le Théorème 1.

$$\boldsymbol{\theta}(k+1) = A^{q(k)} \otimes \boldsymbol{\theta}(k) \oplus B^{q(k)} \otimes \mathbf{u}(k+1) \quad (4)$$

Remarque 2. Il est noté qu'une position donnée des commutateurs ne correspond pas toujours à un fonctionnement correct des RGETCs. La connaissance d'une séquence de transitions réalisable permet d'obtenir une séquence de configuration correcte.

3.1 Exemple

Considérons le réseau du graphe d'événements temporisés en conflit de la figure 5, constitué de deux GETs reliés par une place de conflit avec un marquage sauf.

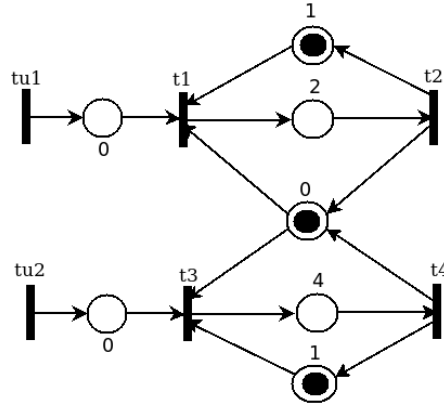


Figure 5: Modèle de RGETC

On appelle $\theta_i(k)$ la fonction dateur associée à la transition t_i et τ_j la temporisation associée à p_j tel que ($0 \leq i, j \leq 4$).

On a quatre configurations possibles pour les commutateurs. Chaque configuration correspond à un mode qu'on modélise par une équation linéaire Max-Plus :

pour le *MODE 1*, le jeton du conflit est consommé par le GET_2 au $(\mathbf{k})^{\text{éme}}$ franchissement et par le GET_1 au $(\mathbf{k} + 1)^{\text{éme}}$ franchissement. Nous obtenons les équations suivantes :

$$\begin{cases} \theta_1(k+1) = 1 \otimes \theta_2(k) \oplus \theta_4(k) \oplus u_1(k+1) \\ \theta_2(k+1) = 3 \otimes \theta_2(k) \oplus 2 \otimes \theta_4(k) \oplus 2 \otimes u_1(k+1) \\ \theta_3(k+1) = \theta_3(k) \\ \theta_4(k+1) = \theta_4(k) \end{cases}$$

La forme matricielle résultante est :

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

Pour le *MODE 2* : le jeton de conflit est consommé par le GET_1 au $(\mathbf{k})^{\text{éme}}$ franchissement et de même au $(\mathbf{k} + 1)^{\text{éme}}$, on a :

$$\begin{cases} \theta_1(k+1) = 1 \otimes \theta_2(k) \oplus u_1(k+1) \\ \theta_2(k+1) = 3 \otimes \theta_2(k) \oplus 2 \otimes u_1(k+1) \\ \theta_3(k+1) = \theta_3(k) \\ \theta_4(k+1) = \theta_4(k) \end{cases}$$

La forme matricielle est donnée par :

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

De manière similaire, les modèles des modes 3 et 4 sont données par les équation suivantes :

$$\text{Mode 3 : } \theta(k+1) = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & 1 \\ \varepsilon & 4 & \varepsilon & 5 \end{bmatrix} \otimes \theta(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & 4 \end{bmatrix} \otimes u(k+1)$$

$$\text{Mode 4 : } \theta(k+1) = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & 5 \end{bmatrix} \otimes \theta(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & 4 \end{bmatrix} \otimes u(k+1)$$

4 Synthèse d'une loi de commande pour les RGETCs sous contraintes temporelle

Plusieurs approches ont été développées pour la commande des GETs sous contraintes du temps [3, 4]. Les méthodes proposées sont basées uniquement sur l'utilisation des GETs et de l'algèbre des dioïdes. Dans cette étude, nous généralisons ces travaux pour résoudre le problème de commande des RGETCs soumis à des contraintes temporelles strictes.

4.1 Contrainte temporelle

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 dans la figure 6.

