

# PROGRAMMATION MATHÉMATIQUE POUR LES RÉSEAUX DE PETRI T-TEMPORELS SAUFS

Thomas Bourdeaud'huy

Saïd Hanafi

LAGIS, Ecole Centrale de Lille  
Cité Scientifique, BP48  
59651 Villeneuve d'Ascq Cedex  
thomas.bourdeaud\_huy@ec-lille.fr

LAMIH, UVHC  
Le Mont Houy  
59313 Valenciennes Cedex 9  
said.hanafi@univ-valenciennes.fr

**RÉSUMÉ :** Dans cet article, nous présentons un modèle de programmation linéaire en nombres entiers pour la résolution du problème d'accessibilité dans les réseaux de Petri temporels. Nous nous intéressons plus précisément aux réseaux de Petri temporels non pondérés, à marquages saufs, dont les intervalles statiques sont finis, et dont le comportement est régi par une sémantique faible.

**MOTS-CLÉS :** Réseaux de Petri temporels, Programmation mathématique

## 1 INTRODUCTION

Cet article présente un modèle de programmation mathématique pour représenter le comportement des réseaux de Petri T-temporels. De tels réseaux permettent de représenter le comportement de systèmes dont le temps apparaît comme un paramètre quantifiable et continu. Les réseaux de Petri temporels sont bien adaptés à l'analyse de comportements, où ils apportent leur généralité. Ils permettent par exemple de modéliser la violation d'un comportement temporel (technique du chien de garde).

Comparés aux réseaux de Petri temporisés exprimant des spécifications « en durées », les réseaux de Petri temporels permettent d'exprimer des spécifications « en délais ». Ces réseaux permettent d'exprimer d'une manière simple la plupart des contraintes temporelles – durées comprises –, alors qu'il est délicat d'exprimer certaines contraintes temporelles à l'aide de durées seulement.

Notre objectif à terme est de faire des réseaux de Petri temporels un formalisme *générique* de spécification et de résolution de problèmes de recherche opérationnelle, capable notamment de décrire des problématiques d'*ordonnancement à fenêtres de temps*. Dans ce cadre, nous choisissons d'utiliser la *sémantique faible* qui nous paraît la plus adaptée au problème considéré, dans lequel les franchissements de transitions doivent correspondre à des choix d'optimisation et non à des propriétés du système considéré.

L'approche présentée ici s'applique uniquement aux réseaux temporels non pondérés et à marquages saufs, dont les intervalles statiques des transitions sont finis. Notre modèle permet de formuler et résoudre le pro-

blème d'accessibilité dans les réseaux de Petri temporels de manière incrémentale, en recherchant des séquences d'accessibilité de taille donnée dont la durée est minimale.

La suite de l'article est organisée comme suit. Dans la section 2, nous donnons la définition des réseaux de Petri temporels et des éléments concernant les principales techniques d'analyse de ces réseaux. Dans la section 3, nous développons progressivement notre modèle de programmation mathématique. Nous concluons en donnant quelques pistes de recherche théoriques et pratiques prometteuses.

## 2 RÉSEAUX DE PETRI TEMPORELS

### 2.1 Réseaux de Petri Places/Transitions (RdP P/T)

#### 2.1.1 Graphe biparti

Un réseau de Petri Places/Transitions (Diaz 2001)  $(R = (\mathbb{P}, \mathbb{T}, W), m_0)$  est un *graphe orienté biparti* où :

- $\mathbb{P}$  et  $\mathbb{T}$  sont deux ensembles finis de nœuds dénommés respectivement *places* et *transitions* avec  $|\mathbb{P}| = m$  et  $|\mathbb{T}| = n$  ;
- $W : \mathbb{P} \times \mathbb{T} \cup \mathbb{T} \times \mathbb{P} \rightarrow \mathbb{N}$  représente les *poids* des arcs reliant les nœuds ;
- $m_0 : \mathbb{P} \rightarrow \mathbb{N}$  associe à chaque place  $p \in \mathbb{P}$  un entier  $m_0(p)$  appelé *marquage* de  $p$ .

On note respectivement  $\bullet p$  et  $p^\bullet$  l'ensemble des pré-décesseurs et des successeurs de la place  $p$ , et réciproquement :  $\bullet t$  et  $t^\bullet$  représentent les places pré-décesseurs et successeurs de la transition  $t$ . Formellement :  $\bullet p = \{t \in \mathbb{T} \mid W(t, p) \neq 0\}$ ,  $p^\bullet = \{t \in$

$\mathbb{T} \mid W(p, t) \neq 0$ ,  $\bullet t = \{p \in \mathbb{P} \mid W(p, t) \neq 0\}$ ,  
 $t^\bullet = \{p \in \mathbb{P} \mid W(t, p) \neq 0\}$ . On emploiera les termes  
*amont* et *aval* pour désigner respectivement les pré-  
 décesseurs et les successeurs d'un noeud du réseau.

### 2.1.2 Représentation graphique

Dans la représentation graphique d'un réseau de Petri, les places sont représentées sous forme de cercles et les transitions sous forme de rectangles. On associe à chaque place  $p$  du réseau un nombre de jetons équivalent à son marquage  $m(p)$ . Les jetons sont représentés par des disques pleins à l'intérieur des places, ou plus simplement par un entier étiquetant la place. La figure 1 présente un exemple de réseau de Petri P/T, pour lequel  $m_0 = (1, 1, 0, 0, 0)^\top$ .

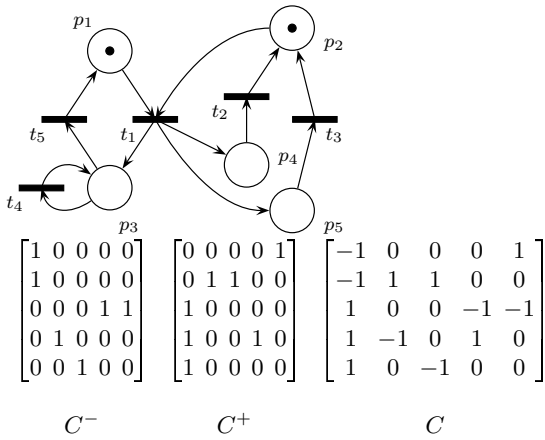


FIG. 1 – Exemple de réseau de Petri P/T et ses matrices d'incidence

### 2.1.3 Représentation Matricielle

Dans la suite, nous utiliserons des notations empruntées à l'algèbre linéaire pour rendre l'exposé plus concis :

