



**HAL**  
open science

# Modélisation, analyse de performance et commande des systèmes à événements discrets

Abdelhak Guezzi

► **To cite this version:**

Abdelhak Guezzi. Modélisation, analyse de performance et commande des systèmes à événements discrets. Automatique. Université d'Angers, 2010. Français. NNT : . tel-00730500

**HAL Id: tel-00730500**

**<https://theses.hal.science/tel-00730500>**

Submitted on 10 Sep 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modélisation, analyse de performance et commande des systèmes à événements discrets.

THÈSE DE DOCTORAT

ÉCOLE DOCTORALE STIM  
SCIENCES ET TECHNOLOGIES DE L'INFORMATION  
ET DE MATHÉMATIQUE  
Spécialité : Automatique et Informatique Appliquée

Présentée et soutenue

le : 24 septembre 2010

à : Angers

par : **Abdelhak GUEZZI**

Devant le jury ci-dessous :

<i>Rapporteurs :</i>	<b>Alessandro GIUA</b> , Professeur	- Univ. of Cagliari, Italy
	<b>Hassane ALLA</b> , Professeur	- INPG Grenoble
<i>Examineurs :</i>	<b>Simon COLLART-DUTILLEUL</b> , Maître de Conférences	- École centrale de Lille
	<b>Mohammed EL BAGDOURI</b> , Professeur	- Univ. de Technologie de Belfort Montbéliard
	<b>Stephane GAUBERT</b> , Directeur de recherche INRIA	- Saclay -île de France
	<b>Jean-Claude HENNET</b> , Directeur de Recherche CNRS	- LSIS Marseille
	<b>Jean Louis BOIMOND</b> , Professeur	- Université d'Angers
	<b>Philippe DECLERCK</b> , Maître de Conférences	- Université d'Angers

Directeurs de thèse : **Jean Louis BOIMOND** et **Philippe DECLERCK**



# Remerciements

Les travaux de thèse présentés dans ce mémoire ont été réalisés au sein du Laboratoire d'Ingénierie des Systèmes Automatisés (LISA) de l'Université d'Angers. Je remercie ses directeurs Mr J.-L. Ferrier et Mr J.-L. Boimond qui lui a succédé de m'avoir accueillie dans ce laboratoire si dynamique.

Je suis particulièrement reconnaissant à Messieurs les Professeurs : A. Guia, Professeur à l'Université de Cagliari et H. Alla, Professeur à l'INPG de Grenoble, pour l'honneur qu'ils m'ont fait en acceptant de rapporter mon travail.

Je voudrais également remercier messieurs : S. Collart-Dutilleul, Maître de conférence à l'école centrale de Lille, Mohammed El Bagdouri, Professeur à l'université de Technologie de Belfort Montbéliard, Stephane Gaubert, Directeur de recherche INRIA et, J.-C. Hennet, Directeur de Recherche CNRS à LSIS Marseille, de l'intérêt qu'ils ont porté à mes travaux en acceptant de les examiner.

Je tiens à exprimer particulièrement ma gratitude envers P. Declerck, co-directeur de thèse, pour sa disponibilité, son intuition scientifique et ses nombreux conseils. La confiance qu'il m'a accordée, la pédagogie dont il a fait preuve à mon égard depuis qu'il m'a offert l'opportunité d'effectuer cette thèse de doctorat sous sa responsabilité, m'a toujours beaucoup honoré.

J'adresse également mes plus vifs remerciements à J.-L. Boimond, directeur de thèse, pour l'intérêt qu'il a porté à mes travaux et ses fréquents encouragements ont été des atouts majeurs dans l'efficacité et l'entrain de ce travail.

J'exprime aussi toute ma sympathie et mes remerciements pour tous les membres du LISA, en particulier, tous les doctorants du bureau E37 qui ont su créer un climat convivial.

Je remercie très chaleureusement mes parents et mes frères pour le soutien considérable qu'ils m'ont apporté et la confiance dont ils m'ont toujours témoigné.



# Table des matières

<b>1 Outils algébriques</b>	<b>7</b>
1.1 Introduction . . . . .	7
1.2 Théorie des treillis . . . . .	7
1.2.1 Ensembles ordonnés . . . . .	8
1.2.2 Notions de base sur les treillis . . . . .	10
1.3 Dioïdes et structures ordonnées . . . . .	12
1.3.1 Notions de base sur les dioïdes . . . . .	12
1.3.2 Dioïdes de matrices . . . . .	14
1.4 Programmation linéaire . . . . .	15
1.4.1 Systèmes d'inégalités linéaires . . . . .	15
1.4.2 Problème de la programmation linéaire selon une fonction objective . . . . .	17
1.4.3 Dualité . . . . .	17
1.4.4 Systèmes d'inégalités linéaires monotones et élément extremum . . . . .	19
1.4.4.1 Définitions . . . . .	19
1.4.4.2 Existence d'une solution extrême . . . . .	21
1.4.5 Bref rappels sur la théorie de la complexité . . . . .	22
1.4.6 Algorithmes . . . . .	23
1.4.6.1 Méthodes génériques . . . . .	23
1.4.6.2 Méthodes graphiques . . . . .	24
1.5 Conclusion . . . . .	25
<b>2 Réseaux de Petri temporisés et temporels, et leurs modèles algébriques</b>	<b>27</b>
2.1 Introduction . . . . .	27
2.2 Les réseaux de Petri (RdPs) . . . . .	28
2.2.1 Rappels et notions de base . . . . .	28
2.2.2 Quelques propriétés des RdPs . . . . .	31
2.2.3 Sous classes des RdPs . . . . .	32
2.3 Graphe d'événements temporisés . . . . .	35

2.3.1	Temporisation et fonctionnement d'un graphe d'événements . . . . .	35
2.3.2	Compteurs . . . . .	36
2.3.3	Dateurs . . . . .	37
2.4	Modèles algébriques des graphes d'événements temporisés . . . . .	37
2.4.1	Premières inégalités en dateur d'un GET . . . . .	37
2.4.2	Premières inégalités en compteur d'un GET . . . . .	39
2.4.3	Modèle d'un GET dans l'algèbre (Max,+) . . . . .	39
2.4.4	Modèle de type dateur d'un GET dans l'algèbre conventionnelle . . . . .	40
2.4.5	Modèle de GET dans l'algèbre (Min,+) . . . . .	43
2.4.6	Modèle de type compteur de GET dans l'algèbre conventionnelle . . . . .	43
2.5	Graphes d'événements temporels . . . . .	44
2.5.1	Graphes d'événements P-temporels . . . . .	44
2.5.2	Premières inéquations d'un graphe d'événements P-temporel . . . . .	45
2.5.3	Modèles de graphe d'événements P-temporels . . . . .	46
2.5.3.1	Modèle de GE P-temporels dans l'algèbre (Max,+) . . . . .	46
2.5.3.2	Modèle de type dateur de GE P-temporels dans l'algèbre conventionnelle . . . . .	48
2.5.4	Graphes d'événements T-temporels . . . . .	50
2.5.4.1	Modèle des GE T-temporels dans l'algèbre (Max,+) . . . . .	52
2.5.4.2	Modèle relaxé de type dateur des GE T-temporels dans l'algèbre conventionnelle . . . . .	53
2.6	Synthèse et comparative des modèles des graphes d'événements temporisés et temporels . . . . .	55
2.7	Conclusion . . . . .	57
<b>3</b>	<b>Taux de production</b> . . . . .	<b>59</b>
3.1	Introduction . . . . .	59
3.2	Propriétés spectrales des matrices de $\mathbb{R}_{max}^{n \times n}$ . . . . .	60
3.3	Comportement périodique d'un graphe d'événements temporisé . . . . .	62
3.4	Evaluation de performances des graphes d'événements dans l'algèbre standard . . . . .	63
3.5	Approche proposée pour l'évaluation de performances des GE P-temporels . . . . .	65
3.5.1	Condition d'existence du temps de cycle dans un GE P-temporel . . . . .	66
3.5.2	Le calcul du temps de cycle d'un graphe d'événements P-temporel . . . . .	69
3.5.2.1	La forme primale . . . . .	69
3.5.2.2	Technique de résolution . . . . .	71
3.5.2.3	La forme duale . . . . .	72
3.5.2.4	La forme duale simplifiée . . . . .	74
3.6	Illustration . . . . .	76

3.7	Conclusion . . . . .	79
<b>4</b>	<b>Standardisation et trajectoire au plus tôt</b>	<b>81</b>
4.1	Introduction . . . . .	81
4.2	Représentation d'état dans $\mathbb{R}_{max}$ . . . . .	81
4.3	Le modèle entrées/sorties pour un graphe d'événements temporisé . . . . .	82
4.4	Modèle entrées/sorties standardisé pour un GET . . . . .	83
4.4.1	Technique en utilisant la programmation linéaire . . . . .	85
4.4.2	Technique duale . . . . .	86
4.4.3	Lien avec la théorie des graphes . . . . .	87
4.4.4	Illustration . . . . .	88
4.5	Trajectoire au plus tôt d'un GET : . . . . .	90
4.5.1	La technique algébrique . . . . .	91
4.5.2	Technique utilisant la fonction objectif . . . . .	93
4.6	Conclusion . . . . .	95
<b>5</b>	<b>Commande des systèmes à événements discrets</b>	<b>97</b>
5.1	Introduction . . . . .	97
5.2	Bref état de l'art de la commande des systèmes à événements discrets . . . . .	97
5.2.1	Commande avec correcteur . . . . .	98
5.2.1.1	Commande avec spécifications . . . . .	98
5.2.1.2	Commande avec retour d'information . . . . .	98
5.2.2	Commande prédictive . . . . .	99
5.2.2.1	La poursuite de trajectoire désirée . . . . .	99
5.2.2.2	Commande avec retour d'information . . . . .	101
5.3	Commande au plus tard d'un GET . . . . .	101
5.3.1	Propriétés des GET . . . . .	101
5.3.2	La technique algébrique . . . . .	103
5.3.3	Technique utilisant la fonction objectif . . . . .	105
5.4	Commande à horizon glissant . . . . .	106
5.4.1	Principe . . . . .	106
5.4.2	Modèle de prédiction . . . . .	107
5.4.3	Fonction objectif . . . . .	107
5.5	Commande d'un GET sur un horizon glissant . . . . .	108
5.5.1	Horizon glissant . . . . .	108
5.5.2	Deux situations importantes . . . . .	108
5.5.3	Approche utilisant relation d'ordre composante par composante . . . . .	109
5.6	Conclusion . . . . .	111

<b>Annexes</b>	<b>116</b>
<b>A La résiduation</b>	<b>119</b>
A.1 Éléments de la théorie de la résiduation . . . . .	119
A.1.1 Homomorphismes de dioïdes, isotonie et continuité . . . . .	119
A.1.2 Applications résiduables et dualement résiduables . . . . .	121
A.1.2.1 Résiduation de l'addition et de la multiplication . . . . .	123
A.1.2.2 Résiduation dans le cas matriciel . . . . .	125
<b>B Théorie des graphes et algorithme du Simplexe</b>	<b>127</b>
B.1 Rappel de la théorie des graphes . . . . .	127
B.2 Algorithme du Simplexe . . . . .	128
B.2.1 Définition - Notations . . . . .	128
B.2.1.1 Forme standard . . . . .	129
B.2.1.2 Mise sous forme standard . . . . .	129
B.2.2 Définitions . . . . .	129
B.2.3 Résultats fondamentaux . . . . .	130
B.2.3.1 Définitions . . . . .	130
B.2.3.2 Théorèmes . . . . .	130
B.2.4 Principe . . . . .	131
B.2.5 Recherche d'un sommet de départ . . . . .	131
B.2.6 Passage d'un sommet à un autre . . . . .	131
B.2.6.1 Choix de la variable entrante . . . . .	132
B.2.6.2 Choix de la variable sortante . . . . .	132
B.2.6.3 Calcul du nouveau sommet . . . . .	132
B.2.6.4 Test d'optimalité . . . . .	132
<b>Bibliographie</b>	<b>133</b>

# Notations

SED : système à événements discret.

RdP : réseau de Petri.

GE : graphe d'événements.

GET : graphe d'événements temporisé.

$P$  : ensemble non vide de places d'un réseau de Petri.

$Tr$  : ensemble non vide de transitions d'un réseau de Petri.

$m_l$  : nombre de jetons initialement contenus dans la place  $p_l$ .

$T$  : vecteur colonne des durées de séjour d'un jeton dans les places.

$T^+$  : vecteur colonne des durées minimales de séjour d'un jeton dans les places.

$T^+$  : vecteur colonne des durées maximales de séjour d'un jeton dans les places.

$W$  : matrice d'incidence d'un réseau de Petri.

$W^+$  : matrice d'incidence arrière d'un réseau de Petri.

$W^-$  : matrice d'incidence avant d'un réseau de Petri.

$w_{ij}^-$  : poids de l'arc allant de la place  $p_i$  vers la transition  $x_j$ .

$w_{ij}^+$  : poids de l'arc allant de la place  $x_j$  vers la transition  $p_i$ .

$\bullet x_i$  : ensemble des places situées en amont de la transition  $x_i$ .

$x_i^\bullet$  : ensemble des places situées en aval de la transition  $x_i$ .

$\bullet p_i$  : ensemble des transitions situées en amont de la place  $p_i$ .

$p_i^\bullet$  : ensemble des transitions situées en aval de la place  $p_i$ .

$\leftarrow x_i$  : l'ensemble des transitions situées en amont de la transition  $x_i$ .

$x_i \rightarrow$  : l'ensemble des transitions situées en amont de la transition  $x_i$ .

$\theta$  : vecteur P-invariant associé au graphe d'événements temporisé.

$\mathbb{R}_{max}$  : algèbre  $(\mathbb{R} \cup \{+\infty\}, \max, +)$ .

$\overline{\mathbb{R}}_{max}$  : algèbre  $(\mathbb{R} \cup \{\pm\infty\}, \max, +)$ .

$\mathbb{Z}_{min}$  : algèbre  $(\mathbb{Z} \cup \{+\infty\}, \min, +)$ .

$\overline{\mathbb{Z}}_{min}$  : algèbre  $(\mathbb{Z} \cup \{\pm\infty\}, \min, +)$ .

$\oplus$  : opérateur maximum dans l'algèbre  $(\max, +)$ .

$\otimes$  : addition dans l'algèbre  $(\max, +)$ .

$\wedge$  : opérateur minimum dans l'algèbre  $(\min, +)$ .

$\otimes$  : addition dans l'algèbre  $(\min, +)$ .

$x_i(k)$  : variable dateur associée à la transition  $x_i$  d'un graphe d'événements temporisé.

$x_i(t)$  : variable compteur associée à la transition  $x_i$  d'un graphe d'événements temporisé.

$\lambda$  : temps de cycle.

$A^t$  : matrice transposée de la matrice  $A$ .

$I$  : matrice identité.