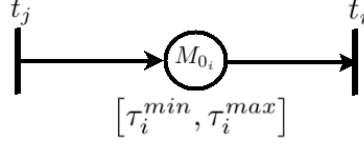


Figure 6: Contrainte temporelle

Cette contrainte se traduit dans l'algèbre \mathcal{R}_{max} par :

$$\tau_i^{min} \otimes x_j(k - M_{0_i}) \leq x_i(t) \leq \tau_i^{max} \otimes x_j(k - M_{0_i}) \quad (5)$$

Où M_{0_i} désigne le marquage initial de la place p_i . 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 :

$$x_i(k) \leq \tau_i^{max} \otimes x_j(k - M_{0_i}) \quad (6)$$

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 τ_i^{max} . On souhaite calculer des lois de commande $u(k)$ qui permettent de garantir le respect de cette contrainte de temps.

4.2 Formalisation du problème de commande

Dans le cadre de ces travaux, nous avons pris en compte quelques hypothèses que nous relaxons à court terme dans les perspectives futures.

Hypothèse 1. *On suppose dans la suite de méthodologie de commande proposée que :*

- Il y a une seule contrainte de temps dans chaque GET.
- Chaque GET g_i possède une unique transition de commande tu_i .

Hypothèse 2. *Toutes les transitions de RGETC sont commandables. (il existe un chemin de la transition de commande à chaque transition interne du GET).*

Hypothèse 3. *Il existe dans chaque GET g_i d'un RGETC au moins un chemin ρ_i vide (sans jetons) qui relie la transition source tu_i à la transition d'entrée t_j de la place sous contrainte de temps.*

Soit un RGETC qui contient q GET et L places de conflit avec des marquages saufs. Chaque GET de ce réseau comporte une seule contrainte de temps à respecter. On sait que le comportement dynamique d'un RGETC est donné sous la forme (7).

$$x(k+1) = A^{q(k)} \otimes x(k) \oplus B^{q(k)} \otimes u^{q(k)}(k+1) \quad (7)$$

où: $q(k) \in \{1, 2, \dots, N\}$, $\theta \in \mathcal{R}_{max}^n$, $A \in \mathcal{R}_{max}^{n \times n}$, $B \in \mathcal{R}_{max}^{n \times p}$, $k \in \mathbb{N}$.
Le $i^{ème}$ composante de l'équation (7) est exprimée par :

$$x_i(k+1) = \bigoplus_{\delta=1}^{q(k)} A_{i\delta}^N \otimes x_\delta(k) \oplus \bigoplus_{\delta=1}^Z (B^{q(k)})_{i\delta} \otimes u^{q(k)}(k+1) \quad (8)$$

Z est le nombre de transitions sources connectées au RGETC.

Les contraintes temporelles imposées pour chaque place d'un GET peuvent se traduire sous la forme de l'inéquation suivante :

$$x_i(k+1) \leq \tau_i^{max} \otimes x_j(k+1) \quad (9)$$

L'objectif est de synthétiser des lois commande (retours d'état ou feedbacks causaux) de la forme (10) pour satisfaire les contraintes temporelles représentées par les équations dans (9). Les feedbacks sont associés aux dateurs des transitions sources $u_i(k)$ pour empêcher la violation des contraintes temporelles :

$$u(k+1) = F_i \otimes x_i(k-1) \quad (10)$$

Les vecteurs F_i sont des gains de retour d'état à calculer pour satisfaire les contraintes temporelles 9.

Le Théorème 3 présente une condition suffisante pour l'existence d'une loi de commande par retour d'état pour la classe des RGETCs sous contraintes de temps.