- Les places du réseau (resp. les transitions) seront indicées par la lettre  $i$  (resp.  $j$ ), de  $p_1$  à  $p_M$  (resp.  $t_1$  à  $t_N$ );
- Les matrices d'incidence décrivant la structure du réseau  $C^-$ ,  $C^+$  et  $C \in \mathbb{N}^{M \times N}$  seront définies par :

$$\forall p_i \in \mathbb{P}, \quad \begin{cases} C_{ij}^- = k \Leftrightarrow W(p_i, t_j) = k \\ C_{ij}^+ = k \Leftrightarrow W(t_j, p_i) = k \end{cases} . \\
 \forall t_j \in \mathbb{T}, \quad C_{ij} = C_{ij}^+ - C_{ij}^-$$

- Les  $i^{\text{e}}$  vecteur ligne et  $j^{\text{e}}$  vecteur colonne extraits des matrices d'incidence  $C^-$ ,  $C^+$  et  $C$  seront dénotés respectivement par  $\vec{C}_i^-$ ,  $\vec{C}_j^-$ ,  $\vec{C}_i^+$ ,  $\vec{C}_j^+$  et  $\vec{C}_i$ ,  $\vec{C}_j$ ;
- Le vecteur  $\vec{e}_{t_k}$  (resp.  $\vec{e}_{p_k}$ ) désignera le vecteur canonique (ou caractéristique) associé à la transition  $t_k$  (resp.  $p_k$ ), dont la  $k^{\text{e}}$  composante vaut 1 et les autres sont nulles;

- Le vecteur  $\vec{m}_0$  désignera le vecteur de marquage du réseau, défini par  $\vec{m}_0 = (m_0(p_1), m_0(p_2), \dots, m_0(p_M))^\top \in \mathbb{N}^M$ .

### 2.1.4 Comportement

Une transition  $t_k$  est dite *tirable* pour un marquage  $m_0$  donné (on note  $m_0[t_k]$ ), si et seulement si les places en *amont* de la transition contiennent assez de jetons :

$$m_0[t_k] \begin{cases} \Leftrightarrow \forall p \in \mathbb{P}, & m_0(p) \geq W(p, t_k) \\ \Leftrightarrow & \vec{m}_0 \geq C^- \cdot \vec{e}_{t_k} \end{cases} .$$

Le tir de  $t_k$  produit alors un nouveau marquage  $m_1$  à partir du marquage  $m_0$  (on note  $m_0[t_k]m_1$ ) :

$$m_0[t_k]m_1 \begin{cases} \Leftrightarrow m_0[t_k] \wedge \forall p \in \mathbb{P}, m_1(p) = \\ & m_0(p) - W(p, t_k) + W(t_k, p) \\ \Leftrightarrow m_0[t_k] \wedge \vec{m}_1 = \vec{m}_0 + C \cdot \vec{e}_{t_k} \end{cases} .$$

### 2.1.5 Equation d'état

Les équations précédentes se généralisent aux franchissements de *séquences de transitions tirables*. Pour cela, on définit le vecteur caractéristique  $\vec{\sigma}$  associé à une suite  $\sigma = t_{\sigma_1} t_{\sigma_2} \dots t_{\sigma_r}$ , une suite de  $r$  transitions de  $\mathbb{T}$  comme le vecteur de  $\mathbb{N}^N$  dont la  $j^{\text{e}}$  composante est égale au nombre d'occurrences de la transition  $t_j$  dans la séquence  $\sigma$  :  $\vec{\sigma} = \sum_{j=1}^r \vec{e}_{t_{\sigma_j}}$ .

En utilisant ce vecteur caractéristique et les notations algébriques précédentes, on obtient une propriété classique des réseaux de Petri :

$$m_0[\sigma]m_f \Rightarrow \vec{M}_f = \vec{M}_0 + C \cdot \vec{\sigma} .$$

Cette équation est connue sous le nom d'*équation d'état* ou *équation fondamentale* des réseaux de Petri. La réciproque n'est pas vraie dans le cas général : l'existence d'un vecteur caractéristique  $\vec{\sigma}$  solution de l'équation d'état entre deux marquages  $m_0$  et  $m_f$  n'implique pas forcément que  $m_f$  soit accessible depuis  $m_0$ .

### 2.1.6 Steps

Il est encore possible de généraliser l'équation d'état aux tirs de séquences de *steps*  $\Phi = \varphi_1, \varphi_2, \dots, \varphi_n$ , où un step correspond à un tir simultané et réentrant de plusieurs transitions. Par exemple, la séquence de steps  $\Phi = \{t_1\}, \{t_1 + t_2 + t_3\}, \{2 \cdot t_1\}$  est tirable à partir du marquage  $m_0 = (2, 1, 0, 0, 0)^\top$  pour le réseau de la figure 1. Nous avons proposé dans (Bourdeaud'huy 2004) un modèle de programmation

mathématique utilisant la notion de step pour résoudre le problème d'accessibilité à horizon fixé dans les réseaux de Petri P/T.

## 2.2 Réseaux de Petri T-Temporels (RdP TT)

Les réseaux de Petri temporels ont été proposés par Merlin (Merlin 1974). La présentation qui suit s'inspire de (Diaz 2001).

Dans les réseaux de Petri T-Temporels, deux délais min et max (avec  $0 \leq \min \leq \max$ , min fini et max éventuellement infini) sont associés à chaque transition du réseau. Ces délais, pour une transition  $t$  du réseau, sont relatifs à la date à laquelle la transition  $t$  a été sensibilisée pour la dernière fois. Supposons qu'à une date  $\theta$  la transition  $t$  devienne sensibilisée ; la transition  $t$  ne peut alors être tirée avant la date  $\theta + \min$  et doit l'être, au plus tard, à la date  $\theta + \max$  (si max est fini), sauf si le tir d'une autre transition désensibilise la transition  $t$  avant que celle-ci ne soit tirée. Le tir des transitions est de durée nulle.

### 2.2.1 Définition

Formellement, un réseau de Petri temporel est un tuple  $(\mathbb{P}, \mathbb{T}, W, m_0, I_s)$  dans lequel  $(\mathbb{P}, \mathbb{T}, W, m_0)$  est un réseau de Petri et  $I_s : \mathbb{T} \rightarrow \mathbb{Q}^+ \times \mathbb{Q}^+ \cup \{\infty\}$  la fonction *intervalle statique*.