# Introduction

L'automatique classique considère des systèmes dynamiques où le changement d'état dépend du temps, en continu. Le comportement des systèmes dynamiques en automatique classique est souvent modélisé par des équations, différentielles ou aux différences. En revanche, les Systèmes Dynamiques à Événements Discrets (SDED) correspondent à des processus dont l'état change à l'occurrence d'un événement. Cela recouvre un grand nombre de processus comme les ateliers flexibles, les réseaux de transport, les systèmes informatiques, les systèmes multimédias,  $\dots$ . Une troisième catégorie correspond aux modèles hybrides qui fusionnent ces deux types de phénomènes comme l'indique son intitulé. Ce mémoire considère ainsi les Systèmes Dynamiques à Événements Discrets qui sont l'objet d'attention et d'études de nombreux chercheurs depuis plusieurs décennies, ceci à l'échelle internationale. Parmi les différents formalismes, les réseaux de Petri jouent un rôle important, présentant la caractéristique d'être un outil à la fois graphique et mathématique de modélisation. L'emploi de ces graphes pour spécifier et analyser ces systèmes, s'est généralisé ces dernières années, car ils permettent de décrire des phénomènes d'assemblage, de synchronisation, de partage de ressources,  $\dots$ .

Indépendamment du domaine des réseaux de Petri, une théorie a considéré les structures des semi-anneaux au début des années 80. Les livres de M. Gondran et M. Minoux constituent une bonne description de ceux-ci. La structure la plus connue est celle des semi-anneaux idempotents également appelés dioïdes qui permettent de décrire et d'analyser différents problèmes traités par la théorie des graphes de manière unifiée. Cette théorie s'est appuyée également sur la notion d'ordre et de treillis qui permettent de comparer les solutions. Appliquée aux graphes d'événements temporisés (GET), l'évolution de l'état est représentée sous la forme d'équations de récurrence qui utilisent des opérateurs  $(\max, +)$  ou alors  $(\min, +)$  et qui sont donc non-linéaires. Mais traduite dans cette structure de dioïde, cette représentation devient linéaire. La représentation est alors dite  $(\max, +)$ -linéaire, dans le cas du dioïde  $(\max, +)$ . L'algèbre des dioïdes est donc apparue comme une structure mathématique adéquate pour modéliser et étudier cette classe de systèmes.

Parallèlement à ces travaux et depuis les années 50, d'autres chercheurs en mathématiques se placent dans l'algèbre conventionnelle tout en exploitant la notion de treillis. Un pionnier est G.B.

Dantzig qui analysa les systèmes dynamiques Leontief vers 1955. Un autre auteur R.W. Cottle a montré une correspondance entre les inégalités et les treillis. Un travail récent est celui de M. Queyranne. En informatique, différents auteurs (B. Aspvall et Y. Shiloach, D. S. Hochbaum, E. Cohen et N. Meggido, A. Shostak, ... ) ont développé des algorithmes performants pouvant traiter des systèmes plus généraux que ceux traitant une matrice d'incidence. Il est à noter que ces travaux n'exploitent ni les travaux des réseaux de Petri, ni ceux des dioïdes. A l'inverse, les communautés réseaux de Petri et dioïdes semblent peu connaître ces travaux malgré l'apport qu'ils pourraient présenter par exemple sur un point de vue algorithmique dans la détermination de trajectoires temporelles dans les réseaux de Petri.

Un objectif de ce mémoire est de créer des liens entre ces différents travaux et d'essayer de montrer, au moins d'une manière partielle, la cohérence de ces travaux qui pourront ainsi se fortifier mutuellement. Nous présentons ainsi une alternative à l'algèbre des dioïdes pour modéliser les graphes d'événements temporisés et les graphes d'événements P-temporels au moyen d'un système d'inégalités dans l'algèbre conventionnelle. Exploitant ce modèle, nous montrons que nous pouvons analyser les performances du système et calculer sa commande. La technique sera d'associer à ces graphes un problème de programmation linéaire et de calculer la commande en utilisant des concepts de systèmes monotones. Un autre objectif plus général est le développement d'une théorie des graphes, complètement définie dans l'algèbre standard. L'objectif est une meilleure compréhension de ces théories, la technique proposée étant accessible à des non-spécialistes en théorie des graphes. Un autre avantage est la possibilité d'appliquer des algorithmes de la programmation linéaire, comme le Simplexe (Annexe B.2). Polyvalent, il peut être appliqué dans de nombreux domaines comme l'automatique classique, ou les systèmes à événements discrets.

Les lignes directrices de ce mémoire, ainsi que son organisation sont présentés ci-dessous. Ce mémoire comporte cinq chapitres :

- Dans le premier chapitre, nous exposons un ensemble de concepts et d'outils mathématiques utiles pour le traitement des problèmes posés dans ce mémoire. Dans une première partie, on présente les concepts et les fondements de base des treillis et des dioïdes. Ensuite, nous décrivons des propriétés de la programmation linéaire et la classe des systèmes linéaires monotones, objets centraux de cette étude. Elle représente en fait un outil important pour la suite de ce mémoire. Dans la dernière partie, nous présentons un bref état de l'art des algorithmes de résolution en programmation linéaire (génériques et graphiques).
- Le deuxième chapitre propose un survol de la modélisation de différents graphes d'événements temporisés et temporels. La première partie de chapitre est essentiellement biblio-

graphique et regroupe des résultats bien connus sur les réseaux de Petri. Quelques résultats classiques sur la modélisation des graphes d'événements temporisés et P-temporels dans l'algèbre des dioïdes, sont également exposés. Dans la deuxième partie, à partir des inégalités en dateur, nous donnons une représentation d'état des graphes d'événements temporisés, P-temporels et T-temporels sous la forme d'un système d'inégalités  $A.x \leq b$  dans l'algèbre conventionnelle. Enfin nous proposons une synthèse et un comparatif des modèles des graphes temporisés et temporels.

- Le troisième chapitre est consacré à l'analyse de performances des graphes d'événements P-temporels. Nous utilisons le modèle de la forme polyédral  $A.x \leq b$  pour évaluer et analyser les performances des graphes d'événements P-temporels. Pour cela, considérant un comportement 1-périodique, l'application d'un lemme de Farkas permettra la détermination des bornes encadrant le temps de cycle. Ensuite, ce dernier est calculée par deux techniques : une première approche sera nommée primale puisque elle se base sur l'écriture primale de la programmation linéaire, et une deuxième approche sera nommée duale. Cette dernière sera obtenue en utilisant le théorème de la dualité, bien connue en programmation linéaire, ainsi que le théorème de Stiemke.
- Dans le quatrième chapitre, l'objectif est de compléter le modèle initial  $A.x \leq b$  en introduisant des entrées et des sorties. Nous présentons une méthode de standardisation dont l'objectif est d'avoir une écriture proche de l'équation d'état de l'automatique classique. Ensuite les concepts d'ordre composante par composante, de demi-treillis et d'inégalités inf-monotones permettent de prouver l'existence d'une trajectoire unique minimum qui représente la trajectoire au plus tôt d'un graphe d'événements temporisé.
- Dans la première partie du chapitre cinq, on réalise la synthèse de la commande sous le critère classique de juste-à-temps : la commande doit retarder, autant que possible le système, de sorte que la sortie se produise avant une sortie désirée définie sur un horizon fini donné. Naturellement, des techniques utilisant l'algèbre  $(\max, +)$  ont déjà été proposées. Dans ce chapitre, on montre qu'une approche peut également être réalisée dans l'algèbre standard. Pour cela, on utilise une classe particulière d'inégalités linéaires (inégalités sup-monotones) et la programmation linéaire. Dans la deuxième partie, nous considérons une commande effectuant une poursuite de trajectoire sur un horizon glissant. Cette approche s'appuie sur les deux approches proposées pour le calcul de la trajectoire au plus tôt et au plus tard.



# Chapitre 1

## Outils algébriques

### 1.1 Introduction

Dans ce chapitre, nous allons rassembler un certain nombre de notions utiles à la compréhension de la théorie de base des treillis, des dioïdes et des systèmes linéaires. Dans un premier temps, nous introduisons les notions de base relatives à la théorie des treillis. Ensuite, nous introduisons une structure algébrique particulière, les dioïdes, également appelés, semi-anneaux idempotents. Un dioïde peut également être considéré comme un sup-demi-treillis.

Nous allons introduire ensuite quelques définitions et propriétés de la théorie polyédrale et de la programmation linéaire, utiles dans les prochaines sections de ce mémoire. Pour plus de détails, on pourra se référer à Pulleyblank et Schrijver [Schrijver 1987]. Ensuite, nous allons considérer une classe particulière de systèmes linéaires, les systèmes monotones, pour lesquels on observe une conjonction entre propriétés syntaxiques et propriétés géométriques. Nous montrons que, dans l'espace produit muni de l'ordre  $\geq$ , les systèmes monotones décrivent des demi-treillis (continus ou discrets). Cette propriété permet de garantir l'existence d'un plus petit (ou plus grand) élément lorsque l'ensemble des solutions est non vide et borné (c'est-à-dire, minoré ou majoré). Réciproquement, d'après un résultat de Veinott [Veinott 1968], tout polyèdre de  $\mathbb{R}^n$  présentant cette particularité géométrique peut être défini par un système monotone. Nous identifions également une sous-classe remarquable de systèmes monotones, à savoir les systèmes bimonotones [Veinott 1968], [Hochbaum 1993]. Nous présentons enfin des algorithmes pour la programmation linéaire et pour la résolution des systèmes monotones.

### 1.2 Théorie des treillis

L'association d'une relation d'ordre avec un ensemble (ou sous ensemble) définit un ensemble ordonné qui, à son tour, introduit un treillis. Nous rappelons brièvement dans cette partie un ensemble de notions, de définitions et de propriétés sur les ensembles ordonnés et les treillis. Le lecteur intéressé par les structures algébriques pourra consulter en particulier [Baccelli 1992],

[Kumar 1995], [Blyth 1972], [Gondran 2001] et [Davey 2002].

### 1.2.1 Ensembles ordonnés

On rappelle qu'une relation d'ordre sur un ensemble  $E$ , notée  $\leq$ , est une relation binaire :

- réflexive  $\forall x \in E : x \leq x$ ,
- antisymétrique  $\forall x, y \in E : x \leq y \text{ et } y \leq x \implies x = y$ ,
- transitive  $\forall x, y, z \in E : x \leq y \text{ et } y \leq z \implies x \leq z$ .

Un ensemble  $E$  est ordonné, s'il est muni d'une relation d'ordre  $\leq$ . On le notera  $(E, \leq)$ .

On dit que deux éléments  $x$  et  $y$  dans  $E$  sont non-comparables, si aucune des relations suivantes  $x \leq y$  et  $y \leq x$  n'est vérifiée.

#### Définition 1.1. (*Ordre partiel*)

Soit un ensemble ordonné  $(E, \leq)$ . S'il existe deux éléments  $x, y \in E$  tels que  $x$  et  $y$  sont non-comparables, alors la relation d'ordre  $\leq$  est partielle. On dit aussi que  $(E, \leq)$  est partiellement ordonné.

**Exemple 1.2** Soit l'ensemble des entiers naturels  $\mathbb{N}^*$ . Considérons la relation d'ordre  $\leq$  définie comme suit :

$$x \leq y \text{ si et seulement si, } x \text{ divise } y.$$

Il est clair alors que  $(E, \leq)$  est partiel ( $3 \not\leq 5$  car 3 n'est pas divisible par 5 dans  $\mathbb{N}^*$ , et inversement).  $\diamond$

#### Définition 1.3. (*Ordre total*)

Soit  $(E, \leq)$ , si  $\forall x, y \in E$ , on a  $x \leq y$  ou  $y \leq x$ , alors la relation d'ordre  $\leq$  est totale.  $E$  est appelé un ensemble totalement ordonné.

**Exemple 1.4** Les deux ensembles  $(\mathbb{R}, \leq)$ ,  $(\mathbb{Z}, \leq)$ , où  $\leq$  est la relation d'ordre naturel, sont totalement ordonnés.  $\diamond$

Un ensemble  $(E, \leq)$  totalement ordonné est appelé aussi une chaîne.

Les ensembles totalement, ou partiellement, ordonnés sont souvent décrits graphiquement en utilisant un diagramme dit de Hasse. Ce dernier est un graphe acyclique direct tel qu'il existe un arc direct qui va de  $x$  vers  $y$  si et seulement si,  $x \leq y$  (dans ce cas,  $x$  et  $y$  sont comparables). Notons que les arcs triviaux  $x \mapsto x$  et les arcs se déduisant par transitivité sont omis. L'ordre de croissance est pris conventionnellement dans le sens du bas vers le haut.

**Exemple 1.5** Soit l'ensemble  $(\{x, y, z, t\}, \leq)$  partiellement ordonné avec son diagramme de Hasse montré dans la figure 1.1. Chaque élément est représenté par un point.

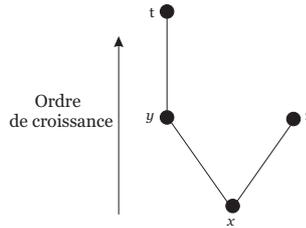


Fig. 1.1 : Diagramme de Hasse.

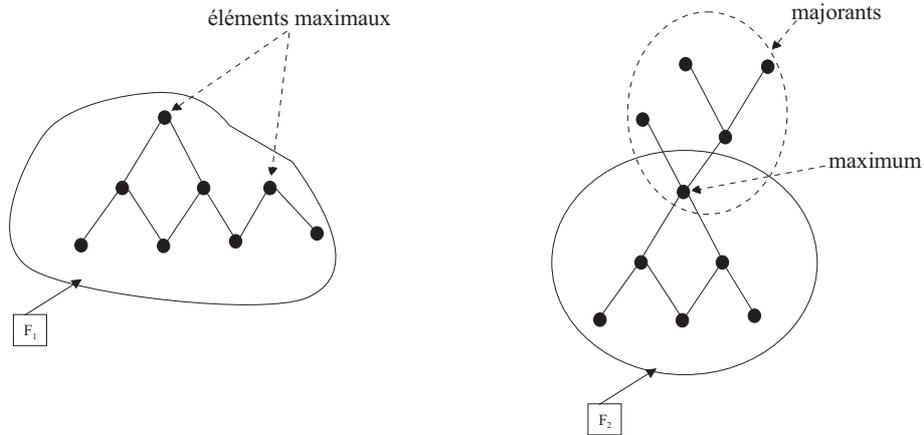
On a  $x \leq y$ ,  $x \leq z$  et  $y \leq t$ . On peut constater dans la figure 1.1 que  $y$  et  $z$ , ainsi que  $z$  et  $t$  sont non-comparables.  $\diamond$

Dans la définition qui suit, nous donnons un vocabulaire relatif aux éléments dans les ensembles ordonnés. Remarquons que ce vocabulaire n'est pas standard et peut être différent selon les auteurs.

**Définition 1.6. (Vocabulaire)**

Soit  $(E, \leq)$  un ensemble ordonné et  $F$  un sous-ensemble non vide de  $E$  ( $F \subseteq E$ ).