Théorème 3. *Etant donné un RGETC qui satisfait les hypothèses (1, 2 et 3). Le contrôleur défini par l'équation :*

$$u^{q(k)}(k+1) = F^{q(k)} \otimes x(k),$$

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

assure le respect des contraintes les contraintes temporelles (9) si les conditions sont vérifiées.

$$\bigoplus_{\delta=1}^Z (B^{q(k)})_{i\delta} \leq \bigoplus_{\delta=1}^Z (\tau_i^{max} (B^{q(k)})_{j\delta})$$

Preuve. Du fait qu'il existe un chemin ρ_i qui relie la transition tu_i à la transition t_j , et à partir de la définition de la fonction dateur des transitions t_j . La relation entre la transition source et la transition d'entrée de la place sous contrainte est donnée par :

$$x_j(k+1) \geq \bigoplus_{\delta=1}^Z ((A^{q(k)})^{m\rho} \otimes B^l)_{j\delta} \cdot u^{q(k)}(k+1 - m\rho) \quad (11)$$

En considérant que $M_\rho = 0$ (qui est une hypothèse considérée initialement), l'équation (11) devient :

$$x_j(k+1) \geq \bigoplus_{\delta=1}^Z (B^{q(k)})_{j\delta} \cdot u^{q(k)}(k+1) \quad (12)$$

En remplaçant l'équation (8) dans (9), on obtient :

$$\bigoplus_{\delta=1}^N (A_{i\delta}^{q(k)}) x_\delta(k) \oplus \bigoplus_{\delta=1}^Z ((B^{q(k)})_{i\delta}) u^{q(k)}(k+1) \leq \tau_i^{max} x_j(k+1) \quad (13)$$

En prenant en considération l'hypothèse 1 et la fonction (12) on, on déduit :

$$\bigoplus_{\delta=1}^N (A_{i\delta}^{q(k)}) x(k) \oplus \bigoplus_{\delta=1}^Z ((B^{q(k)})_{i\delta}) u^{q(k)}(k+1) \leq \bigoplus_{\delta=1}^Z (\tau_{max}^{q(k)} B_{j\delta}^{q(k)}) u(k+1) \quad (14)$$

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

$$\bigoplus_{\delta=1}^N (A_{i\delta}^{q(k)}) x_{\delta}(k) \leq \bigoplus_{\delta=1}^Z (\tau_{max}^{q(k)} (B^{q(k)})_{j\delta}) u^{q(k)}(k+1) \quad (15)$$

$$\bigoplus_{\delta=1}^Z ((B^{q(k)})_{i\delta}) u^{q(k)}(k+1) \leq \bigoplus_{\delta=1}^Z (\tau_{max}^{q(k)} (B^{q(k)})_{j\delta}) u^{q(k)}(k+1) \quad (16)$$

Les deux conditions (15) et (16) peuvent s'écrire comme suit :

$$u^{q(k)}(k+1) \geq \bigoplus_{\delta=1}^N \left[(A^{q(k)})_{i\delta} - \tau_{max}^{q(k)} - B_{j\delta}^{q(k)} \right] x_{\delta}(k) \quad (17)$$

$$\bigoplus_{\delta=1}^Z ((B^{q(k)})_{i\delta}) u^{q(k)}(k+1) \leq \bigoplus_{\delta=1}^Z (\tau_{max}^{q(k)} (B^{q(k)})_{j\delta}) u^{q(k)}(k+1) \quad (18)$$

L'inéquation (17) représente la loi de commande et l'inéquation (18) correspond à la condition à vérifier.

5 APPLICATION : Gestion d'un croisement de lignes ferroviaires

Dans cette section, nous abordons un cas d'étude réaliste qui concerne un système de croisement des rails ferroviaires représenté par la figure 7. Ce réseau de transport est composé de trois lignes unidirectionnelles notées : L1, L2 et L3. Chaque ligne comporte trois feux de signalisation, présentées comme suit :

- L1 : (a á b).
- L2 : (c d e).
- L3 : (f f' g).

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

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

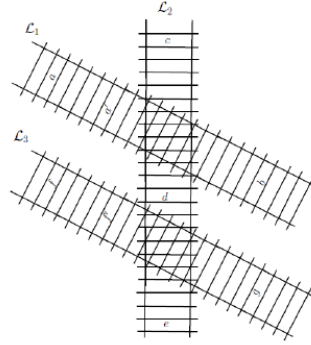


Figure 7: Réseau du croisement des 3 rails

présent sur les tronçons $[a\acute{a}]$ et $[\acute{a}b]$ (respectivement $[f\acute{f}]$ et $[f\acute{g}]$). Sur la ligne L2, un seul train est autorisé par tronçon.