La fonction  $I_s$  associe, à chaque transition  $t$  du réseau, un intervalle à bornes rationnelles  $I_s(t) = [\min, \max]$  (avec  $0 \leq \min \leq \max$ ; max peut être infini). Le plus petit de ces temps est appelé *date statique de tir au plus tôt de  $t$*  (notée  $S_{\min}(t)$ ) et le plus grand *date statique de tir au plus tard de  $t$*  (notée  $S_{\max}(t)$ ).

La figure 2 représente un réseau de Petri temporel. Cet exemple sera utilisé dans la suite pour illustrer notre approche.

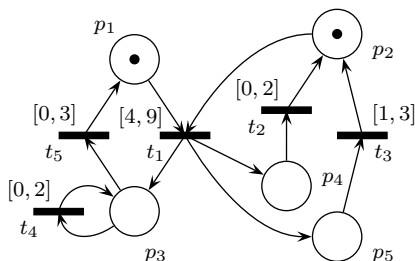


FIG. 2 – Exemple de réseau de Petri T-Temporel.

Dans un réseau temporel, franchir une transition sensibilisée  $t$  n'est permis que dans l'intervalle de temps qui lui est associé. Cet intervalle est relatif à la date de sensibilisation de la transition. Initialement (à la date 0), et si  $t$  est sensibilisée par le marquage initial, cet

intervalle coïncide avec son intervalle statique  $I_s(t)$ . Lorsque le réseau évolue (lorsque le temps s'écoule), l'intervalle de temps associé à une transition sensibilisée évolue également : il est décalé, vers l'origine des temps, d'une quantité égale à la durée écoulée depuis la date de sensibilisation de la transition (ceci, bien sûr, pour toutes les transitions sensibilisées).

Ces intervalles «dynamiques» sont exprimés comme une application  $I_d$  qui fait correspondre, à chaque transition  $t$ , l'intervalle de temps  $I_d(t)$  dans lequel elle peut être tirée. Les bornes inférieure et supérieure de l'intervalle  $I_d(t)$ , pour une transition  $t$ , sont désignées par la *date de tir au plus tôt*  $D_{\min}(t)$  et la *date de tir au plus tard*  $D_{\max}(t)$  de la transition.

Dans ce qui précède, nous avons implicitement associé un intervalle temporel au plus à chaque transition sensibilisée – qu'elle soit ou non multisensibilisée<sup>1</sup>. Cette interprétation de la sensibilisation, que nous qualifierons de standard, est précisée dans les paragraphes qui suivent, en définissant une notion d'état et une relation d'accessibilité entre états. D'autres interprétations de la multisensibilisation qui sortent du cadre de cet article conduisent à associer plusieurs variables temporelles aux transitions multisensibilisées.

## 2.3 Etats et règle de tir

Un état d'un réseau temporel est un couple  $E = (m, I_d)$  dans lequel  $m$  est un marquage et  $I_d$  l'application *intervalle de tir*. L'état initial  $E_0$  est constitué du marquage initial  $m_0$  et de l'application intervalle de tir  $I_0$  qui fait correspondre, à chaque transition  $t$  sensibilisée par  $m_0$ , son intervalle statique de tir  $I_s(t)$ , et à toute autre transition non sensibilisée, l'intervalle vide.

### 2.3.1 Sensibilisation

Le tir d'une transition  $t_j$ , à une date relative  $\theta$ , depuis un état  $E = (m, I_d)$ , est permis si et seulement si les conditions suivantes sont satisfaites :

1. La transition  $t_j$  est sensibilisée par  $m$  :

$$\vec{m} \geq \vec{C}_j^- \quad (1)$$

2.  $\theta$  n'est pas inférieure à la date de tir au plus tôt de  $t_j$  :

$$\theta \geq D_{\min}(t_j) \quad (2)$$

3.  $\theta$  n'est supérieure à la date de tir au plus tard d'aucune transition sensibilisée par  $m$  :

$$\forall t_k \in \mathbb{T}, \vec{m} \geq \vec{C}_{t_k}^- \Rightarrow \theta \leq D_{\max}(t_k) \quad (3)$$

<sup>1</sup>Une transition  $t_j$  est multisensibilisée par un marquage  $m$  si l'on a  $\vec{m} \geq k \cdot \vec{C}_j^-$ .

La condition (1) est celle qui autorise le tir dans les réseaux de Petri. Les conditions (2) et (3) résultent de l'obligation de tirer les transitions dans leur intervalle de tir. Notons que la condition (3) correspond à l'utilisation de la *sémantique forte* des RdP TT, qui impose que toute transition sensibilisée doit être tirée au pire à sa date d'échéance. Une autre sémantique, dite «*faible*» peut également être définie, dans laquelle la date de tir d'une transition n'est pas soumise aux conditions de sensibilisation d'autres transitions du réseau, comme l'exprime la condition suivante :

3. [Sémantique faible]  $\theta$  n'est pas supérieure à la date de tir au plus tard de  $t$  :

$$\theta \leq D_{\max}(t_j) \quad (4)$$

### 2.3.2 Franchissement

Rappelons que deux transitions  $t_j$  et  $t_k$  sont en conflit pour un marquage  $m$  si toutes deux sont sensibilisées par  $m$  mais que l'on a, pour une place  $p_i$  au moins,  $m(p_i) < C_{ij}^- + C_{ik}^-$ .

Le tir d'une transition sensibilisée  $t_j$ , à la date  $\theta$ , depuis l'état  $E = (m, I_d)$ , conduit en un état  $E' = (m', I'_d)$  déterminé comme suit :

- Le nouveau marquage  $m'$  est déterminé classiquement, par  $\vec{m}' = \vec{m} + C \cdot \vec{e}_{t_j}$ ;
- Le nouvel intervalle de tir  $I'_d(t_k)$ , pour toute transition  $t_k$ , est défini par :
  - si  $t_k$  est non sensibilisée par  $m'$ , alors  $I'_d(t_k)$  est vide,
  - si  $t_k$  est distincte de  $t_j$ , est sensibilisée par  $m$ , et n'est pas en conflit avec  $t_j$  pour  $m$ , alors :
 
$$I'(k) = \begin{cases} [\max(0, D_{\min}(t_k) - \theta), D_{\max}(t_k) - \theta] & \text{si } D_{\max}(t_k) \text{ est fini,} \\ [\max(0, D_{\min}(t_k) - \theta), \infty[ & \text{sinon,} \end{cases}$$
- sinon  $I'(t_k) = I_s(t_k)$ .