- Soit  $F \subseteq E$ , tout élément  $b \in F$  vérifiant :  $\nexists x \in F, x \neq b$  tel que  $b \leq x$  est appelé **élément maximal** de  $F$ . De même, on appelle **élément minimal** de  $F \subseteq E$  tout  $a \in F$  vérifiant :  $\nexists x \in F, x \neq a$  tel que  $x \leq a$ .
- Lorsqu'il existe un **élément maximal** (respectivement **minimal**), il n'est pas toujours unique.
- Un élément  $b \in E$  vérifiant  $\forall x \in F : x \leq b$  est appelé un **majorant**, ou **borne supérieure**. Si  $b \in F$ , alors  $b$  est unique et on l'appelle **maximum** de  $F$ .
- De même  $a \in E$  est un **minorant**, ou **borne inférieure**, de  $F$  si et seulement si,  $\forall x \in F : a \leq x$ . Si  $a \in F$ , alors  $a$  est appelé **minimum** de  $F$  et il est unique.
- Un sous ensemble  $F \subseteq E$  est dit borné s'il admet un majorant et un minorant.
- Lorsque l'ensemble des majorants, ou bornes supérieures, de  $F \subseteq E$  a un plus petit élément, ce dernier est appelé **plus petite borne supérieure** de  $F$ , et notée  $\sup(F)$ . De même, lorsque l'ensemble des minorants, ou bornes inférieures, de  $F$  a un plus grand élément, on l'appelle **plus grande borne inférieure** de  $F$ . On le note  $\inf(F)$ . On parle aussi respectivement de **suprimal** ou **infimal** de  $F$  pour  $\sup(F)$  et  $\inf(F)$ . Ces éléments sont uniques s'ils existent.
- L'ensemble  $(E, \leq)$  est dit complet si toute partie  $F \subseteq E$  admet une borne supérieure, notée  $\bigvee_{x \in F} x$ . De même, il est dit complet pour l'ordre dual si  $\inf(F)$  existe, elle est notée  $\bigwedge_{x \in F} x$ .



**Fig. 1.2 :** Représentation des éléments maximaux, maximums et majorants des ensembles  $F_1$  et  $F_2$ .

La figure 1.2 illustre les différentes définitions.

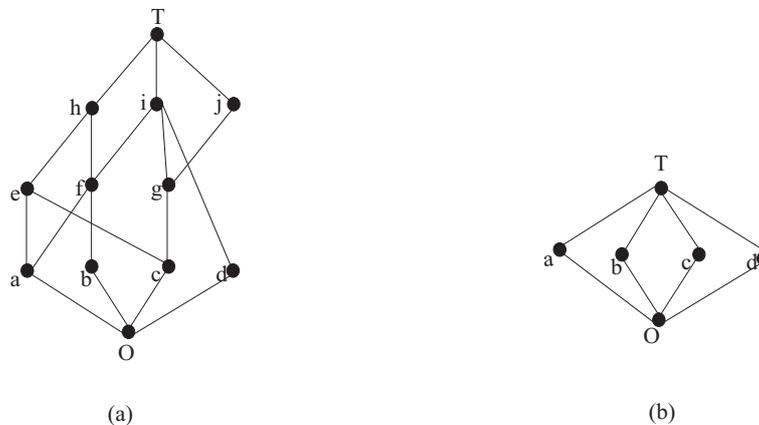
### 1.2.2 Notions de base sur les treillis

#### Définition 1.7. (*Demi-treillis*)

Un ensemble ordonné  $(E, \leq)$  est un **sup-demi-treillis** si toute partie de deux éléments  $x_1$  et  $x_2$  de  $E$  admet une plus petite borne supérieure (plus petit majorant)  $b$ . On notera  $b = x_1 \vee x_2$ . Duale,  $E$  est un **inf-demi-treillis** si toute partie  $\{x_1, x_2\}$  de  $E$  admet une plus grande borne inférieure (plus grand minorant)  $a = x_1 \wedge x_2$ .

#### Définition 1.8. (*Treillis*)

On dit que l'ensemble  $(E, \leq)$  est un **treillis**, si à la fois  $(E, \leq)$  est sup-demi-treillis et inf-demi-treillis.



**Fig. 1.3 :** Diagrammes de Hasse pour deux ensembles ordonnés différents [Davey 2002].

**Exemple 1.9** Soit l'ensemble  $E = \{O, a, b, c, d, e, f, g, h, i, j, T\}$  muni de la relation d'ordre  $\leq$  dont le diagramme de Hasse est montré sur la figure 1.3a. On est tenté au vue du diagramme, de dire que  $b \vee c = i$ . Or, l'ensemble des majorants de  $b$  et  $c$  est égal à  $\{T, h, i\}$ . Comme cet ensemble a deux éléments minimaux distincts,  $h$  et  $i$ , par conséquent  $b \vee c$  n'existe pas. Il en découle que l'ensemble  $E$  n'est pas un sup-demi-treillis, et donc, n'est pas un treillis.  $\diamond$

**Exemple 1.10** L'ensemble des intervalles de  $\mathbb{R}$  ordonné par l'inclusion a une structure de treillis, où la borne supérieure (respectivement, inférieure) est donnée par  $A \vee B = A \cup B$  (respectivement,  $A \wedge B = A \cap B$ ) avec l'ensemble vide comme plus petit sous-ensemble.  $\diamond$

**Définition 1.11. (Demi-treillis complet)**

Un *sup-demi-treillis complet* (respectivement, *inf-demi-treillis complet*) est un ensemble  $E$  ordonné tel qu'il existe une plus petite borne supérieure (respectivement, une plus grande borne inférieure) pour tout sous-ensemble fini, ou infini, de  $E$ .

**Définition 1.12. (Treillis complet)**

Un ensemble  $(E, \leq)$  est dit complet s'il est, à la fois sup-demi-treillis complet et inf-demi-treillis complet.

**Exemple 1.13** Soit l'ensemble ordonné dont le diagramme de Hasse est donné dans la figure 1.3b. Cet ensemble ordonné est un treillis complet.  $\diamond$

**Remarque 1.14**

Si  $E$  muni des deux opérations  $\vee$  et  $\wedge$  est un treillis complet, alors il existe un plus petit élément (en anglais appelé bottom), noté  $\varepsilon$ , et un plus grand élément (top), noté  $T$ . L'élément  $\varepsilon$  est neutre pour  $\vee$ , il est aussi absorbant pour l'opération  $\wedge$ ,  $\varepsilon \wedge x = \varepsilon, \forall x \in E$ . Dualement,  $T$  est neutre pour  $\wedge$ , il est absorbant pour  $\vee$ ,  $T \vee x = T, \forall x \in E$ .  $\diamond$

**Définition 1.15. (Treillis distributif)**

Un treillis  $E$  doté des deux opérations  $\vee$  et  $\wedge$  est distributif si  $\vee$  distribue par rapport à  $\wedge$ ,  $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$  et réciproquement,  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ .

Nous concluons cette partie par le résultat fondamental suivant :

**Théorème 1.16.** *Un sup-demi-treillis complet  $E$  est un treillis complet si et seulement si, il admet un élément minimum.*

**Preuve :** La démonstration est donnée par exemple dans [Menguy 1997].

□

### 1.3 Dioïdes et structures ordonnées

Dans cette partie, nous rappelons les fondements concernant les demi-anneaux idempotents, ou dioïdes, nous montrons aussi le lien entre ces derniers et les structures ordonnées. Les dioïdes apparaissent ainsi comme des demi-treillis. De plus, un dioïde complet a la structure d'un treillis complet.

Ce travail bibliographique a été fait à partir des documents suivants auxquels le lecteur pourra se référer : [Baccelli 1992], [Gaubert 1992], [Cuninghame-Green 1979], [Menguy 1997] et [Didi-Alaoui 2005].

#### 1.3.1 Notions de base sur les dioïdes

##### Définition 1.17. (*Monoïde*)

Un monoïde  $(E, \oplus)$  est un ensemble doté d'une loi de composition interne notée  $\oplus$ , associative, et possédant un élément neutre. Si cette loi est commutative, le monoïde est dit commutatif.

##### Définition 1.18. (*Monoïde idempotent*)

Un monoïde  $(E, \oplus)$  est dit idempotent si la loi  $\oplus$  est commutative, associative et idempotente, c'est-à-dire, si elle vérifie :

$$\forall a \in E, \quad a \oplus a = a.$$

##### Définition 1.19. (*Demi-anneau*)

Un demi-anneau est un ensemble  $\mathcal{D}$  muni de deux lois de compositions internes  $\oplus$  et  $\otimes$ , appelées respectivement somme et produit, telles que :

1.  $(\mathcal{D}, \oplus)$  est un monoïde commutatif avec un élément neutre (zéro), noté  $\varepsilon$ .
2.  $(\mathcal{D}, \otimes)$  est un monoïde dont l'élément neutre  $e$  est appelé identité<sup>1</sup>.
3. La loi  $\otimes$  est distributive, par rapport à la loi  $\oplus$ .  

$$\forall a, b, c \in \mathcal{D}, \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c),$$

$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c).$$
4. Le zéro est absorbant pour le produit :  $\forall a \in \mathcal{D}, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$ .

Si le produit  $\otimes$  est commutatif,  $(\mathcal{D}, \oplus, \otimes)$  est un demi-anneau commutatif.

##### Définition 1.20. (*Dioïde*)

Un dioïde  $(\mathcal{D}, \oplus, \otimes)$  est un demi-anneau tel que le monoïde  $(\mathcal{D}, \oplus)$  est idempotent.

Si la loi  $\otimes$  est commutative, alors le dioïde  $(\mathcal{D}, \oplus, \otimes)$  est commutatif.

##### Exemple 1.21

• L'ensemble  $\mathbb{R} \cup \{-\infty\}$ , muni de  $\oplus$  et  $\otimes$  correspondant respectivement à l'opération max et à l'addition usuelle, est un dioïde noté  $\mathbb{R}_{max}$ . Classiquement, ce dioïde est appelé algèbre (max,

1. Certains auteurs désignent  $e$  comme étant l'élément unité.

+) avec  $e = 0$  et  $\varepsilon = -\infty$ .

• De même, on définit l'algèbre  $(\min, +)$  comme le dioïde  $(\mathcal{R} \cup \{+\infty\}, \oplus, \otimes)$  où  $\oplus$  et  $\otimes$  coïncident respectivement avec l'opérateur  $\min$  et l'addition usuelle,  $e = 0$  et  $\varepsilon = +\infty$ . L'algèbre  $(\min, +)$  est aussi appelée algèbre tropicale.

Notons que les deux structures algébriques  $(\max, +)$  et  $(\min, +)$  sont commutatives.  $\diamond$

**Définition 1.22. (Plus grande borne inférieure)**

Soit  $\mathcal{D}$  un dioïde complet,  $a, b \in \mathcal{D}$ . La plus grande borne inférieure de  $a$  et  $b$  existe et est donnée par :

$$a \wedge b = \bigvee \{x \mid x \leq a \text{ et } x \leq b\}. \quad (1.1)$$

D'une manière analogue, la plus grande borne inférieure d'une partie  $X$  de  $\mathcal{D}$  est donnée par :

$$\bigwedge X = \bigvee \{y \in \mathcal{D} \mid \forall x \in X, y \leq x\}. \quad (1.2)$$

Remarquons que les ensembles à droite de 1.1 et 1.2 sont non vides puisque  $\mathcal{D}$  admet  $\varepsilon$  comme minimum.

L'opération  $\wedge$  est associative, commutative et idempotente et a pour l'élément neutre  $T$  ( $\forall a, T \wedge a = a$ ). Cette loi ( $\wedge$ ) fait du dioïde  $(\mathcal{D}, \oplus, \wedge)$  un treillis complet. Par conséquent, les équivalences ci-dessous sont vérifiées :

$$a \geq b \iff a = a \oplus b \iff b = a \wedge b. \quad (1.3)$$

En effet, le théorème 4.28 dans [Baccelli 1992] montre que tout couple d'éléments a une plus petite borne supérieure et ainsi que la relation d'idempotence de  $\mathcal{D}$  induit une structure de sup-demi-treillis sur  $\mathcal{D}$ . Le sup-demi-treillis relatif à un dioïde complet est également complet car tout sous-ensemble non vide admet une plus petite borne supérieure. Comme un dioïde a un plus petit élément  $\varepsilon$ , le théorème 1.16 peut être appliqué et ainsi, un dioïde complet est un treillis complet. Aussi, tous les résultats sur les treillis complets sont transposables sur les dioïdes complets.

**Définition 1.23. (Dioïde distributif)**

Un dioïde  $(\mathcal{D}, \oplus, \otimes)$  est distributif s'il est complet et si, pour tout sous-ensemble  $\mathcal{C} \in \mathcal{D}$  et  $\forall a \in \mathcal{D}$ , on a :

$$\left( \bigwedge_{x \in \mathcal{C}} x \right) \oplus a = \bigwedge_{x \in \mathcal{C}} (x \oplus a),$$

et

$$\left( \bigwedge_{x \in \mathcal{C}} x \right) \otimes a = \bigwedge_{x \in \mathcal{C}} (x \otimes a).$$

L'introduction de la borne inférieure pose la question du comportement mutuel entre  $\oplus$  et  $\wedge$ .

### 1.3.2 Dioïdes de matrices

Soit un dioïde  $\mathcal{D}$  dont les éléments sont des scalaires. Nous considérons l'ensemble des matrices carrées de dimension  $n \times n$  dont les éléments appartiennent à  $\mathcal{D}$ . On définit  $(\mathcal{D}^{n \times n}, \oplus, \otimes)$  comme un dioïde matriciel, avec l'élément neutre (zéro) pour  $\oplus$  sous la forme d'une matrice dont tous les coefficients sont égaux à  $\epsilon$ . Le produit  $\otimes$  a un élément neutre (identité), noté  $I_{\mathcal{D}}$ , composé de la manière suivante :

$$I_{\mathcal{D}(i,j)} = \begin{cases} e & \text{si } i = j, \\ \epsilon & \text{si } i \neq j. \end{cases}$$

Si  $\mathcal{D}$  est complet alors  $\mathcal{D}^{n \times n}$  l'est aussi.

#### Opérations sur les matrices

La somme et le produit des matrices de dimension appropriée, sont définis ainsi :

$$\begin{aligned} \forall A \in \mathcal{D}^{n \times n}, \forall B \in \mathcal{D}^{n \times n}, \quad (A \oplus B)_{ij} &= A_{ij} \oplus B_{ij}, \\ \forall A \in \mathcal{D}^{n \times n}, \forall B \in \mathcal{D}^{n \times n}, \quad (A \otimes B)_{ij} &= \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}. \end{aligned}$$

**Exemple 1.24** Nous donnons ici un exemple d'un produit et une somme de deux matrices carrées dans l'algèbre  $(\max, +)$ .

$$A = \begin{pmatrix} 3 & 7 \\ 4 & 5 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 6 \\ 2 & 8 \end{pmatrix},$$

$$A \oplus B = \begin{pmatrix} \max[3, 1] & \max[7, 6] \\ \max[4, 2] & \max[5, 8] \end{pmatrix} = \begin{pmatrix} 3 & 7 \\ 4 & 8 \end{pmatrix},$$

$$A \otimes B = \begin{pmatrix} \max[3+1, 7+2] & \max[3+6, 7+8] \\ \max[4+1, 5+2] & \max[4+6, 5+8] \end{pmatrix} = \begin{pmatrix} 9 & 15 \\ 7 & 13 \end{pmatrix}.$$

◇

On peut définir la multiplication d'une matrice  $A$  par une constante  $\alpha$ , comme suit :

$$(\alpha \otimes A)_{ij} = \alpha \otimes A_{ij} = \alpha + A_{ij}.$$

En général, la multiplication des matrices dans  $\mathcal{D}^{n \times n}$  n'est pas commutative, même si  $(\mathcal{D}, \oplus, \otimes)$  est commutative. Par contre,  $\mathcal{D}^{n \times n}$  est distributif si  $\mathcal{D}$  l'est. On a  $A \geq B$  dans  $\mathcal{D}^{n \times n}$ , où  $\geq$  est la relation d'ordre du dioïde  $\Leftrightarrow \{A_{ij} \geq B_{ij} \text{ dans } \mathcal{D}, \quad i = 1, \dots, n, \quad j = 1, \dots, n\}$ .