Sur la ligne L2, un seul train est autorisé par tronçon. Cette gestion des trains sur le croisement est modélisé par le réseau de Petri de la figure 8.

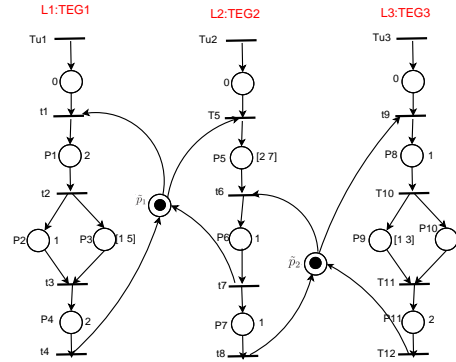


Figure 8: Modèle RGETC pour le système de Croisement de 3 rails ferroviaires

- Mode 1 : les jetons des conflits \tilde{p}_1 et \tilde{p}_2 sont consommés par le GET1 et le GET3.
- Mode 2 : les jetons des conflits \tilde{p}_1 et \tilde{p}_2 sont consommés par le GET2 .

5.1 Modèles Max-Plus linéaire à commutations du réseau ferroviaire

D'après la figure 8, nous avons deux places de conflit \tilde{p}_1 ET \tilde{p}_2 , ce qui se traduit par 16 modes de fonctionnement. Pour cette application, nous considérons uniquement deux modes pour illustrer les résultats développés.

suivantes :

$$x_3(k) \leq 5 \otimes x_2(k) \quad (19)$$

$$x_6(k) \leq 7 \otimes x_5(k) \quad (20)$$

$$x_{11}(k) \leq 3 \otimes x_{10}(k) \quad (21)$$

Une commande pour le MODE 1 :

Nous déterminons des lois de commande qui vont satisfaire simultanément les contraintes imposées sur les places p_3 et p_5 .

- Puisqu'il existe un chemin vide ρ_1 qui relie la transition tu_1 à la transition t_2 , nous avons :

$$B_{2\delta}^1 = [2 \quad \varepsilon \quad \varepsilon], B_{3\delta}^1 = [3 \quad \varepsilon \quad \varepsilon], A_{3\delta}^1 = [\varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad 3 \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon]$$

Nous vérifions que :

$$\bigoplus_{\delta=1}^N B_{3\delta}^1 = \bigoplus_{\delta=1}^N \tau_{p_2}^{max} \otimes B_{2\delta}^1 \quad (22)$$

Nous remarquons que la condition du **Théorème 3** est vérifiée. Alors il existe une commande qui satisfait la contrainte imposée sur le GET_1 , cette loi est donnée par l'équation suivante :

$$u_1^1(k+1) = \bigoplus_{\delta=1}^N (A_{3\delta}^1 - \tau_{p_2}^{max} - B_{2\delta}^1) \otimes x_\delta(k) \quad (23)$$

En remplaçant par les valeurs des matrices établies en section 5.1, on obtient la loi de commande ci-dessous :

$$u_1^1(k+1) = \bigoplus_{\delta=1}^N (A^1)_{3\delta} - 5 - 3 \otimes x_\delta(k) \quad (24)$$

Comme le terme $((A^1)_{3\delta} - 5 - 3) \leq 0$, nous pouvons le remplacer par l'élément neutre e . Finalement le contrôleur est donné par l'équation 25.

$$u_1^1(k+1) = e \otimes x_7(k) \quad (25)$$

En suivant la même démarche, on retrouve une loi de commande pour satisfaire la contrainte temporelle du GET 2. Ce feedback est donné comme suit :

$$u_3^1(k+1) = e \otimes x_8(k) \quad (26)$$

La commande pour le MODE 2:

Nous suivons le même raisonnement pour calculer la loi de commande du Mode 2.