En d'autres termes, les transitions non sensibilisées par le nouveau marquage  $m'$  reçoivent des intervalles de tir vides ; les transitions distinctes de  $t_j$  et qui sont restées sensibilisées pendant le tir de  $t_j$  voient leur intervalle de tir décalé, vers l'origine du temps, de la valeur  $\theta$ , date relative à laquelle la transition  $t_j$  a été tirée (ces intervalles sont restreints, si nécessaire, aux valeurs de temps non négatives) ; toutes les autres transitions sensibilisées par  $m'$  reçoivent, pour intervalle de tir, leur intervalle statique. Notons que si  $t_j$  est restée sensibilisée pendant son propre tir, alors elle reçoit, pour intervalle, son intervalle statique.

## 2.4 Analyse de réseaux de Petri temporels

### 2.4.1 Analyse structurelle

L'ensemble des marquages atteints d'un RdP TT est un sous-ensemble des marquages atteints du RdP P/T sous-jacent. Par conséquent, les notions d'invariant de transitions et de places sont préservées entre le RdP TT considéré et son RdP P/T sous-jacent. De plus, la bornitude du RdP P/T sous-jacent constitue une condition suffisante pour celle du RdP TT concerné. Un premier ensemble de démarches d'analyse est donc constitué de méthodes structurelles. Le lecteur intéressé se reportera à (Roux 1985) pour plus de détails.

### 2.4.2 Analyse énumérative

La démarche de vérification formelle la plus classique pour les réseaux de Petri temporels est l'*analyse énumérative*. L'analyse énumérative s'intéresse à la définition et à la construction d'un *graphe de classes d'états* (Berthomieu & Diaz 1991). Par conséquent, elle permet d'étudier les propriétés dynamiques des RdP TT.

De manière intuitive, une classe d'états représentera un ensemble d'états obtenus, partant du marquage initial  $m_0$ , en tirant une séquence de transitions  $\sigma$ , cette dernière vérifiant un ensemble de contraintes temporelles. Les états membres d'une même classe auront alors le même marquage et le domaine de tir associé à la classe correspondra à la réunion des intervalles dynamiques des différents états constitutifs.

Cette approche souffre du problème classique de l'explosion combinatoire : le nombre d'états à construire peut grandir exponentiellement avec la taille du système considéré. De plus, la construction de chaque état nécessite une procédure de normalisation également coûteuse en temps de calcul.

### 2.4.3 Programmation Mathématique

Nous proposons dans cet article de développer une méthode d'analyse «*guidée par le but*», utilisant la programmation linéaire en nombres entiers, qui évite la construction du graphe de comportement complet du réseau de Petri temporel considéré. De ce fait, les problématiques d'explosion combinatoire sont «*repoussées*» à l'étape de résolution proprement dite, pour laquelle de nombreux outils efficaces ont été développés.

Dans cet article, nous nous intéresserons plus précisément au *problème d'accessibilité* dans les réseaux de Petri temporels. Nous développons un algorithme

de *semi-décision*<sup>2</sup>, en procédant à des recherches à horizon fixé.

Puisque nous cherchons à formuler puis résoudre des problèmes de recherche opérationnelle utilisant des fenêtres de temps, nous chercherons parmi les séquences d'accessibilité solution celles qui *minimisent un critère temporel* simple correspondant à la durée nécessaire pour atteindre le marquage souhaité.

Le problème considéré s'exprime au final comme suit :

« *Etant donné un entier  $K$ , un marquage initial  $m_0$  et un marquage final  $m_f$ , trouver la plus rapide séquence d'accessibilité  $\sigma$  constituée de  $K$  steps telle que  $m_0[\sigma]m_f$  ».*

### 3 MODÈLE INCRÉMENTAL POUR LES RDP TT

Dans cette section, nous construisons un modèle de programmation mathématique pour la résolution du problème d'accessibilité dans les réseaux de Petri temporels.

#### 3.1 Hypothèses de travail

De manière à définir un modèle mathématique linéaire simple pouvant être résolu *efficacement* par des techniques de programmation linéaire en nombres entiers, nous proposons de restreindre notre approche à un *sous-ensemble* des réseaux de Petri temporels. Ainsi, nous considérons dans la suite un réseau de Petri temporel vérifiant les conditions suivantes :

- Les intervalles statiques des transitions sont *finis*<sup>3</sup> :

$$\forall t \in \mathbb{T}, S_{\min}(t) \in \mathbb{Q}^+, S_{\max}(t) \in \mathbb{Q}^+ \quad (5)$$

Dans la suite, nous dénoterons respectivement par  $\overrightarrow{S_{\min}}$  et  $\overrightarrow{S_{\max}}$  les vecteurs de  $(\mathbb{Q}^+)^N$  dont la  $j^e$  composante est égale respectivement à  $S_{\min}(t_j)$  et  $S_{\max}(t_j)$ .

- Le réseau de Petri est *non pondéré* :

$$\forall i \in \llbracket 1, M \rrbracket, \forall j \in \llbracket 1, N \rrbracket, C_{ij}^- \in \{0, 1\}, C_{ij}^+ \in \{0, 1\} \quad (6)$$

- Le marquage initial du réseau de Petri est *sauf* :

$$\forall p \in \mathbb{P}, m_0(p) \in \{0, 1\} \quad (7)$$

Notons que même soumis aux contraintes précédentes, les réseaux de Petri temporels que nous manipulons restent raisonnablement expressifs, et

<sup>2</sup>Si le marquage est accessible, notre démarche finit par trouver une séquence d'accessibilité, mais ne peut conclure dans le cas inverse.

<sup>3</sup>Avec bien entendu :  $\forall t \in \mathbb{T}, S_{\min}(t) \leq S_{\max}(t)$ .

couvrent une grande partie des réseaux utilisés couramment. Comme nous le verrons par la suite, les deux premières conditions vont permettre des linéarisations utilisant un faible nombre de variables et de contraintes additionnelles, ce qui permettra d'obtenir un modèle mathématique performant. La dernière condition permettra de ne considérer que les séquences de franchissement menant à des marquages saufs, ce qui est également nécessaire pour assurer la consistance du modèle proposé.