## 1.4 Programmation linéaire

Une des formes de programmation mathématique les plus étudiées est la programmation linéaire qui correspond au cas où  $f(x)$  est une fonction linéaire de  $n$  variables et où le domaine  $\mathcal{D}$  est déterminé par un ensemble fini d'inégalités linéaires.

### 1.4.1 Systèmes d'inégalités linéaires

Soit  $A$  une matrice réelle ayant  $m$  lignes et  $n$  colonnes et  $b \in \mathbb{R}^m$ . Nous noterons  $0_n$  le vecteur ayant toutes ses composantes nulles. Alors le système  $Ax \leq b$  est appelé un système d'inégalités linéaires et le système  $Ax = b$  est appelé un système d'équations linéaires. Soient  $a \in \mathbb{R}^n \setminus \{0_n\}$  et  $a_0 \in \mathbb{R}$ . On appelle demi-espace de  $\mathbb{R}^n$  l'ensemble des points  $\{x \in \mathbb{R}^n \mid a^t x \leq a_0\}$ .

Un polyèdre  $P$  est un ensemble de points de  $\mathbb{R}^n$  engendré par l'intersection d'un nombre fini de demi-espaces de  $\mathbb{R}^n$ . D'une manière équivalente,  $P$  est l'ensemble des solutions d'un système d'inégalités linéaires, c'est-à-dire :

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

où  $A$  est une matrice à  $m$  lignes et  $n$  colonnes, et  $b$  un vecteur à  $m$  composantes. Nous dirons alors que le système  $Ax \leq b$  détermine (définit ou caractérise) le polyèdre  $P$ .

**Définition 1.25.** L'objet  $\{\sum_{i=1}^{i=p} \lambda_i x_i \text{ avec } \lambda_i \geq 0, \forall i \in \{1, \dots, p\}\}$  est un cône.

**Définition 1.26.** Un cône  $C$  est dit polyédral s'il peut être décrit par un ensemble fini d'inégalités linéaires du type  $Ax \leq b$ , c'est-à-dire, s'il existe une matrice  $A$  telle que  $C = \{x \mid Ax \leq b\}$ .

Une inégalité (ou contrainte)  $a^t x \leq a_0$  est dite valide pour un polyèdre  $P$  si elle est vérifiée par tous les points de  $P$ .

**Théorème 1.27.** (Farkas, Minkowsky, Weyl)

$C$  est un cône polyédral si et seulement s'il existe une famille finie de vecteurs  $\{x_1^*, \dots, x_p^*\}$  telle que  $\forall x \in C$ ,  $x = \sum_{i=1}^{i=p} \lambda x_i^*$  avec  $\lambda_i \geq 0, \forall i \in \{1, \dots, p\}$ .

Ce théorème peut être vu comme une conséquence d'un théorème souvent appelé théorème fondamental des inégalités linéaires dû à J. Farkas et H. Minkowsky, puis repris par H. Weyl et C. Carathéodory.

Avant de déterminer la solution d'un système d'inégalités, il faut répondre à la question : est-ce-que cette région est non vide ?

On doit à J. Farkas [Farkas 1894] et [Farkas 1898], H. Minkowski [Minkowski 1896] et H. Weyl, [Weyl 1935] la caractérisation de telles régions en termes d'inégalités linéaires.

**Lemme 1.28 (Farkas)** Soit  $A$  une matrice et  $b$  un vecteur. Alors le système  $Ax = b, x \geq 0$  a une solution si, et seulement si,  $y^t b \geq 0$  pour tout vecteur  $y$  tel que  $y^t A \geq 0$ .  $\diamond$

**Preuve :**

La nécessité de la condition est immédiate du fait que  $y^t b = y^t Ax \geq 0$  pour tout vecteur  $x \geq 0$ , tout vecteur  $y$  tel que  $y^t A \geq 0$  et que par hypothèse  $Ax = b$ . Maintenant supposons qu'il n'existe pas de vecteur  $x \geq 0$  tel que  $Ax = b$  alors  $b$  n'appartient pas au cône généré par les vecteurs colonnes de la matrice  $A$ . ■

Maintenant nous allons voir comment savoir si une région de l'espace définie par un système d'inégalités linéaires est vide et dans le cas contraire, comment trouver au moins un point de cette région. Un des premiers à s'être intéressé aux inégalités linéaires est Fourier qui décrit en 1827 [Fourier 1827], une méthode permettant de trouver une solution d'un système d'inégalités linéaires s'il en existe une. Ses travaux seront repris par Motzkin en 1936 [Motzkin 1936], donnant lieu à la méthode de Fourier-Motzkin que nous exposons brièvement ci-dessous.

Soit  $A$  une matrice  $m \times n$  et  $b$  un vecteur de dimension  $m$ , considérons alors le système d'inégalités linéaires  $Ax \leq b$ . Nous désirons éliminer  $x_1$ . Multiplions chacune de ces inégalités par un scalaire positif de sorte que nous ayons dans la première colonne de  $A$  des coefficients appartenant à  $\{-1, 0, +1\}$ . Le système peut ainsi se réécrire de la manière suivante :

$$\begin{aligned} x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\leq b_i, & i \in \{1, \dots, m_1\}, \\ -x_1 + a_{j2}x_2 + \cdots + a_{jn}x_n &\leq b_j, & j \in \{m_1 + 1, \dots, m_2\}, \\ a_{k2}x_2 + \cdots + a_{kn}x_n &\leq b_k, & k \in \{m_2 + 1, \dots, m\}. \end{aligned}$$

Il en résulte que nous pouvons écrire que :

$$\text{Max}\left\{\sum_{l=2}^n a_{jl}x_l - b_j, j \in \{m_1 + 1, \dots, m_2\}\right\} \leq x_1 \leq \text{Min}\left\{b_i - \sum_{l=2}^n a_{il}x_l, i \in \{1, \dots, m_1\}\right\}. \quad (1.4)$$

Nous pouvons éliminer la variable  $x_1$  et réécrire un système réduit de la forme suivante :

$$\begin{aligned} \sum_{l=2}^n (a_{il} + a_{jl})x_l &\leq b_i + b_j & i \in \{1, \dots, m_1\}, j \in \{m_1 + 1, \dots, m_2\}, \\ \sum_{l=2}^n a_{kl}x_l &\leq b_k & k \in \{m_2 + 1, \dots, m\}. \end{aligned} \quad (1.5)$$

Le système ci-dessus ne possède plus que  $n - 1$  variables. Chacune de ses solutions peut ensuite être utilement complétée pour obtenir une solution du système original en choisissant pour valeur de  $x_1$  une valeur satisfaisant la relation  $\text{Min} - \text{Max}$  vue plus haut. Pour obtenir une solution, on itère jusqu'à l'obtention d'un système à une inconnue que l'on pourra résoudre.

**Lemme 1.29 (Farkas)[Kuhn 1956]** Le système d'inégalité linéaires  $Ax \leq b$  a une solution si, et seulement si,  $y^t b \geq 0$  pour tout vecteur  $y \geq 0$  tel que  $y^t A = 0$ .  $\diamond$

### 1.4.2 Problème de la programmation linéaire selon une fonction objective

Le problème primal ( $P$ ) se pose de la manière suivante :

$$cx^* = \text{Max} \{cx \mid Ax \leq b, \quad x \in \mathbb{R}^n\}. \quad (1.6)$$

La fonction  $cx : \mathbb{R}^n \rightarrow \mathbb{R}$  est appelée *fonction économique* (ou *objectif*), les inégalités dans le système  $Ax \leq b$  sont appelées contraintes et  $x^*$  est appelé solution optimale du programme linéaire ( $P$ ). Cette solution n'est pas nécessairement unique.

Pour chaque programme linéaire ( $P$ ), on associe un programme linéaire dual ( $D$ ), dans lequel on doit trouver un vecteur  $y^m \in \mathbb{R}^m$ , tel que :

$$y^*b = \text{Min}\{yb : yA = c, y \geq 0\}. \quad (1.7)$$

Nous analysons ci-dessous les liens entre les formes ( $P$ ) et ( $D$ ).

### 1.4.3 Dualité

Nous présentons la notion de la dualité. Le lecteur pourra se reporter au livre de A. Schrijver [Schrijver 1987]. Considérons un programme linéaire  $\text{Max} \{cx \mid Ax \leq b, x \geq 0\}$  et supposons que nous cherchions à déterminer un majorant de la valeur optimale de ce programme qui soit une fonction des contraintes de ce programme linéaire. Nous pouvons obtenir ce majorant de la manière suivante : Considérons un vecteur  $y \geq 0$  dont la dimension est égale au nombre de lignes de la matrice  $A$ ; notre problème se ramène à  $\text{Min} \{yb \mid yA \geq c, y \geq 0\}$ . La quantité  $yb$ ,  $y$  satisfaisant certaines contraintes, est bien un majorant de la valeur optimale du programme linéaire. Aussi aimerions-nous que ce majorant soit le plus petit possible. On est amené à chercher la valeur optimale du programme linéaire  $\text{Min}\{y^t b \mid y^t A \geq c, y \geq 0\}$ . Ce programme est appelé programme dual. Cette définition de la dualité en programmation linéaire nous permet d'énoncer le théorème suivant.

**Théorème 1.30.** (*Dualité faible*)  $\text{Max}\{cx \mid Ax \leq b, x \geq 0\} \leq \text{Min}\{yb \mid yA \geq c, y \geq 0\}$ .

On remarque que si le programme primal a une solution alors le programme dual a un optimum fini.

**Théorème 1.31.** (*Dualité forte*) Soit  $A$  une matrice  $m \times n$ ,  $c \in \mathbb{R}^n$  et  $b \in \mathbb{R}^m$ , alors on a :

$$\text{Max}\{cx \mid Ax \leq b, \quad x \in \mathbb{R}^n\} = \text{Min}\{yb \mid yA = c, \quad y \in (\mathbb{R}^+)^m\}.$$

Le théorème de dualité forte nous permet d'écrire que pour tout  $x$  solution optimale du programme primal et pour tout  $y$  solution optimale du programme dual, on a  $cx = yAx = yb$  et donc  $y(Ax - b) = 0$ . Cette dernière condition, appelée condition d'écart complémentaires,

signifie que si une contrainte du primal est vérifiée, la variable duale correspondante est nulle. En réécrivant cette condition dans le sens variables primales contraintes duales, on obtient  $(c - yA)x = 0$ , ce qui signifie que si  $x$  est une solution optimale du problème primal alors la fonction objective  $c$  appartient au cône des contraintes qui définissent  $x$ .

Notons que le dual de  $(D)$  est  $(P)$ . La relation entre ces deux programmes est donnée par le théorème suivant.

**Théorème 1.32.** [*Schrijver 1987*] *Étant donnés deux programmes linéaires duaux  $(P)$  et  $(D)$ ,*

- *si  $(P)$  et  $(D)$  ont des solutions réalisables alors ils ont des solutions optimales, et les valeurs des solutions optimales sont égales.*
- *si  $(P)$  n'admet pas de solution réalisable alors, soit  $(D)$  n'admet pas de solution réalisable, soit  $(D)$  est non borné.*
- *si  $(P)$  est non borné alors  $(D)$  n'admet pas de solution réalisable.*

### Remarque 1.33

Tout programme linéaire  $\text{Min}\{cx \mid Ax \leq b \ x \in \mathbb{R}^n\}$  peut être transformé pour ne comporter que des égalités. On ajoute pour cela une variable, appelée variable d'écart, pour chaque contrainte. Le programme linéaire s'écrit alors comme suit :

$$\text{Min}\{cx(A - I) \begin{pmatrix} x \\ s \end{pmatrix} = b, \begin{pmatrix} x \\ s \end{pmatrix} \in \mathbb{R}^{n+m} \text{ et } \begin{pmatrix} x \\ s \end{pmatrix} \geq 0\}, \quad (1.8)$$

où  $I$  est la matrice identité  $m \times m$ ,  $s$  est le vecteur des variables d'écarts. Notons que les deux programmes linéaires sont équivalents. La matrice  $A$  définissant les contraintes du programme linéaire est désormais de dimension  $m \times (m + n)$ , la dimension du vecteur coût  $c$  est  $m + n$  tel que les  $m$  derniers éléments de  $c$  sont nuls.  $\diamond$

Une base  $B$  de la matrice  $A \in \mathbb{R}^{m \times (m+n)}$  de rang plein, est une sous-matrice de  $A$  inversible de taille  $m \times m$ . Les colonnes (variables) de  $A$  qui forment la matrice  $B$  sont dites *de base*, les autres  $N$  sont dites *hors base*. La colonne  $d$  de  $A$  telle que  $B^{-1}d$  est le  $r^{i\text{ème}}$  vecteur unitaire, est dite de base à la  $r^{i\text{ème}}$  ligne. Pour un vecteur  $x \in \mathbb{R}^{m+n}$ , le vecteur  $x_B$  est défini tel que  $(x_B)_r = x_i$  où  $i$  est la  $r^{i\text{ème}}$  colonne de  $A$  dans  $B$  qui est de base à la  $r^{i\text{ème}}$  ligne. Le vecteur  $x_N$  est l'autre partie de  $x$  correspondant aux colonnes de  $N$ . Une solution primale de base  $x \in \mathbb{R}^{m+n}$ , associée à la base courante, est telle que  $x_B = B^{-1}b$  et  $x_N = 0$ . Si  $B^{-1}b \geq 0$  alors la base  $B$  est dite réalisable. La solution de base associée est aussi réalisable. Le vecteur  $y = c_B^t B^{-1}$  est la solution duale de base associée à  $B$ .

**Théorème 1.34.** [*Schrijver 1987*] *Étant donnés deux programmes linéaires duaux  $(P)$  et  $(D)$ , les propositions suivantes sont équivalentes :*

- *La valeur de la solution optimale de  $(P)$  est  $\delta^*$ .*

- Il existe une solution de base réalisable  $x^*$  de  $(P)$  et une solution de base réalisable  $y^*$  de  $(D)$  telles que  $cx^* = y^*b = \delta^*$ .
- Il existe une solution de base réalisable  $x^*$  de  $(P)$  et une solution de base réalisable  $y^*$  de  $(D)$  telles que  $y^*(Ax^* - b) = 0$  et  $cx^* = \delta^*$ .

L'algorithme primal (respectivement, dual) du Simplexe démarre avec une base réalisable de  $(P)$  (respectivement, de  $(D)$ ) et fait une itération en gardant la réalisabilité de la base tout en satisfaisant la troisième condition du théorème. L'optimalité est atteinte quand on obtient une solution duale (respectivement, primale) de base réalisable.

Le coût réduit d'une variable  $x_i$  est égal à  $c_i - c_B^t B^{-1} a_{.i}$  où  $a_{.i}$  est la  $i^{\text{ème}}$  colonne de  $A$ . Un programme linéaire est non borné (donc son dual n'est pas réalisable) quand il existe une variable  $x_i$  de coût réduit négatif et  $B^{-1} a_{.i} \leq 0$ .