- Nous remarquons qu'il existe également un chemin vide ρ_2 qui relie la transition tu_2 à la transition t_5 . Nous avons :

$$B_{5\delta}^2 = [\varepsilon \quad e \quad \varepsilon], B_{6\delta}^2 = [\varepsilon \quad 2 \quad \varepsilon]$$

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

Nous vérifions que :

$$\bigoplus_{\delta=1}^N B_{6\delta}^2 = \bigoplus_{\delta=1}^N \tau p_5^{max} \otimes B_{5\delta}^2 \quad (27)$$

La condition du **Théorème (3)** est bien vérifiée, alors il existe une loi de commande qui satisfait la contrainte imposée sur la place p_5 du GET_2 . Cette loi de commande est donnée par l'équation suivante :

$$u_2^2(k+1) = \bigoplus_{\delta=1}^N (A^2)_{6\delta} - \tau p_5^{max} - B_{5\delta}^2 \otimes x_\delta(k) \quad (28)$$

ce qui s'écrit :

$$u_2^2(k+1) = \bigoplus_{\delta=1}^N (A^2)_{3\delta} - 5 - B_{5\delta}^2 \otimes x_\delta(k) \quad (29)$$

Due, la loi de commande pour le mode 2 est donnée :

$$u_2^2(k+1) = e \otimes x_4(k) \bigoplus e \otimes x_{12}(k) \quad (30)$$

La loi de commande qui permet de satisfaire les contraintes de temps imposées au réseau de croisement ferroviaire est le vecteur suivant :

$$\begin{bmatrix} u_1^1(k+1) \\ u_3^1(k+1) \\ u_2^2(k+1) \end{bmatrix} = \begin{bmatrix} e \otimes x_7(k) \\ e \otimes x_8(k) \\ e \otimes x_4(k) \bigoplus e \otimes x_{12}(k) \end{bmatrix} \quad (31)$$

Ces lois de commande sont représentées par des places de contrôle (en double cercle sur la figure 9), marquées et temporisées et qui sont connectées au modèle initial du réseau. Le rôle de ces places de contrôle est de garantir le respect des contraintes temporelles pendant le fonctionnement du système.