#### 3.2 Principe de modélisation

Dans la présentation des réseaux de Petri temporels de la section 2.2, le comportement d'un réseau de Petri est défini à partir de la mise à jour des intervalles dynamiques de tirs de ses *transitions*. Cette approche conduit à considérer le cas de transitions en conflit *effectif*, i.e. des cas de conflits qui dépendent du marquage et pas uniquement des propriétés structurelles initiales du réseau, ce qui a tendance à complexifier les approches de modélisation. L'intérêt de cette approche est de prendre en compte naturellement le comportement de réseaux de Petri dont les marquages peuvent présenter plusieurs jetons par place.

Au contraire, nous proposons dans cet article un modèle fondé sur une structure de données associée aux *places*. Au prix d'une limitation aux réseaux de Petri temporels dont le comportement reste *sauf* (ce qui est nécessaire pour associer une unique date à chaque place), ce principe de modélisation nous permet de ne pas avoir à considérer le calcul d'intervalles dynamiques et ses problématiques de conflits de transitions.

Notre modèle se fonde sur une propagation incrémentale de contraintes correspondant à un nombre croissant d'instantants de tirs des transitions du réseau de Petri temporel. Nous avons proposé un mécanisme de construction similaire dans (Bourdeaud'huy, Hanafi & Yim 2006) pour les réseaux de Petri temporisés. Comme dans cet article, nous proposons ici de considérer le franchissement de *steps*, i.e. de plusieurs transitions à la fois. Cependant, nous nous limiterons à l'étude de steps *non réentrants* puisque nous souhaitons rechercher des exécutions laissant le réseau dans un état *sauf*. Finalement, les steps considérés seront donc à valeurs binaires.

À chaque franchissement, le step est contraint à respecter les conditions de franchissement dans un réseau de Petri temporel, dans le cadre de la *sémantique faible* :

- Validité *structurelle* : il y a un jeton dans chaque place en amont des transitions tirées ;
- Validité *temporelle* : les jetons utilisés par le franchissement du step respectent les intervalles tem-

poels associés à chacune des transitions tirées lors de ce franchissement.

L'état du réseau est ensuite recalculé après chaque franchissement. Cet état se compose :

- Du *vecteur de marquage* indiquant la présence d'un jeton dans une place. Nous imposons à ce vecteur de prendre ses valeurs dans  $\{0, 1\}$  de manière à ne rechercher que parmi les comportements qui présentent des marquages *saufs*<sup>4</sup>.
- D'un *vecteur d'âge* des jetons indiquant la durée depuis laquelle un jeton est présent dans la place concernée.

Dans la suite, nous proposons pour chacune des conditions et contraintes précédentes une formulation *linéaire* permettant d'étudier le problème d'accessibilité dans les RdP TT.

### 3.3 Notations

Les notations qui seront utilisées pour exprimer le modèle mathématique sont données ci-dessous :

- Le nombre total d'instants de tir est dénoté par  $K \in \mathbb{N}^+$  ;
- Le marquage initial  $m_0$  est donné à la date 0 sans nuire à la généralité. Un premier franchissement de step  $\sigma_1$  permet d'accéder au marquage  $m_1$  depuis  $m_0$  ;
- Le  $k^{\text{e}}$  step franchi  $\sigma_k$  est dénoté par son vecteur caractéristique  $\vec{\sigma}_k \in \{0, 1\}^N$  ;
- L'état atteint après le  $k^{\text{e}}$  step franchi est défini par son vecteur de marquage  $\vec{m}_k \in \{0, 1\}^M$  et son vecteur d'âge  $\vec{a}_k \in \mathbb{N}^M$  ;
- Les dates des franchissements des steps du réseau sont dénotées par  $v_k \in \mathbb{R}^+$ . Pour simplifier l'écriture des équations, on utilisera également la durée écoulée entre deux franchissements successifs :  $\Delta_{v_k} = v_k - v_{k-1} \in \mathbb{R}^+$  (avec éventuellement,  $v_k = v_{k-1}$ ).

Pour faciliter la lecture de cet article, les notations précédentes sont reprises dans le tableau 1.

$$\begin{array}{c}
 m_0 \quad \left| \begin{array}{c} \longrightarrow \\ \sigma_1 \end{array} \right| \begin{array}{c} m_1 \dots m_k \\ \longrightarrow \\ \sigma_{k+1} \\ v_{k+1} \end{array} \left| \begin{array}{c} \longrightarrow \\ \sigma_K \\ v_K \end{array} \right| m_K \\
 v_0 = 0 \quad \left| \begin{array}{c} \longrightarrow \\ v_0 + \Delta_{v_1} \end{array} \right| \begin{array}{c} a_1 \dots a_k \\ \longrightarrow \\ a_{k+1} \dots a_{K-1} \end{array} \left| \begin{array}{c} \longrightarrow \\ a_K \end{array} \right| a_K
 \end{array}$$

TAB. 1 – Schéma synoptique des notations

<sup>4</sup>Même dans le cas d'un réseau de Petri autorisant des comportements amenant plusieurs jetons dans la même place, comme c'est le cas par exemple pour le réseau de Petri de la figure 2.

## 3.4 Mise en équation

### 3.4.1 Validité structurelle

Cette condition est classique dans les réseaux de Petri. Pour chaque step, il doit y avoir assez de jetons dans les places en amont de la transition. L'équation correspondante est :

$$\forall k \in \llbracket 0, K-1 \rrbracket, \vec{m}_k \geq C^- \cdot \vec{\sigma}_{k+1} \quad (8)$$

### 3.4.2 Validité temporelle

Nous avons choisi de représenter l'état du réseau de Petri à partir de considérations portant sur ses *places* (marquage et âge des jetons). Pour traduire les règles de tir décrites dans la section 2.3, il faut commencer par exprimer les contraintes associées aux *intervalles dynamiques* des transitions à l'aide des variables d'état dont nous disposons.

Pour vérifier si une transition  $t_j$  franchie au step  $k+1$  vérifie les conditions temporelles définies plus haut, nous exprimons la *durée* depuis laquelle elle est sensibilisée à la date  $v_{k+1}$ . Cette durée est exactement égale à l'âge du *dernier jeton* arrivé en amont de  $t_j$ , à la date  $v_{k+1}$  (juste avant le tir du step  $\sigma_{k+1}$ ), soit :  $\Delta_{v_{k+1}} + \min_{p_i \in \bullet t_j} \{\vec{a}_k(i)\}$ .