#### 1.4.4 Systèmes d'inégalités linéaires monotones et élément extremum

Dans cette section, nous considérons sur une classe particulière de systèmes linéaires, les systèmes monotones, nous étudions la résolution sur l'espace des réels. Ces systèmes sont aisément identifiables au vu de leur structure syntaxique et géométrique.

##### 1.4.4.1 Définitions

Soient le vecteur  $x = (x_1 \cdots x_n) \in \mathbb{R}^n$  et le vecteur  $a = (a_1 \cdots a_n) \in \mathbb{R}_+^n$ .

**Définition 1.35.** Une inégalité linéaire est **inf-monotone** si elle est de la forme : soit  $x_j \geq ax + b$ , avec  $a \geq 0$  et  $a_j = 0$ , ou soit de la forme :  $0 \geq ax + b$ , avec  $a \geq 0$ .

Cette inégalité exprime que la valeur d'une variable borne supérieurement une fonction linéaire croissante des autres variables du système.

##### Exemple 1.36

$$\text{L'inégalité } x_1 \geq 2x_2 + 3x_3 \text{ avec } a = \begin{pmatrix} 0 \\ 2 \\ 3 \end{pmatrix}$$

◇

**Définition 1.37.** Une inégalité linéaire est **sup-monotone** si elle est de la forme :  $x_j \leq ax + b$ , avec  $a \geq 0$  et  $a_j = 0$ , ou encore de la forme :  $0 \leq ax + b$ , avec  $a \geq 0$ .

##### Exemple 1.38

$$\text{L'inégalité } x_1 \leq 2x_2 + 3x_3 \text{ avec } a = \begin{pmatrix} 0 \\ 2 \\ 3 \end{pmatrix}$$

◇

**Définition 1.39.** *Un système linéaire est monotone si toutes les inégalités qui le composent sont soit toutes inf-monotones, soit toutes sup-monotones.*

Ci-dessous, nous donnons les propositions suivantes sans démonstration qui sont immédiates.

**Proposition 1.40.** *Un système inf-monotone peut être normalisé sous la forme  $Ax \leq b$  où la matrice  $A$  a au plus un coefficient strictement négatif par ligne.*

**Exemple 1.41** Soit le système  $Ax \leq b$  où :  $A = \begin{pmatrix} 3 & -2 & 1 \\ 1 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix}$  et  $b = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$ . On remarque que chaque ligne de la matrice  $A$  a un coefficient strictement négatif au plus. Alors le système est inf-monotone. ◇

**Proposition 1.42.** *Un système sup-monotone peut être normalisé sous la forme  $Ax \geq b$  où la matrice  $A$  a au plus un coefficient strictement positif par ligne.*

**Exemple 1.43** Le système  $P = \{(x, y) \mid 3x + 2y \leq 3, y \leq 2x + 1, 2x \leq 2y + 1, 0 \leq x, 0 \leq y\}$ . Ce système peut s'écrire sous la forme  $A_P x \geq b$  avec :

$$A_P = \begin{pmatrix} -3 & -2 \\ +2 & -1 \\ -2 & +2 \\ +1 & 0 \\ 0 & +1 \end{pmatrix} \text{ et } b = \begin{pmatrix} -3 \\ -1 \\ -1 \\ 0 \\ 0 \end{pmatrix}.$$

La matrice  $A_P$  a au plus un coefficient strictement positif par ligne. Alors le système représente l'ensemble des solutions d'un système sup-monotone. ◇

**Exemple 1.44** Le système  $L = \{(x, y) \mid 4 \leq x + y, 2x + 2y \leq 5, 2y \leq 3x + 1, x \leq 2y + 3\}$  peut se réécrire  $A_L x \leq b$  avec :

$$A_L = \begin{pmatrix} -1 & -1 \\ +2 & +2 \\ -3 & 2 \\ +1 & -2 \end{pmatrix} \text{ et } b = \begin{pmatrix} -4 \\ 5 \\ 1 \\ 3 \end{pmatrix}.$$

Le système  $L$  n'est ni inf-monotone, ni sup-monotone et n'est donc pas un système monotone. ◇

**Définition 1.45.** *Une inégalité linéaire bimonotone est de la forme  $a_i x_i + a_j x_j \leq c$ , avec le produit  $a_i a_j$  négatif.*

**Exemple 1.46** Soit le système  $Ax \leq b$  avec :

$$A = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \text{ et } b = \begin{pmatrix} -2 \\ 3 \\ 1 \end{pmatrix}. \text{ Le produit des coefficients de chaque inégalité est négatif,}$$

alors, les inégalités sont bimonotone. Chaque ligne de la matrice  $A$  a au plus un coefficient strictement positif et négatif  $\diamond$

#### 1.4.4.2 Existence d'une solution extrême

**Proposition 1.47.** Soit  $\Gamma$  l'ensemble des solutions d'un système monotone  $Ax \leq b$ . Si  $x$  et  $y$  sont deux éléments de  $\Gamma$ , alors leur borne inférieure,  $\inf(x, y)$ , appartient à  $\Gamma$ .

**Preuve :**

Posons  $z = \inf(x, y)$ . Pour toute contrainte  $C$  du système d'inégalités, on a :

- pour la solution  $x : x_j \geq ax + b \geq az + b$ ,
- pour la solution  $y : y_j \geq ay + b \geq az + b$ .

Par conséquent :  $\min\{x_j, y_j\} = z_j \geq az + b$ , ce qui montre que  $z$  satisfait  $C$ .

□

**Théorème 1.48 (Veinott, Cottle)** [Veinott 1968] [Cottle 1972]. Soit  $\Gamma$  l'ensemble des solutions d'un système inf-monotone (respectivement, sup-monotone)  $Ax \leq b$ . Les points (1) et (2) suivants sont équivalents :

1. L'ensemble  $\Gamma$  est un inf-demi-treillis (respectivement, sup-demi-treillis).
2. Si  $x$  et  $y$  sont deux éléments de  $\Gamma$  alors leur minimum  $x \wedge y$  (respectivement, maximum  $x \vee y$ ) appartient à  $\Gamma$ .

**Propriété 1.** L'ensemble des solutions  $\Gamma$  d'un système bimonotone  $Ax \leq b$  est un treillis.

Le théorème suivant est important car il garantit l'existence d'une solution minimale, ou maximale, d'un ensemble  $\Gamma$ . Il est appliqué dans les parties suivantes car il prouve l'existence de la trajectoire au plus tôt, ou le plus grand contrôle.

**Théorème 1.49.** L'ensemble  $\Gamma$  a un plus grand (respectivement, plus petit) élément si l'ensemble est non vide et a un majorant (respectivement, minorant).

**Exemple 1.50** Soit  $Ax \leq b$  avec  $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \\ -1 & 0 \end{pmatrix}$  et  $b = \begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix}$  (voir figure 1.4). On remarque

que le système est bimonotone. L'ensemble  $\Gamma$  est non vide, comme  $\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \Gamma$  et  $\begin{pmatrix} 4 \\ 6 \end{pmatrix}$  est majorant

de  $\Gamma$ . Le plus grand élément est :  $\begin{pmatrix} 8/3 \\ 10/3 \end{pmatrix}$ . Aussi  $\begin{pmatrix} -5 \\ -6 \end{pmatrix}$  est minorant de  $\Gamma$ . Le plus petit élément est  $\begin{pmatrix} 0 \\ -2 \end{pmatrix}$ .  $\diamond$

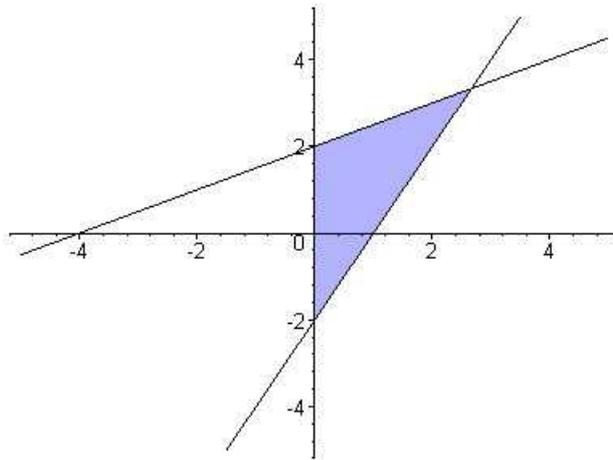


Fig. 1.4 : Courbe du système  $\Gamma$ .

#### 1.4.5 Bref rappels sur la théorie de la complexité

La théorie de la complexité est née à la suite des travaux d'Edmonds [Edmonds 1962] et de Cook [Cook 1971] [Kerivin 2000]. Elle offre un cadre d'étude mathématique dans lequel les problèmes peuvent être classés en problèmes faciles ou difficiles. Dans cette partie, nous allons en rappeler brièvement quelques définitions de base. Plus d'informations sur la théorie de la complexité peuvent être trouvées dans des ouvrages, comme Garey et Johnson [Garey 1979].

Nous appellerons *problème* une question possédant différents paramètres, ou variables libres, dont les valeurs ne sont pas spécifiées. Un *problème*  $P$  est décrit par la donnée de :

- une description générale de tous ses paramètres ;
- les contraintes auxquelles doit satisfaire la réponse à la question, c'est-à-dire, *la solution* du problème.

La donnée d'un ensemble de valeurs des paramètres du problème sera appelée une *instance*. Nous considérons que tous les nombres présent dans chaque *instance*  $I$  sont codés en binaire. La taille de l'instance, notée  $|I|$ , définit le nombre de bits nécessaire pour écrire  $I$  avec cette convention. Un algorithme  $A$  est une suite séquentielle d'instructions qui modifient l'état de la mémoire. Il prend en entrée un élément  $I$  et calcule un *résultat*  $A(I) \in R$  ( $R$  est l'ensemble de solutions de  $P$ ). La fonction de complexité en temps pour un algorithme exprime le nombre d'unités de temps nécessaires pour chaque taille d'instance donnée, pour produire la solution du problème dans le *pire des cas*. Usuellement, un algorithme est dit *polynomial* (respectivement, *exponentiel*) si sa complexité est bornée par une fonction polynômiale (respectivement, exponentielle) dépendant

de  $|I|$ .

Une notion moins exigeante de complexité définie à partir d'un autre codage dit unaire est la suivante. Notons  $I_u$  l'instance  $I$  où tous les nombres sont écrits en unaire (chaque symbole est assimilé à un entier naturel  $n$  et est représenté par  $n$  occurrences de 1 suivies d'un 0. Par exemple 5 son codage unaire est : 11 1110). La taille en unaire de l'instance  $I$ , notée  $|I_u|$  est alors le nombre de bits nécessaire pour écrire  $I_u$ . Un algorithme *pseudo-polynomial* est un algorithme dont la complexité est bornée par une fonction polynômiale de  $|I_u|$  et dépend de l'amplitude des coefficients.

## 1.4.6 Algorithmes

Les algorithmes basés sur la programmation linéaire (PL) sont de plus en plus utilisés pour l'optimisation de divers problèmes industriels tels que l'optimisation des flux de transports, ou la planification de production. Dans cette partie, on présente un bref état de l'art de ces algorithmes.

### 1.4.6.1 Méthodes génériques

La méthode du Simplexe [AnnexeB.2], conçue par Dantzig [Tj. C. Koopmans 1951], est une méthode de la programmation linéaire. Bien que certains exemples montrent que la durée d'exécution est exponentielle, la méthode est efficace dans la pratique car sa durée d'exécution est polynômiale *en moyenne* ([Schrijver 1987], chapitre 11). L'idée de la méthode du Simplexe est de partir d'une solution de base (un point extrême) de l'ensemble des contraintes du problème linéaire, de minimiser (respectivement, maximiser) la valeur de la fonction objectif jusqu'à ce que le point optimal soit atteint. Cette idée est due à Fourier [Olms 1970].

Fourier-Motzkin est une méthode d'élimination introduite par Fourier [Fourier 1827], et redécouverte plus tard par Dines (1918-1919) et Motzkin (1936) [Motzkin 1936], (voir [Schrijver 1987] pour plus de détails). Cet algorithme n'est pas polynomial mais doublement exponentiel. Il génère un nombre exponentiel d'inégalités dans le processus d'élimination de variables, ce qui demande un grand espace mémoire. Toutefois, D. Hochbaum [Hochbaum 1993] montre comment mettre en œuvre cet algorithme d'une manière efficace pour les programmes linéaires où chaque inégalité contient au plus deux variables : la complexité en temps est en  $O(mn^2 \log m)$ . Dans la pratique, le processus d'élimination de Fourier-Motzkin est exploité dans de nombreux problèmes et généralement le Simplexe.

Dans d'autres algorithmes génériques, le nombre d'opérations ne dépend pas seulement du nombre de variables  $n$  et des contraintes  $m$ , mais aussi de *l'ampleur des coefficients* [Karmarkar 1984]. En 1984, N. Karmarkar propose la méthode projective. Sa complexité dans *le pire cas* est pseudo-polynomiale en  $O(Ln^{3,5})$  ( $L$  étant le nombre de bits nécessaires pour représenter les données du problème) et les expérimentations sur les problèmes pratiques montrent que la méthode peut "raisonnablement" être comparée à l'algorithme du Simplexe. Depuis lors, plusieurs méthodes de point intérieur ont été proposées et étudiées. Une des méthodes les plus

célèbres est la méthode prédictive/corrective qui fonctionne bien en pratique même si son étude théorique est encore imparfaite. La recherche d'un algorithme polynomial, avec une complexité indépendante des coefficients, a continué dans le dernier quart de siècle. Andersson et Vorobyov [D.Andersson 2006] proposent un algorithme pour la résolution d'un programme linéaire  $\min\{\Sigma x_i \mid S\}$  ( $S$  l'ensemble des contraintes de la forme  $x_i \geq x_j + \beta$  avec  $0 < \beta < 1$ ). Sa complexité, pour le cas où  $\beta$  est le même pour toutes les contraintes, est en  $O(mn^2)$ , alors que la complexité de cet algorithme pour le cas où  $\beta$  varie dans les contraintes est en  $O(mn^2 \log m)$ .

#### 1.4.6.2 Méthodes graphiques

Il est parfois plus facile de discuter d'un problème en utilisant la théorie des graphes qui peuvent être traités par la théorie des chemins (algorithme de Bellman-Ford [Cormen 2001]). On ne considère pas ici des systèmes spécialement monotones. Pratt [Pratt 1977] a suggéré qu'un programme linéaire, où chaque inégalité a deux variables de type  $x - y + c \leq 0$ , peut être représenté par un graphe : chaque variable est représentée par un sommet, et un sommet supplémentaire  $x_0$ . Toute inégalité de deux variables est représentée par une arête entre les deux sommets  $(x, y)$  et le poids de cette arête est  $c$ . Le graphe se compose donc de  $n$  sommets et  $m$  arêtes, sachant qu'il peut y avoir plusieurs arêtes entre deux sommets. Shostak [Shostak 1981] a généralisé le graphe pour des inégalités de type  $a \times x + b \times y \leq c$ , où chaque arête entre les deux sommets  $x, y$  est associée par le triplet  $(a, b, c)$ . Dans ce mémoire, on se limite à la première interprétation avec des conditions sur les coefficients  $a$  et  $b$ .