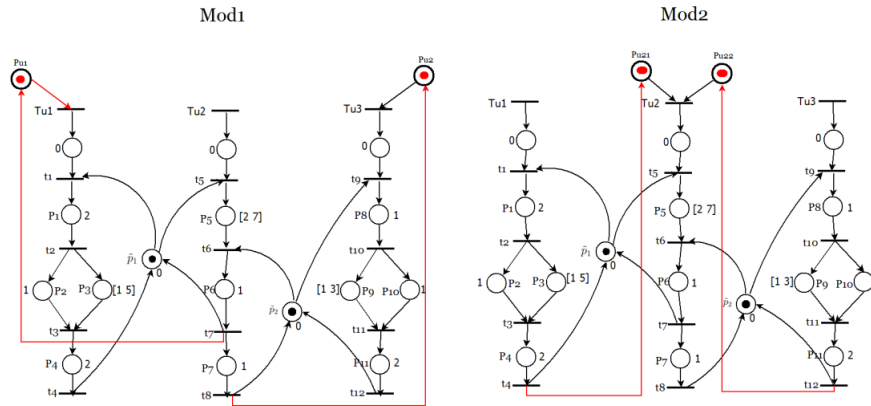


Figure 9: RGETC contrôlé du réseau de croisement des 3 rails

6 Conclusion

Dans ces travaux, nous nous sommes intéressés à la description analytique et à la commande sous contraintes temporelles des réseaux de graphes d'événements temporisés en conflit. Nous avons dans un premier temps, représenté le comportement des RGETCs par des équations Max-Plus linéaires à commutations. Par la suite, et sous certaines hypothèses, nous avons généralisé les travaux existants sur la commande des GETs soumis à des contraintes de temps strictes pour la classe des RGETCs. Plus précisément, une nouvelle approche algébrique de commande par retour d'état des RGETCs est proposée. Pour cela, des conditions suffisantes pour l'existence de lois de commande qui assurent le respect des contraintes temporelles strictes imposées à certaines places du RGETC sont définies. Les lois de commande calculées sont représentées par des places de contrôle connectées au RGETC initial qui permettent d'empêcher la violation des contraintes de temps. Finalement, nous avons illustré les approches théoriques développées pour la synthèse de commande d'un croisement de lignes ferroviaires soumis à des spécifications temporelles. Plusieurs perspectives de ces résultats sont envisageables pour des travaux futurs. Il serait intéressant de relaxer l'hypothèse sur la cyclicité des RGETCs et de considérer d'autres politiques de partage de ressources. On pourrait également généraliser les travaux pour traiter des contraintes temporelles imposées aux chemins des RGETCs.

References

- [1] Boussad Addad, Saïd Amari, and Jean-Jacques Lesage. Modélisation de réseaux de graphes d'événements temporisés avec conflits dans l'algèbre $(\max,+)$. In *Conference Internationale Francophone en Automatique*, page 22, 2010.
- [2] Boussad Addad, Saïd Amari, and Jean-Jacques Lesage. Networked conflicting timed event graphs representation in $(\max,+)$ algebra. *Discrete Event Dynamic Systems*, 22(4):429–449, 2012.
- [3] Saïd Amari, Isabel Demongodin, Jean Jacques Loiseau, and Claude Martinez. Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transactions on Automatic Control*, 57(2):462–467, 2012.
- [4] Abdourrahmane M Atto, Claude Martinez, and Saïd Amari. Control of discrete event systems with respect to strict duration: Supervision of an industrial manufacturing plant. *Computers & Industrial Engineering*, 61(4):1149–1159, 2011.
- [5] François Baccelli, Guy Cohen, and Bruno Gaujal. *Recursive equations and basic properties of timed Petri nets*. PhD thesis, INRIA, 1991.
- [6] François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. Synchronization and linearity: an algebra for discrete event systems. 1992.
- [7] Gerard Berry and Ravi Sethi. From regular expressions to deterministic automata. *Theoretical computer science*, 48:117–126, 1986.
- [8] Bertil A Brandin and W Murray Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, 1994.
- [9] Stéphane Gaubert and Jean 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)*, pages 158–169. IEEE, 1999.
- [10] H Proth Hillion and J-M Proth. Performance evaluation of job-shop systems using timed event-graphs. *IEEE transactions on automatic control*, 34(1):3–9, 1989.
- [11] Laurent Houssin, Sébastien Lahaye, and Jean-Louis Boimond. Control of $(\max,+)$ -linear systems minimizing delays. *Discrete Event Dynamic Systems*, 23(3):261–276, 2013.

- [12] Carlos Andrey Maia, CR Andrade, and Laurent Hardouin. On the control of max-plus linear system subject to state restriction. *Automatica*, 47(5):988–992, 2011.
 - [13] Liza Mameri, Redouan Kara, and Saïd Amari. Modelling and analysis of discrete event systems with shared resources in $(\max,+)$ algebra, submitted to mathematics of control, signals, and systems, 2019.
 - [14] Peter J Ramadge and W Murray Wonham. Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1):206–230, 1987.
 - [15] Karima Tebani, Saïd Amari, and 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):1295–1306, 2017.
 - [16] Benoit Trouillet, Ahmer Benasser, and J-C Gentina. Transformation of the cyclic scheduling problem of a large class of fms into the search of an optimized initial marking of a linearizable weighted t-system. In *Sixth International Workshop on Discrete Event Systems, 2002. Proceedings.*, pages 83–90. IEEE, 2002.
 - [17] Benoit Trouillet, Ouajdi Korbaa, and Jean-Claude Gentina. Formal approach of fms cyclic scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(1):126–137, 2007.
 - [18] Ton JJ van den Boom and Bart De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control engineering practice*, 14(10):1199–1211, 2006.
-