Les conditions portant sur les intervalles dynamiques de tir des transitions (sémantique faible) correspondent donc à l'inéquation :

$$\forall t_j \text{ tirée au step } k+1, S_{\min}(t_j) \leq \Delta_{v_{k+1}} + \min_{p_i \in \bullet t_j} \{\vec{a}_k(i)\} \leq S_{\max}(t_j) \quad (9)$$

L'inéquation précédente est à ajouter au modèle en cours de construction si la transition  $t_j$  est effectivement tirée dans le step  $\sigma_{k+1}$ , c'est à dire si  $\vec{\sigma}_{k+1}(t_j) = 1$ . Pour traduire cette condition de manière *linéaire* et *uniforme* sur l'ensemble des transitions du réseau, nous utiliserons les inéquations suivantes, où  $B \in \mathbb{N}$  est choisi suffisamment grand<sup>5</sup>.

$$\forall j \in \llbracket 1, N \rrbracket,$$

$$B \cdot (\vec{\sigma}_{k+1}(j) - 1) \leq \Delta_{v_{k+1}} + \min_{p_i \in \bullet t_j} \{\vec{a}_k(i)\} - \vec{S}_{\min}(j) \quad (10)$$

$$\Delta_{v_{k+1}} + \min_{p_i \in \bullet t_j} \{\vec{a}_k(i)\} - \vec{S}_{\max}(j) \leq B \cdot (1 - \vec{\sigma}_{k+1}(j)) \quad (11)$$

De manière évidente, lorsque  $t_j$  est tirée lors du step  $\sigma_{k+1}$ , la quantité  $\vec{\sigma}_{k+1}(j) - 1$  est nulle, et l'on retrouve

<sup>5</sup>On prendra par exemple pour  $B$  une valeur supérieure à la durée maximale de l'ordonnancement à réaliser, obtenue par des considérations de sens physique sur le système considéré.

les inégalités décrites à l'équation (9). Au contraire, si  $t_j$  n'est pas tirée lors du step  $\sigma_{k+1}$ , et pourvu que  $B$  soit suffisamment grand, les équations (10) et (11) sont trivialement vérifiées.

Pour conclure sur la formulation de la condition de validité temporelle, nous proposons une linéarisation du calcul du minimum des composantes d'un vecteur.

**Calcul de la composante minimale d'un vecteur** Nous nous fondons sur l'énoncé suivant :

**Proposition 1 (Composante minimale).** *Soit  $\vec{X} \in \mathbb{N}^M$  et  $B \in \mathbb{N}$  « suffisamment grand »<sup>6</sup>. Soit  $\varepsilon \in \mathbb{N}$ , et  $\forall i \in \llbracket 1, M \rrbracket, \mu_i \in \{0, 1\}$ , tels que :*

$$\forall i \in \llbracket 1, M \rrbracket, \quad \varepsilon \leq \vec{X}(i) \quad (12)$$

$$\forall i \in \llbracket 1, M \rrbracket, \quad \vec{X}(i) - \varepsilon \leq (1 - \mu_i) \cdot B \quad (13)$$

$$\sum_{i=1}^M \mu_i = 1 \quad (14)$$

On a alors :  $\varepsilon = \min_{i \in \llbracket 1, M \rrbracket} \vec{X}(i)$ .

*Démonstration.* Nous allons montrer que les équations précédentes admettent toujours une solution  $(\varepsilon, (\mu_i)_{i \in \llbracket 1, M \rrbracket})$ , telle que  $\varepsilon$  soit bien égal au minimum des composantes du vecteur  $\vec{X}$ .

- Supposons que les équations (12) à (14) soient vérifiées pour les valeurs  $(\varepsilon, (\mu_i)_{i \in \llbracket 1, M \rrbracket})$ . Puisque les variables  $\mu_i$  sont binaires, l'équation (14) impose qu'il existe un unique  $k \in \llbracket 1, M \rrbracket$  t.q.  $\mu_k = 1$ . L'équation (13) impose alors :  $\vec{X}(k) - \varepsilon \leq 0$ , et l'équation (12) donne :  $\varepsilon \leq \vec{X}(k)$ . La composante  $\varepsilon = \vec{X}(k)$  est inférieure à toutes les autres composantes de  $\vec{X}$ , c'est donc bien le minimum des composantes de ce vecteur.
- Inversement, définissons  $\chi = \min_{i \in \llbracket 1, M \rrbracket} \vec{X}(i)$ , et  $k \in \llbracket 1, M \rrbracket$  tel que  $\chi = \vec{X}(k)$ . Nous allons montrer que le couple<sup>7</sup>  $(\varepsilon = \chi, (\mu_i = \delta_i^k)_{i \in \llbracket 1, M \rrbracket})$ , est solution des équations précédentes. Les équations (12) et (14) sont trivialement vérifiées. Considérons alors une composante quelconque  $j$  de  $\llbracket 1, M \rrbracket$ .
  - Si  $\vec{X}(j) > \chi$  (et donc  $j \neq k$ ), les inéquations (12) induisent  $\vec{X}(j) - \varepsilon > 0$ , et l'inéquation (13) donne :  $\mu_j = 0 = \delta_j^k$ ;
  - Si  $\vec{X}(j) = \chi$ , l'équation (13) est validée pour toute valeur de  $\mu_j$ .  $\square$

En utilisant la propriété précédente restreinte aux composantes du vecteur d'âge correspondant aux

<sup>6</sup>Une borne minimale pour  $B$  peut bien entendu être exhibée *a priori*, suivant le domaine de valeur des composantes de  $\vec{X}$ . Par exemple,  $B > \max_{i \in \llbracket 1, M \rrbracket} X(i)$  suffit.

<sup>7</sup>où  $\delta_i^j$  désigne le symbole de Kronecker, tel que  $\delta_i^j = 1$  si  $i = j$  et 0 sinon.

places en *amont* de chacune des transitions  $t_j$  considérées, nous obtenons finalement une formulation linéaire des conditions de validité temporelles (10) et (11). Cette formulation est présentée dans la figure 3.

Nous procédons dans la suite à l'expression des contraintes permettant de *mettre à jour* les vecteurs d'états après chaque franchissement.