Développé par Aspvall et Shiloach [Aspvall 1980], un autre algorithme polynomial a le même principe que l'algorithme de Bellman-Ford et a un temps de calcul en  $(O(mn))$ . Un autre développé par Megiddo [Megiddo 1983] a une complexité en  $O(mn^3 \log m)$ . Cohen et Megiddo [Cohen 1993] ont développé un algorithme pour ce problème avec une complexité en  $O(mn^2(\log m + \log^2 n))$ . Ils ont aussi donné un algorithme pour trouver une solution faisable dans un cas particulier d'inégalités monotones, avec une complexité en  $O(n \log n + n + mn \log^3 m \log n + mn \log^5)$ . Leur algorithme exploite une structure décrite par Shostak [Shostak 1981], et la principale caractéristique commune entre l'ensemble de ces algorithmes est la détermination des bornes supérieures et inférieures pour chaque variable en suivant des chemins et des cycles dans le graphe.

La figure 1.5 représente un comparatif entre les algorithmes de résolution d'un programme linéaire en fonction de leur complexité dans le pire des cas.

Pour l'algorithme du Simplexe, il reste un challenge pour les chercheurs, de prouver, ou de réfuter la croyance habituelle que cet algorithme est supérieur à tous les autres algorithmes utilisés pour la résolution d'un programme linéaire.

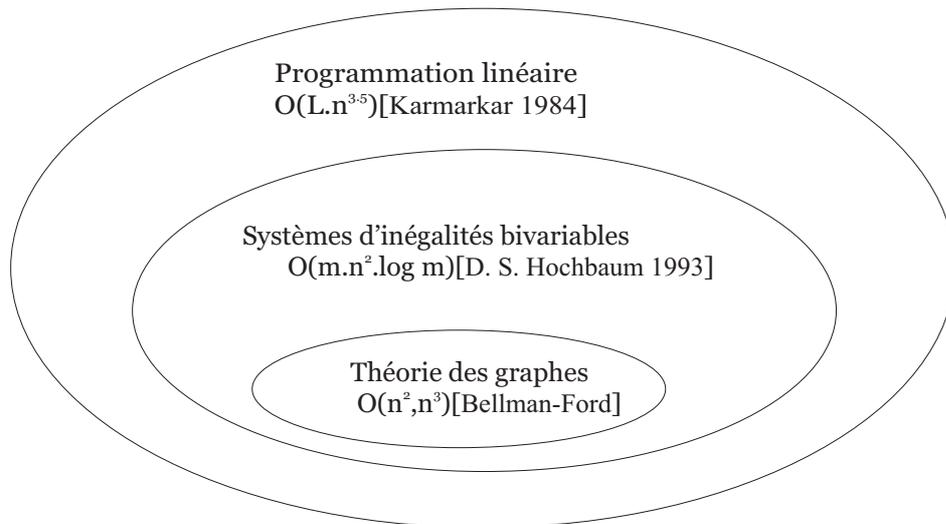


Fig. 1.5 : La complexité des algorithmes de différents systèmes d'inégalités dans le pire des cas.

## 1.5 Conclusion

Dans ce premier chapitre, nous avons présenté les principaux outils mathématiques utilisés par la suite. Après un bref rappel sur la structure algébrique des dioïdes, la théorie de la programmation linéaire a été présentée. Cette théorie permet l'analyse et la résolution des systèmes d'inégalités  $A.x \leq b$ . Ensuite, la notion de systèmes monotones, sera utilisée pour calculer le plus petit (respectivement, plus grand) élément des systèmes d'inégalités en algèbre standard ; cette notion sera appliquée pour le calcul de la commande présentée au chapitre 5. Dans la dernière partie de ce chapitre, nous avons présenté un bref état de l'art de quelques algorithmes utilisés pour la résolution des problèmes généraux ou spécifiques de la programmation linéaire.



## Chapitre 2

# Réseaux de Petri temporisés et temporels, et leurs modèles algébriques

### 2.1 Introduction

Les réseaux de Petri (RdP) sont des modèles de représentation souvent utilisés dans le domaine des SED, car ils permettent de modéliser, d'évaluer voire de piloter des systèmes dynamiques.

Afin de prendre en compte les aspects temporels de ces derniers, plusieurs extensions du modèle des réseaux de Petri ont été proposées. Dans ce chapitre, nous nous intéresserons notamment au cas des systèmes temporels. Les réseaux que nous étudions sont constitués d'une partie statique, donnée par la topologie du réseau, et d'une partie dynamique, donnée par les règles de tir des transitions.

Les graphes que nous étudions dans ce chapitre sont une sous classe des réseaux de Petri, appelée classe des graphes d'événements temporisés (notés GET), fournissent une représentation graphique pertinente des systèmes à événements discrets (SED). Les graphes d'événements temporisés admettent une représentation linéaire sur une structure algébrique de dioïde, les résultats sont bien connus [Cohen 1998], [Cottenceau 1999], [Menguy 1997] et [Lhommeau 2003]. Concernant les graphes d'événements temporels, une modélisation dans l'algèbre des dioïdes a été faite par [Didi-Alaoui 2005].

L'objectif de ce chapitre est d'établir un modèle dans l'algèbre standard qui reflète le comportement dynamique des graphes considérés, à savoir, les graphes d'événements temporisés, les graphes d'événements P-temporels, et les graphes d'événements T-temporels. Le modèle proposé admet une représentation linéaire. Cette représentation est bien adaptée pour aborder, par exemple, les problèmes de commande ou d'évaluation de performances.

Ce chapitre est structuré comme suit. Dans la première partie, nous rappellerons, tout d'abord, les notions de base concernant les réseaux de Petri, en évoquant quelques propriétés de ces derniers. L'introduction de sous-classes importantes que constituent les graphes d'événements

et les graphes d'état, se fera aussi dans cette partie. Cela nous permettra dans la deuxième partie de ce chapitre de présenter le modèle algébrique des graphes d'événements temporisés dans l'algèbre des dioïdes, et de présenter une nouvelle modélisation utilisant les dateurs, ou les compteurs, non pas dans l'algèbre  $(\max, +)$ , ou  $(\min, +)$ , mais dans l'algèbre standard.

Dans une troisième partie, nous présenterons les graphes d'événements temporels. Nous introduirons, en particulier, une nouvelle modélisation des graphes d'événements P-temporels qui utilise les dateurs dans l'algèbre standard. Ensuite, nous présenterons les graphes d'événements T-temporels et leur modèle dans l'algèbre standard. Enfin, nous présenterons une classification de différents modèles vus dans ce chapitre. Le critère pris afin de réaliser cette classification sera le type du système mathématique représentant chacun de ces modèles. Ce chapitre s'appuie sur les documents [Declerck 2007] et [Guezzi 2009].

## 2.2 Les réseaux de Petri (RdPs)

### 2.2.1 Rappels et notions de base

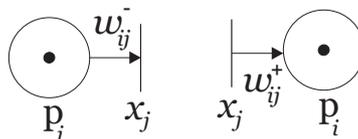
Les RdP ont été introduits par Carl Adam Petri en 1962. Ils constituent un puissant outil graphique et mathématique de représentation de phénomènes complexes et de mécanismes séquentiels [Petri 1962]. Ils sont largement utilisés pour l'analyse et la modélisation des systèmes à événements discrets [Murata 1989]. Ils permettent la modélisation des processus complexes mettant en jeu des phénomènes de synchronisme et de choix.

Nous présentons dans cette section des notions élémentaires sur cet outil. Des développements plus approfondis peuvent être trouvés dans [Petri 1962], [Murata 1989] et [David 2004].

**Définition 2.1. (Réseau de Petri)** Un réseau de Petri (RdP) est représenté par un triplet  $\mathcal{N} = \langle P, Tr, W \rangle$  où :

- $P$  et  $Tr$  sont des ensembles finis disjoints ( $P \cap Tr = \emptyset$ );
- $W = (P \times Tr) \cup (Tr \times P) \rightarrow \mathbb{N}$  est la fonction de poids;

Les éléments de  $P$  sont appelés des places et les éléments de  $Tr$  sont appelés des transitions.



**Fig. 2.1** : Les valeurs  $w_{ij}^+$  et  $w_{ij}^-$ .

La valeur  $w_{ij}^-$  (respectivement,  $w_{ij}^+$ ) représente le poids de l'arc qui relie une transition  $x_j$  à une place  $p_i$  située en amont (respectivement, une transition  $x_j$  à une place  $p_i$  située en aval). Si  $w_{ij}^- = 0$  (respectivement,  $w_{ij}^+ = 0$ ), alors, il n'existe pas d'arc reliant une transition  $x_j$  à une

place  $p_i$  située en amont (respectivement, une transition  $x_j$  à une place  $p_i$  située en aval). On définit la matrice d'incidence d'un RdP comme suit :

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

La matrice  $W^-$  est la matrice d'incidence dite "avant" telle que  $W^- = [w^-]$ , et  $W^+$  est la matrice d'incidence dite "arrière" telle que  $W^+ = [w^+]$ .

### Remarque 2.2

Les sous-classes (graphes d'événements) de réseaux de Petri (RdPs) traitées par la suite, seront considérées comme ordinaires : les poids des arcs seront unitaires.  $\diamond$

**Définition 2.3. (Réseau de Petri marqué)** Un réseau de Petri marqué est un couple  $\langle N, M_0 \rangle$  tel que :

- $N$  est un réseau de Petri,
- $M_0 : P \longrightarrow \mathbb{N}$  est un marquage dit marquage initial.

Nous adoptons désormais les notations suivantes dans la suite de ce mémoire :

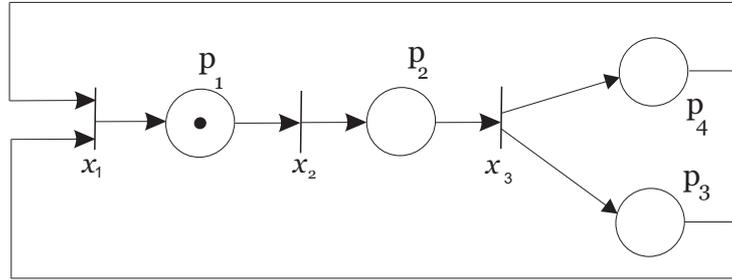
- $\bullet x$  l'ensemble des places situées en amont de la transition  $x$ ,
- $x^\bullet$  l'ensemble des places situées en aval de la transition  $x$ ,
- $\bullet p$  l'ensemble des transitions situées en amont de la place  $p$ ,
- $p^\bullet$  l'ensemble des transitions situées en aval de la place  $p$ .

Un RdP est un modèle dynamique dont l'évolution est liée à celle du marquage. Son état à un instant donné est représenté par son marquage  $M_0$  à cet instant. La caractérisation d'un RdP à un instant donné est souvent donnée par le couple  $(N, M_0)$ .

Pour un RdP non temporel, on dit qu'une transition est tirable (franchissable) si, quelque soit  $p_i \in \bullet x_j$ ,  $M(p_i) \geq w_{ij}^-$ . Autrement dit, si toute place  $p_i$  située en amont de  $x_j$  contient un nombre de jetons au moins égal au poids attaché à l'arc allant de  $p_i$  vers  $x_j$ . Dans le cas d'un RdP ordinaire, il suffit que toutes les places d'entrée d'une transition contiennent au moins un jeton pour qu'elle soit franchissable.

Une transition franchissable peut être franchie, ou non. Franchir une transition  $x_j$  consiste à :

- retirer  $w_{ij}^-$  jetons de toute place  $p_i \in \bullet x_j$ ,
- ajouter  $w_{ij}^+$  jetons dans toute place  $p_i \in x_j^\bullet$ .



**Fig. 2.2 :** Exemple d'un RdP ordinaire.

Une séquence de franchissement à partir d'un marquage  $M_0$  est représentée par une suite de transitions. Si nous considérons le RdP de la figure 2.2 et son marquage initial  $M_0 = (1, 0, 0, 0)^t$ , nous dirons que nous sommes passés de  $M_0$  à  $M_2 = (0, 0, 1, 1)$  en effectuant le tirage de la séquence  $\sigma < x_2, x_3 >$  et nous écrivons  $M_0 \xrightarrow{\sigma} M_2$ .

L'équation fondamentale qui caractérise l'évolution dynamique d'un RdP est :

$$M_k = M_0 + W.S, \quad (2.1)$$

où  $M_k$  est le marquage que l'on atteint à partir de  $M_0$  après la séquence de franchissement  $\sigma$  réalisable.

Une séquence de franchissement est une suite de transition  $x_i, x_j, \dots, x_k$  qui peuvent être franchies successivement à partir d'un marquage donnée.

$S$  est un vecteur qui représente cette séquence  $\sigma$ . Sa dimension est égale au nombre de transitions que contient le RdP. La composante  $S_j$  correspond au nombre de fois où la transition  $x_j$  a été franchie pendant la séquence  $\sigma$ .

Reprenons l'exemple de la figure 2.2. Le vecteur  $S$  est égal à  $(0, 1, 1)^t$ , ce que l'on peut interpréter de la manière suivante :  $x_2$  et  $x_3$  sont franchies chacune une fois pour passer de  $M_0$  à  $M_2$ . La matrice d'incidence  $W$  est donnée par :

$$W = W^+ - W^- = \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{array}{ccc} x_1 & x_2 & x_3 \\ \left( \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array} \right) & - & \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{array}{ccc} x_1 & x_2 & x_3 \\ \left( \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{array} \right) & = & \left( \begin{array}{ccc} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array} \right) .$$

Le marquage  $M_2$  peut être aussi calculé à partir des matrices  $M_0$ ,  $S$  et  $W$  données ci-dessus et en appliquant l'équation 2.1, soit

$$M_2 = M_0 + W.S = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

### 2.2.2 Quelques propriétés des RdPs

Les principales propriétés des RdPs, sont classées en deux groupes : celles qui dépendent du marquage initial sont appelées "propriétés comportementales", celles qui ne dépendent que de la structure du RdP sont appelées "propriétés structurelles".

Pour étudier et vérifier les propriétés comportementales, on établit l'arbre de marquages, appelé aussi graphe de marquages. Il est constitué de nœuds correspondant à un marquage atteignable, et d'arcs orientés, correspondant au franchissement d'une transition qui fait évoluer le RdP d'un état, ou marquage, à un autre.

- **Accessibilité** : Le problème d'accessibilité consiste à savoir si l'on peut atteindre un marquage  $M'$  à partir de  $M_0$ .

Soit un RdP  $(N, M_0)$ , on dit qu'un marquage  $M'$  est accessible (ou atteignable) à partir de  $M_0$  s'il existe une séquence de franchissement  $\sigma$  telle que  $M_0 \xrightarrow{\sigma} M'$ .  $\mathcal{R}(M_0)$  est l'ensemble des marquages que l'on peut atteindre en partant de  $M_0$ .

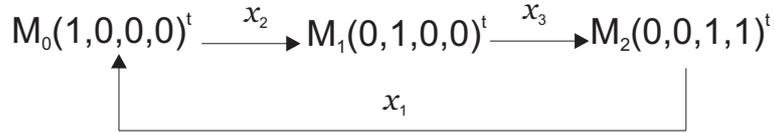
- **Bornitude** : Un RdP  $(N, M_0)$  est borné si, quelques soient la place  $p_i$  et le marquage accessible  $M'$  à partir  $M_0$ , le nombre de jetons dans cette place  $p_i$  est borné :

$$\forall p \in P, \exists k \in \mathbb{N} \text{ tel que } \forall M \in \mathcal{R}(M_0) \quad M(p) \leq k.$$

- **Vivacité** : Une transition  $x_j$  est dite vivante si elle peut être franchie quelque soit le marquage atteint :

$$\forall M \in \mathcal{R}(M_0), \exists M' \in \mathcal{R}(M) \text{ tel que } x_j \text{ soit franchissable pour } M'.$$

- Un RdP  $(N, M_0)$  est vivant si chacune de ses transitions est vivante.
- Dans un RdP  $(N, M_0)$ , un état de blocage est un marquage tel qu'aucune transition n'est validée.



**Fig. 2.3 :** Graphe des marquages accessibles du RdP de la figure 2.2.

Nous pouvons dire que le graphe de la figure 2.2 est vivant et borné, en analysant son graphe de marquages représenté dans la figure 2.3.

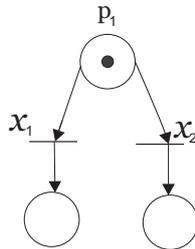
Comme la vérification de ces propriétés (vivacité et accessibilité) utilise le graphe de marquages basé sur le marquage, cette méthode peut devenir inefficace dans le cas d'un RdP non borné.

### 2.2.3 Sous classes des RdPs

Dans l'étude des systèmes dynamiques à événements discrets (SDED), on rencontre fréquemment des phénomènes de types concurrence et synchronisation, qui sont modélisés par différentes sous-classes de RdPs que nous décrivons maintenant.

- **Graphe d'états** : Un RdP ordinaire est un graphe d'états, si et seulement si, toute transition a exactement une place d'entrée et une place de sortie :

$$|\bullet x| = |x^\bullet| = 1, \quad \forall x \in Tr.$$



**Fig. 2.4 :** Conflit structurel représenté par un graphe d'états.

Le RdP de la figure 2.4 représente un graphe d'états. Les transitions  $x_1$  et  $x_2$  de la figure 2.4 sont en conflit structurel puisqu'elles ont une place d'entrée commune  $p_1$ . En effet, le jeton de la place  $p_1$  peut franchir soit  $x_1$  soit  $x_2$ , mais pas les deux transitions simultanément.

• **Graphe d'événements** : La définition d'un graphe d'événements est duale de celle d'un graphe d'états. Un RdP est un graphe d'événements si, et seulement si, toute place a exactement une transition d'entrée et une transition de sortie :

$$|\bullet p| = |p\bullet| = 1, \quad \forall p \in P.$$

Le graphe de la figure 2.2 est un graphe d'événements.

Nous rappelons ici brièvement quelques propriétés structurelles des graphes d'événements qui sont indépendantes du marquage initial. L'analyse de ces propriétés utilise l'algèbre linéaire [Sifakis 1978]. Nous définissons ci-dessous les p-invariants et les t-invariants.

**Définition 2.4.** On dit qu'un vecteur  $\theta$  non nul est un p-invariant, si  $\theta$  vérifie la relation suivante :

$$\theta.W = 0,$$

où  $\theta$  est un vecteur ligne entier non-négatif, appelé aussi P-semi-flot, et  $W$  est la matrice d'incidence.

L'ensemble des places correspondant aux composante non nulles d'un p-invariant  $\theta$ , noté  $\|\theta\|$ , est appelé support du p-invariant. Le support d'un p-invariant est minimal s'il ne contient strictement aucun ensemble qui soit un support de p-invariant. Un p-invariant  $\theta$  est minimal s'il n'en existe aucun autre dont toutes les composantes soient inférieures ou égales à celles de  $\theta$ . Il existe un unique p-invariant minimal correspondant à un support minimal donné; il est appelé p-invariant de support minimal [Murata 1989].

La recherche des p-invariants pour les graphes d'événements est présentée dans [Proth 1995] sous la forme suivante.

Soit  $\theta = (\theta_1, \theta_2, \dots, \theta_n)$  un vecteur à composantes entières non négatives, où  $n$  est le nombre de places du graphe d'événements considéré. Soit  $c$  un circuit élémentaire de ce graphe et  $p = \{p_1, p_2, \dots, p_n\}$  l'ensemble des places du graphe. Si, pour  $i$  allant de 1 à  $n$ , on a :

$$\theta_i = \begin{cases} 1 & \text{si } p_i \in c, \\ 0 & \text{sinon,} \end{cases}$$

alors,  $\forall M_k \in \mathcal{R}(M_0)$ , on a :

$$\theta.M_0 = \theta.M_k.$$

Ce résultat découle immédiatement de l'équation fondamentale de l'évolution dynamique d'un RdP (l'équation 2.1.)

Si nous considérons un graphe d'événements pour lequel toutes les places appartiennent à un support de p-invariant, alors ce graphe d'événements est borné.

**Propriété 2.** *Dans un graphe d'événements, la somme des marques des places d'un circuit donné est constante.*

**Preuve :** Rappelons qu'un circuit est un chemin qui commence et se termine au même sommet. Alors, si une transition franchissable appartenant à un circuit est franchie, son franchissement prend un jeton dans la place en amont de la transition pour le remettre dans la place en aval de la transition. L'opération de franchissement d'un circuit laisse donc invariant le nombre de jetons du circuit. □

**Définition 2.5.** *Un vecteur  $\omega$  non nul est un t-invariant, s'il vérifie l'équation suivante :*

$$W.\omega^t = 0,$$

où  $\omega$  est un vecteur ligne à composantes entières non-négatives, appelé aussi T-semi-flot, dont la dimension est le nombre de transitions du RdP.

L'ensemble des transitions qui correspondent aux composantes non nulles d'un t-invariant noté  $\|\omega\|$ , est appelé support du t-invariant. Il est minimal s'il ne contient strictement aucun ensemble non vide qui soit un support de t-invariant. Un t-invariant  $\alpha$  est minimal s'il n'en existe aucun autre dont toutes les composantes soient inférieures ou égales à celles de  $\alpha$ . Il existe un unique T-invariant minimal correspondant à un support minimal donné ; il est appelé t-invariant de support minimal [Murata 1989].

L'existence d'un t-invariant pour un graphe d'événements est une condition nécessaire pour que le graphe ait un fonctionnement répétitif (cyclique). Soit maintenant un graphe d'événements avec  $m$  transitions. Le vecteur  $\omega$ , à  $m$  composantes toutes égales à 1, est l'unique t-invariant [Proth 1995]. Une autre manière d'exprimer ce résultat est de dire que l'on retrouve le même marquage initial après avoir franchi une seule fois chaque transition.

**Théorème 2.6.** [Proth 1995] *Un graphe d'événements autonome<sup>1</sup> est sans blocage et vivant, si et seulement si, chaque circuit élémentaire contient au moins un jeton.*

Après avoir présenté des notions de base sur les RdPs, portons notre attention sur les graphes d'événements (temporisés et temporels). L'étude du comportement au cours du temps des systèmes dynamiques, notamment l'évaluation de leur performance (temps de cycle,...), a conduit à associer la notion de temps aux modèles de type RdP.

Plusieurs modèles RdP intégrant le temps ont été proposés. Ces modèles peuvent être regroupés en trois classes : les modèles déterministes, les modèles de type intervalle et les modèles stochastiques. La première classe considère des valeurs déterministes pour les durées d'activité, alors que la troisième classe considère des valeurs probabilistes. Dans nos travaux on s'intéresse aux modèles de type intervalle où les tâches sont définies sur des intervalles de temps. Le paramètre temps sur ces derniers sera introduit de différentes manières.

1. On dit qu'un graphe d'événements est autonome s'il ne contient pas de transition source.

## 2.3 Graphe d'événements temporisés

### 2.3.1 Temporisation et fonctionnement d'un graphe d'événements

L'introduction d'un nouveau paramètre qui est le temps sur les graphes d'événements va permettre de définir les graphes d'événements temporisés (GET). Deux approches sont envisageables :

1. soit en associant à chaque transition  $x$  une durée minimale de tir  $\tau(x)$ , qui représente le temps de réservation d'un jeton dans une place en amont, avant d'être disponible à nouveau, dans une place située en aval. On parle alors de graphes d'événements t-temporisés.
2. soit en associant à chaque place une durée minimale  $\tau(p)$ , qui correspond à un temps d'indisponibilité d'une marque après son arrivée dans cette place, avant d'être utile pour un nouveau franchissement. On parle alors de graphes d'événements p-temporisés.

Du fait de la possibilité de transformer une place temporisée en une transition temporisée, et *vice versa*, l'équivalence entre les deux types de graphes peut être établie [Chrétienne 1983].

#### Fonctionnement au plus tôt d'un GET

On appelle fonctionnement au plus tôt d'un GET, le mode de fonctionnement suivant.

Toutes les transitions internes et puits (pas de place en aval de la transition) sont franchies dès que possible. Les jetons qui arrivent dans une place, pourront contribuer au franchissement de la transition située en aval dès que le temps de séjour minimum se termine. Les transitions sources (pas de place en amont de la transition) sont en permanence franchissables, mais ne sont franchies que sur l'occurrence d'événements externes associés à ces transitions.

**Définition 2.7.** (*Règles de fonctionnement temporisé*) *Le fonctionnement au plus tôt d'un réseau de Petri temporisé obéit aux points suivants.*

1. *Une transition  $x$  est validée s'il y a au moins un jeton disponible dans chacune des places de  $\bullet x$  ( i.e., située en amont de  $x$ ).*
2. *Une transition  $x$  est tirée dès qu'elle est validée. On note  $x(k)$  l'instant où le tir numéroté  $k$  de la transition  $x$  a lieu. Ce tir transforme le marquage : un jeton est prélevé dans chacune des places de  $\bullet x$ , et simultanément un jeton est ajouté dans chacune des places de  $x\bullet$ .*

#### Fonctionnement FIFO

Le fonctionnement FIFO va permettre une numérotation des événements et ainsi une écriture correcte des relations. Nous définissons maintenant ce fonctionnement FIFO.

**Définition 2.8.** - *Si le  $k$ -ième jeton qui arrive dans une place d'un graphe d'événements p-temporisé est aussi le  $k$ -ième jeton disponible pour franchir une transition en aval, alors cette place fonctionne selon la règle FIFO<sup>2</sup>.*

---

2. First In, First Out : premier entré, premier sorti.



**Fig. 2.5** : Recyclage d'une transition et d'une place pour garantir le fonctionnement FIFO.

- Si le  $k$ -ième tir d'une transition d'un graphe d'événements  $t$ -temporisé à commencer est aussi le  $k$ -ième tir à se terminer, alors cette transition fonctionne selon la règle FIFO.
- On dit qu'un graphe d'événements est FIFO, si toutes ses places et ses transitions sont FIFO.

On peut obtenir ce fonctionnement FIFO de la manière suivante.

- On peut imposer une condition structurelle sur le graphe d'événements pour garantir le fonctionnement FIFO des transitions. Cela peut se traduire par l'introduction d'une boucle sur chaque transition comme indiqué dans la figure 2.5a.

- De même, on peut introduire une boucle de façon identique à celle faite sur une transition, pour garantir un fonctionnement FIFO pour les places de la figure 2.5b. Il ne peut avoir de dépassement car le nombre de jetons est limité à un.

- Il suffit aussi que le temps de séjour, lié à une place, soit constant, pour que celle-ci fonctionne sous l'hypothèse FIFO.

Nous pouvons maintenant numérotter les événements. Chaque jeton, ajouté dans une place  $p \in x^\bullet$  par le tir de  $x$  numéroté  $k$ , sera numéroté  $k + M_0(p)$  et sera disponible pour le tir des transitions de  $p^\bullet$  à partir de l'instant  $x(k + M_0(p)) + \tau(p)$ , les grandeurs  $\tau(p)$  et  $M_0(p)$  étant la temporisation et le marquage initial associés à la place  $p$ .

### 2.3.2 Compteurs

Comme dans le cadre de la théorie des systèmes classiques, les équations qui décrivent le fonctionnement dynamique de ces systèmes, sont fonction du temps  $t$ . D'une manière analogue, nous allons associer à chaque transition, une fonction  $x(t)$  appelée **compteur**.

Contrairement aux dateurs, nous pouvons distinguer plusieurs interprétations de la définition des compteurs.

- Le compteur  $x(t)$  indique le numéro de la dernière activation de la transition  $x$  survenue avant, ou à,  $t$ .

$$x(t) = \sup\{k \text{ tel que } d(k) \leq t\}.$$

- Le compteur  $x(t)$  indique le numéro de la première activation de la transition  $x$  survenue à  $t$ , ou après,  $t$ .

$$x(t) = \inf\{k \text{ tel que } d(k) \geq t\}.$$

**Définition 2.9.** *Un compteur  $x$  est une application monotone croissante de  $\mathbb{R}$  dans  $\mathbb{Z} \cup \{+\infty, -\infty\}$ . On a donc :*

$$x(t + \Delta t) \geq x(t), \quad \text{avec } \Delta t \geq 0.$$

Dans ces deux interprétations, il faut interpréter le compteur comme une fonction, affectant à chaque instant  $t$ , un numéro ou un rang. Le compteur  $x(t)$  est un entier qui peut être négatif, il doit alors être défini sur  $\mathbb{Z}$ .

### 2.3.3 Dateurs

**Définition 2.10.** *Un dateur  $d$  est une application monotone (croissante) de  $\mathbb{Z}$  dans  $\mathbb{R} \cup \{-\infty, +\infty\}$ . Un dateur doit vérifier :*

$$x(k + 1) \geq x(k).$$

Nous associons à chaque transition  $x_i$  (respectivement,  $u_i$  ou  $y_i$ ) un dateur  $x_i(k)$  (respectivement,  $u_i(k)$  ou  $y_i(k)$ ), qui représente la date, possiblement négative, à laquelle l'événement  $k$ , possiblement négatif, associé à chaque variable, a lieu.

## 2.4 Modèles algébriques des graphes d'événements temporisés

Le graphe d'événements temporisé est un outil de modélisation graphique qui constitue une première étape. La seconde étape est la mise en équations du modèle graphique, c'est-à-dire, la définition d'une représentation analytique du système. Le comportement d'un graphe d'événements temporisé peut être décrit par un modèle linéaire dans l'algèbre  $(max, +)$  (ou l'algèbre  $(min, +)$ ), ce modèle amène à manipuler des variables discrètes qui correspondent aux dates d'activation des transitions du GET. Notre contribution dans cette partie est de prouver qu'il est possible d'exprimer par un modèle linéaire les variables dateurs (compteurs) les uns en fonction des autres dans l'algèbre conventionnelle.

### 2.4.1 Premières inégalités en dateur d'un GET

Nous supposons, lors de l'établissement des équations d'évolution du système, que les jetons situés dans chaque place  $p_j$  à  $k = 1$  sont disponibles depuis  $-\infty$ . Cela veut dire qu'ils ont été générés par des tirs aux dates  $x_i(k) = -\infty \forall k \leq 0$ . Ils peuvent donc contribuer immédiatement aux tirs des transitions. Les tirs des transitions en évolution courante sont ainsi numérotés à

partir de 1 (le premier tir d'une transition porte le numéro  $k = 1$ ). Par convention, on considère qu'un événement  $k$  qui n'a jamais lieu est formulé mathématiquement comme suit :  $x_i(k) = +\infty$ .