### 3.4.3 Mise à jour du vecteur de marquage

La mise à jour du vecteur de marquage pour les réseaux de Petri temporels s'exprime de la même manière que pour les réseaux de Petri classiques. Nous imposons toutefois une contrainte supplémentaire permettant de ne considérer que des franchissements menant à des marquage saufs. Les contraintes à imposer sont donc :

$$\forall k \in \llbracket 0, K - 1 \rrbracket, \quad \vec{m}_{k+1} = \vec{m}_k + C \cdot \vec{\sigma}_{k+1} \quad (15)$$

$$\forall k \in \llbracket 0, K - 1 \rrbracket, \quad \vec{m}_{k+1} \in \{0, 1\}^M \quad (16)$$

### 3.4.4 Mise à jour du vecteur d'âge

Le vecteur d'âge des jetons est mis à jour lors de chaque franchissement. Comme indiqué dans la définition des réseaux de Petri temporels en section 2.3, deux types de mises à jour sont possibles :

- « *Vieillessement* » des âges des jetons qui n'ont pas bougé;
- « *Réinitialisation* » des âges associés aux places dans lesquelles un *nouveau jeton* est apparu<sup>8</sup>;

Pour procéder à la mise à jour du vecteur d'âge de  $\vec{a}_k$  à  $\vec{a}_{k+1}$  suite au franchissement du step  $\vec{\sigma}_{k+1}$ , nous considérons tout d'abord deux inégalités valides<sup>9</sup> :

$$\vec{a}_{k+1} \geq 0 \quad (17)$$

$$\vec{a}_{k+1} \leq \vec{a}_k + \Delta_{v_{k+1}} \cdot \vec{1}_M \quad (18)$$

L'équation (17) toujours vérifiée nous sera utile pour forcer la remise à zéro de l'âge d'un jeton par un double encadrement par zéro. L'équation (18) également toujours vraie servira pour le « *vieillessement* » des jetons.

Nous imposons alors les inégalités suivantes, où  $B \in \mathbb{N}$  est suffisamment grand :

$$\vec{a}_{k+1} \leq B \cdot (\vec{1}_M - C^+ \cdot \vec{\sigma}_{k+1}) \quad (19)$$

<sup>8</sup>Notons que cette formulation tient compte naturellement des problématiques de « *transitions en conflit* », puisque si une transition  $t_j$  en conflit avec  $t_k$  est tirée, menant à un marquage qui *re-sensibilise*  $t_k$ , l'âge d'un des jetons en amont de  $t_k$  aura été remis à 0, ce qui correspond bien à fixer le nouvel intervalle dynamique de tir de  $t_k$  à sa valeur statique.

<sup>9</sup>Où  $\vec{1}_M$  représente le vecteur unitaire de dimension  $M$ .

$$\overrightarrow{a_{k+1}} - \overrightarrow{a_k} - \Delta_{v_{k+1}} \cdot \overrightarrow{1_M} \geq -B \cdot C^+ \cdot \overrightarrow{\sigma_{k+1}} \quad (20)$$

Puisque nous travaillons avec des steps menant à des marquages *saufs*, la quantité  $(C^+ \cdot \overrightarrow{\sigma_{k+1}})$  est faite de composantes purement binaires, et représente les jetons ajoutés par le franchissement du step  $\sigma_{k+1}$  dans les places en aval des transitions franchies. Considérons alors une place  $p_i \in \mathbb{P}$ , et les  $i^e$  composantes des inégalités vectorielles (17) à (20).

Si une transition franchie lors du step  $\sigma_{k+1}$  ajoute un jeton dans  $p_i$ , la  $i^e$  composante du vecteur  $(C^+ \cdot \overrightarrow{\sigma_{k+1}})$  vaut 1, et les équations précédentes deviennent :

$$\overrightarrow{a_{k+1}}(i) \geq 0 \quad (21)$$

$$\overrightarrow{a_{k+1}}(i) \leq \overrightarrow{a_k}(i) + \Delta_{v_{k+1}} \quad (22)$$

$$\overrightarrow{a_{k+1}}(i) \leq 0 \quad (23)$$

$$\overrightarrow{a_{k+1}} - \overrightarrow{a_k} - \Delta_{v_{k+1}} \geq -B \quad (24)$$

Les équations (21) et (23) induisent la remise à 0 de la composante du vecteur d'âge  $\overrightarrow{a_{k+1}}(i)$ , puisqu'un nouveau jeton est apparu dans cette place. Les équations (22) et (24) sont trivialement vérifiées.

Inversement, si le franchissement du step  $\sigma_{k+1}$  ne produit aucun jeton dans la place  $p_i$ , la  $i^e$  composante de  $(C^+ \cdot \overrightarrow{\sigma_{k+1}})$  est nulle et les équations précédentes deviennent :

$$\overrightarrow{a_{k+1}}(i) \geq 0 \quad (25)$$

$$\overrightarrow{a_{k+1}}(i) \leq \overrightarrow{a_k}(i) + \Delta_{v_{k+1}} \quad (26)$$

$$\overrightarrow{a_{k+1}}(i) \leq B \quad (27)$$

$$\overrightarrow{a_{k+1}}(i) - \overrightarrow{a_k}(i) - \Delta_{v_{k+1}} \geq 0 \quad (28)$$

Les équations (25) et (28) impliquent le « *vieillessement* » de la  $i^e$  composante du vecteur d'âge, de la durée  $\Delta_{v_{k+1}}$  écoulée depuis le franchissement précédent :  $\overrightarrow{a_{k+1}}(i) = \overrightarrow{a_k}(i) + \Delta_{v_{k+1}}$ . Les équations (26) et (27) sont trivialement vérifiées.

Notons que le vieillissement du vecteur d'âge de  $p_i$  se produit même si aucun jeton n'est présent dans la place correspondante, ou si un jeton vient d'être supprimé de cette place. Cependant, cela n'influence pas le comportement de notre modèle puisque nous considérons des marquages *saufs*, et que les transitions en aval de la place  $p_i$  ne peuvent être tirées du fait de l'équation (8).

### 3.4.5 Modèle de programmation mathématique

Nous donnons dans la figure 3 le modèle mathématique complet correspondant au problème d'accessibilité dans les réseaux de Petri temporels défini en section 2.4.3. Les équations sont présentées dans l'ordre où elles ont été développées plus haut.

## 4 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons proposé un modèle de programmation mathématique en nombres entiers pour la résolution du problème d'accessibilité dans les réseaux temporels. L'approche présentée ici se fonde sur des hypothèses simplificatrices (réseau non pondéré, comportements à marquages *saufs*, intervalles finis) qui restent compatibles avec son utilisation dans un contexte de recherche opérationnelle.

De nombreuses perspectives pour de futurs travaux sont à envisager. Parmi celles-ci, nous livrons celles qui nous paraissent les plus intéressantes et donnons quelques pistes de recherche ci-dessous.

- La réalisation d'expériences numériques est en cours, à l'aide de la librairie d'optimisation Cplex. Les résultats numériques seront comparés aux outils Tina et Romeo de manière à comparer l'efficacité respective des différentes approches. Si le besoin s'en fait sentir, le développement d'heuristiques dédiées pourra alors être envisagé ;
- L'élargissement de notre modèle aux réseaux de Petri non pondérés, à bornes infinies, le traitement des cas de multi-sensibilisation dans le cas de réseaux à marquages non *saufs* permettrait de rendre l'approche aussi générale que possible. Dans ce cadre, un modèle fondé sur une structure de données associée aux transitions est à envisager.

## Références

- Berthomieu, B. & Diaz, M. (1991). Modeling and verification of time dependent systems using time petri nets, IEEE Trans. on Software Eng. **17**(3) : 259–273.
- Bourdeaud'huy, T. (2004). Techniques d'Abstraction pour l'Analyse et la Synthèse de Réseaux de Petri, PhD thesis, Ecole Centrale de Lille.
- Bourdeaud'huy, T., Hanafi, S. & Yim, P. (2006). Scheduling of flexible manufacturing systems using timed petri nets and mathematical programming, WODES'06, Workshop on Discrete Event Systems – Post-publication note at <http://www3.ec-lille.fr/tomnab/wodes06>, Ann Arbor, Michigan, USA.
- Diaz, D. (2001). Les réseaux de Petri - Modèles fondamentaux, Hermes Science, Traité IC2.
- Merlin, P. (1974). A Study of the Recoverability of Computing Systems, PhD thesis, Irvine.
- Roux, J. (1985). Modélisation et Analyse des Systèmes Distribués par les Réseaux de Petri Temporels, PhD thesis, Institut National des Sciences Appliquées de Toulouse.

Soit un réseau de Petri temporel défini par un ensemble de places  $\mathbb{P} = \{p_1, \dots, p_M\}$ , un ensemble de transitions  $\mathbb{T} = \{t_1, \dots, t_N\}$ , des matrices d'incidence  $C^-, C^+ \in \{0, 1\}^{M \times N}$ , des vecteurs  $\overrightarrow{S_{\min}}, \overrightarrow{S_{\max}} \in \mathbb{Q}^{+*}$  correspondant aux bornes des intervalles statiques des transitions, et un marquage initial  $\overrightarrow{m_0} \in \{0, 1\}^M$ . Soit  $B$  un entier suffisamment grand.

Soit  $\overrightarrow{m_f} \in \{0, 1\}^M$  un marquage cible, et  $K$  un entier strictement positif. Le modèle de programmation linéaire en nombre entiers  $IP(K, m_f)$  est défini par :

$$\text{Minimiser } \sum_{i=1}^K \Delta v_i$$

sous les contraintes :

$$\begin{array}{llll} \forall i \in [1, M], & & m_{0i} & = & \overrightarrow{m_0}(i) \\ \forall i \in [1, M], & & a_{0i} & = & 0 \\ \forall j \in [1, N], & & \varepsilon_{0j} & = & 0 \\ \forall i \in [1, M], & & m_{Ki} & = & \overrightarrow{m_f}(i) \\ \\ \forall k \in [0, K-1], & \forall i \in [1, M], & m_{ki} - \sum_{j=1}^N C_{ij}^- \cdot \sigma_{k+1,j} & \leq & 0 \\ \\ \forall k \in [1, K-1], & \forall j \in [1, N], & \forall p_i \in \bullet t_j, & \varepsilon_{kj} - a_{ki} & \leq 0 \\ \forall k \in [1, K-1], & \forall j \in [1, N], & \forall p_i \in \bullet t_j, & a_{ki} - \varepsilon_{kj} + B \cdot \mu_{kji} & \leq B \\ \forall k \in [1, K-1], & \forall j \in [1, N], & & \sum_{p_i \in \bullet t_j} \mu_{kji} & = 1 \\ \forall k \in [0, K-1], & \forall j \in [1, N], & \Delta v_{k+1} + \varepsilon_{kj} - B \cdot \sigma_{k+1,j} & \geq & \overrightarrow{S_{\min}}(j) - B \\ \forall k \in [0, K-1], & \forall j \in [1, N], & \Delta v_{k+1} + \varepsilon_{kj} + B \cdot \sigma_{k+1,j} & \leq & \overrightarrow{S_{\max}}(j) + B \\ \\ \forall k \in [0, K-1], & \forall i \in [1, M], & m_{k+1,i} - m_{ki} - \sum_{j=1}^N C_{ij}^- \cdot \sigma_{k+1,j} & = & 0 \\ \\ \forall k \in [0, K-2], & \forall i \in [1, M], & a_{k+1,i} - a_{ki} - \Delta v_{k+1} & \leq & 0 \\ \forall k \in [0, K-2], & \forall i \in [1, M], & a_{k+1,i} + B \cdot \sum_{j=1}^N C_{ij}^+ \cdot \sigma_{k+1,j} & \leq & B \\ \forall k \in [0, K-2], & \forall i \in [1, M], & a_{k+1,i} - a_{ki} - \Delta v_{k+1} + B \cdot \sum_{j=1}^N C_{ij}^+ \cdot \sigma_{k+1,j} & \geq & 0 \\ \\ \forall k \in [1, K-1], & \forall j \in [1, N], & \forall p_i \in \bullet t_j, & \mu_{kji} & \in \{0, 1\} \\ \forall k \in [0, K-1], & \forall j \in [1, N], & & \varepsilon_{kj} & \in \mathbb{Q}^+ \\ \forall k \in [0, K-1], & \forall i \in [1, M], & & a_{ki} & \in \mathbb{Q}^+ \\ \forall k \in [0, K], & \forall i \in [1, M], & & m_{ki} & \in \{0, 1\} \\ \forall k \in [1, K], & \forall j \in [1, N], & & \sigma_{kj} & \in \{0, 1\} \\ \forall k \in [1, K], & & & \Delta v_k & \in \mathbb{Q}^+ \end{array}$$

FIG. 3 – Modèle de Programmation mathématique en nombres entiers