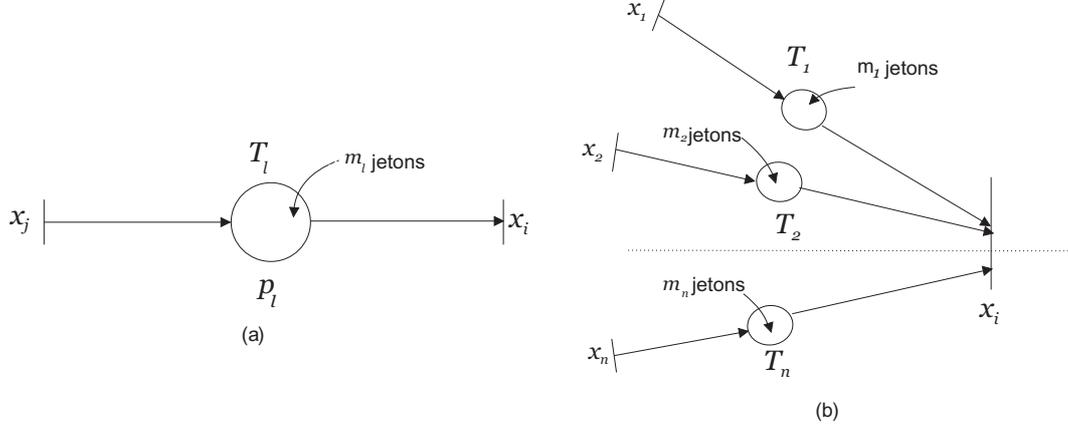


Fig. 2.6 : Parties d'un graphe d'événements temporisé.

Si nous supposons un fonctionnement FIFO des places qui garantit qu'il n'y aura pas de dépassement de jeton, une numérotation des événements peut être utilisée. L'évolution peut alors être représentée par des inégalités qui décrivent les relations entre les dates de tir. Si la constante  $T_l$  est la temporisation associée à la place  $p_l$  et  $m_l$  le nombre de jetons présents initialement dans  $p_l$ , alors le tir de  $x_i$  aura lieu au plus tôt pour la  $k$ -ième fois après le tir de  $x_j$  pour la  $(k-m_l)$ -ième fois, plus un temps de séjour minimal égal à  $T_l$ . Les deux dateurs  $x_j(k)$  et  $x_i(k)$  de la figure (2.6a) satisfont l'inéquation suivante.

Nous pouvons décrire le système à partir de chaque place  $p_l$  où  $(x_j, x_i) = (\bullet p_l, p_l \bullet)$ , soit :

$$x_i(k) \geq x_j(k - m_l) + T_l. \quad (2.2)$$

De même, pour la transition  $x_i$  de la figure (2.6b), le franchissement de  $x_i$  pour la  $k$ -ième fois est conditionné par :

- le tir de  $x_1$  pour la  $(k-m_1)$ -ième fois,
- le tir de  $x_2$  pour la  $(k-m_2)$ -ième fois,
- et ainsi de suite jusqu'au tir de  $x_n$  pour la  $(k-m_n)$ -ième fois.

En tenant compte de ces conditions, nous obtenons le système suivant :

$$\left\{ \begin{array}{l} x_i(k) \geq x_1(k - m_1) + T_1, \\ x_i(k) \geq x_2(k - m_2) + T_2, \\ \quad \quad \quad \vdots \\ x_i(k) \geq x_n(k - m_n) + T_n, \end{array} \right.$$

soit :

$$x_i(k) \geq \max(x_1(k - m_1) + T_1, x_2(k - m_2) + T_2, \dots, x_n(k - m_n) + T_n). \quad (2.3)$$

**Remarque 2.11**

Notons qu'un graphe d'événements peut être considéré comme un ensemble de sous-graphes composés d'une place  $p_l$  liant une transition amont  $\{x_j\} = \bullet p$  et une transition aval  $\{x_i\} = p \bullet$ .  $\diamond$

**2.4.2 Premières inégalités en compteur d'un GET**

Reprenons l'exemple de la figure (2.6), en considérant les mêmes conditions initiales que celles considérées pour les dateurs, on suppose un fonctionnement FIFO des places (garantissant qu'il n'y aura pas de dépassement de jeton en fonction du temps). L'évolution peut alors être représentée par des inégalités qui décrivent les relations entre le nombre de tir. Nous associons pour chaque transition  $x_j$  la variable  $x_j(t)$  qui représente le nombre de fois où la transition  $x_j$  a été activée, avant, ou à, la date  $t$ . Le compteur  $x_i(t)$  ne peut pas excéder le compteur  $x_j(t - T_j)$  car tous les jetons doivent séjourner dans la place  $p_l$ , au moins  $T_l$  unités de temps. Par conséquent, la valeur  $x_i(t)$  ne peut pas dépasser  $x_j(t - T_l) + m_j$ , du fait qu'il existe  $m_l$  jetons en réserve venant du franchissement de  $x_j$  antérieurement. Les deux compteurs  $x_j(t)$  et  $x_i(t)$  de la figure (2.6a) satisfont l'inéquation suivante :

$$x_i(t) \leq x_j(t - T_l) + m_l. \quad (2.4)$$

Avec un raisonnement similaire à celui qui a été fait dans le cas des dateurs, les transitions de la figure 2.6b obéissent au système suivant :

$$\begin{cases} x_i(t) \leq x_1(t - T_1) + m_1, \\ x_i(t) \leq x_2(t - T_2) + m_2, \\ \quad \quad \quad \vdots \\ x_i(t) \leq x_n(t - T_n) + m_n, \end{cases}$$

soit :

$$x_i(t) \leq \min(x_1(t - T_1) + m_1, x_2(t - T_2) + m_2, \dots, x_n(t - T_n) + m_n). \quad (2.5)$$

**2.4.3 Modèle d'un GET dans l'algèbre (Max,+)**

Le premier modèle, utilise le dioïde  $\overline{\mathbb{R}}_{max}$  (où  $a \oplus b = \max(a, b)$  et  $a \otimes b = a + b$ ), l'inéquation 2.3 peut se mettre sous forme d'un modèle linéaire, on obtient la forme implicite :

$$x_i(k) \geq \bigoplus_{j=1}^n x_j(k - m_j) \otimes T_j. \quad (2.6)$$

Notons, que pour l'instant, il n'est pas possible de déduire l'évolution future de la transition  $x_i$  avec l'équation 2.6 à partir des valeurs de  $x_j(k - m_j)$ . Pour résoudre cette difficulté, l'hypothèse d'un fonctionnement au plus tôt, peut être choisie au niveau de la transition  $x_i$ , ce qui peut être considéré comme une règle de comportement ou de commande locale. Sous cette hypothèse de fonctionnement, le comportement deviendra déterministe et le système 2.6 pourra se mettre sous

la forme :

$$x_i(k) = \bigoplus_{j=1}^n x_j(k - m_j) \otimes T_j. \quad (2.7)$$

#### 2.4.4 Modèle de type dateur d'un GET dans l'algèbre conventionnelle

Dans cette partie, nous présentons un modèle qui utilise les dateurs, non pas dans l'algèbre  $(\max, +)$ , mais dans l'algèbre standard. Chaque variable  $x_i(k)$  représente la date du  $k$ -ème tir de la transition  $x_i$ .

##### Forme générale dans l'algèbre standard.

L'ensemble des inégalités précédentes qui décrivent un graphe d'événements temporisé, peut être exprimé selon la forme symétrique suivante :

$$\left( G \right) \times \begin{pmatrix} x(k - \Delta) \\ x(k - \Delta + 1) \\ \dots \\ x(k - 1) \\ x(k) \end{pmatrix} \leq -T, \quad (2.8)$$

où,  $\Delta$  est la plus grande composante de  $M_0$  et le vecteur colonne  $-T$  est le vecteur des temporisations.

La matrice  $G$  est définie  $\forall (p, x) \in P \times Tr$  comme suit :  $G = [G_\Delta G_{\Delta-1} G_{\Delta-2} \dots G_1 G_0]$  avec une dimension égale à  $(|P| \times (\Delta + 1)) \cdot |Tr|$ .

Chaque place  $p_l$  correspond à une ligne  $l$  de  $G$  qui contient les poids des arcs entrants et sortants relativement à la place  $p_l$ . En particulier, on a :

1. La matrice  $G_i$  contient les poids avec le signe plus, des arcs entrant dans les places et dont le marquage initial est  $i$  pour  $i \in [0, \Delta]$ .
2. En plus des poids définis ci-dessus, la matrice  $G_0$  contient avec le signe moins, le poids de l'arc sortant de chaque place et dont le marquage initial est nul (habituellement exprimé par  $-W^-$ ).

$$G = \left( \begin{array}{c|ccc|ccc|ccc|ccc} & \overbrace{\quad x(k-\Delta) \quad} & & & & \overbrace{\quad x(k-s) \quad} & & & & \overbrace{\quad x(k) \quad} & & & & \\ & \cdot & x_i & \dots & x_j & \cdot & & & & \cdot & x_i & \dots & x_j & \cdot & \\ p_1 & 0 & 0 & \dots & 0 & 0 & \dots & & & 0 & 0 & \dots & 0 & 0 & \\ \vdots & & \vdots & & & & & & & & \vdots & & & & \\ p_l & 0 & 0 & \dots & 1 & 0 & \dots & & & 0 & -1 & \dots & 0 & 0 & \\ p_n & 0 & 0 & \dots & 0 & 0 & \dots & & & 0 & 0 & \dots & 0 & 0 & \end{array} \right),$$

où  $s$  représente le marquage des différentes places  $0 \leq s \leq \Delta$

$$G_{i \in [1, \Delta]}(p, x) = \begin{cases} +1, & \text{si } t \in \bullet p, \\ 0, & \text{sinon,} \end{cases}$$

et pour le marquage  $s = 0$

$$G_{i=0}(p, x) = \begin{cases} +1, & \text{si } x \in \bullet p, \\ -1, & \text{si } x \in p \bullet, \\ 0, & \text{sinon.} \end{cases}$$

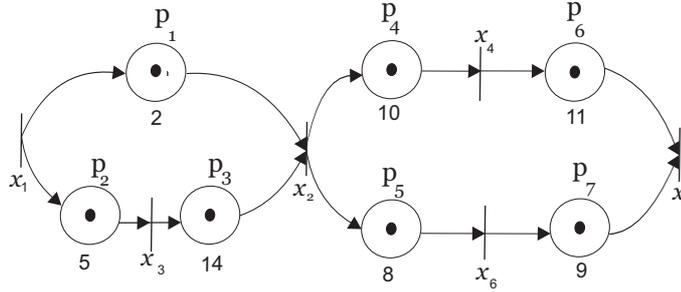
De la description ci-dessus sur le poids des arcs, nous pouvons déduire la relation suivante avec la matrice d'incidence  $W$  :

$$W = \sum_{i=0}^{\Delta} G_i.$$

La matrice  $G$  d'un graphe d'événements permet une description synthétique du système correspondant. Elle est en quelque sorte l'analogue de la matrice d'incidence arcs-sommets de la théorie des graphes.

**Exemple 2.12**

Soit le graphe d'événements temporisé suivant :



**Fig. 2.7 :** Graphe d'événements temporisé.

Le modèle algébrique est donné comme suit :

$$\begin{pmatrix} W^+ & -W^- \end{pmatrix} \cdot \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq (-T),$$

avec  $x(k) = (x_1(k) \ x_2(k) \ x_3(k) \ x_4(k) \ x_5(k) \ x_6(k))^t$ ,

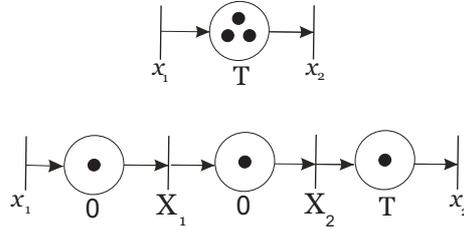
$$W^+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad W^- = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{et } T = \begin{pmatrix} 2 \\ 5 \\ 14 \\ 10 \\ 8 \\ 11 \\ 9 \end{pmatrix}.$$

On obtient la matrice  $G$  suivante :

$$G = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} x^{(k-1)} \\ \\ \\ \\ \\ \\ \\ \end{array} \left| \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} x^{(k)} \\ \\ \\ \\ \\ \\ \\ \end{array} \right. \diamond$$

### Modèle algébrique sur un horizon réduit

Afin de simplifier les expressions présentes dans ce document, le système d'inégalités (2.8) va être exprimé sur un horizon réduit  $[k-1, k]$ . La technique consiste à établir un modèle équivalent tel que chaque place du graphe contienne seulement au plus un jeton. L'idée est de dupliquer chaque place contenant plusieurs jetons en plusieurs places, voir figure 2.8, chacune contenant un seul jeton. Chaque place contenant au maximum  $\Delta$  jetons, cette opération se fera au maximum  $\Delta$  fois.



**Fig. 2.8** : Technique utilisée pour dupliquer une place.

La procédure est alors d'introduire une nouvelle variable  $X$ , définie comme suit :

$X(k) = \left( X_0(k) \quad \dots \quad X_i(k) \quad \dots \quad X_{\Delta-1}(k) \right)^t$  avec  $X_i(k) = x(k - \Delta + i + 1)$ . Par construction, on obtient :  $X_{\Delta-1}(k) = x(k)$  et  $X_i(k) = X_{i+1}(k-1)$  pour  $i \in [0, \Delta - 2]$ .

Donc, le système (2.8) devient :

$$\begin{pmatrix} G'_1 & G'_0 \end{pmatrix} \times \begin{pmatrix} X(k-1) \\ X(k) \end{pmatrix} \leq -T,$$

où :

$$G'_1 = \begin{pmatrix} G_\Delta & 0 & \dots & \dots & 0 \end{pmatrix} \text{ et } G'_0 = \begin{pmatrix} G_{\Delta-1} & G_{\Delta-2} & \dots & G_1 & G_0 \end{pmatrix}.$$

Le fait de rajouter des nouvelles places, avec une temporisation nulle fait qu'il est nécessaire de compléter le système avec les  $(\Delta - 1) \times |Tr|$  relations suivantes :

$$X_i(k) - X_{i+1}(k-1) \leq 0,$$

pour  $i = 0$  à  $\Delta - 2$ .

Alors on peut écrire le système de la manière suivante :

$$\begin{pmatrix} V_{01} & V_{00} \end{pmatrix} \times \begin{pmatrix} X(k-1) \\ X(k) \end{pmatrix} \leq 0,$$

où la matrice  $V_{00}$  est la matrice identité tandis que  $V_{01}$  est une matrice de dimension  $((m - 1) \times |Tr|. \Delta)$  composée d'une sur-diagonale de coefficients  $-1$ .

Finalement, la concaténation des deux systèmes donne la forme algébrique suivante :

$$\begin{pmatrix} H \end{pmatrix} \times \begin{pmatrix} X(k-1) \\ X(k) \end{pmatrix} \leq \begin{pmatrix} -T \\ 0 \end{pmatrix}, \quad (2.9)$$

avec :

$$H = \begin{pmatrix} G'_1 & G'_0 \\ V_{01} & V_{00} \end{pmatrix}.$$

#### 2.4.5 Modèle de GET dans l'algèbre $(\text{Min}, +)$

Dans le dioïde  $\overline{\mathbb{Z}}_{min}$  (où  $a \wedge b = \min(a, b)$  et  $a \otimes b = a + b$ ), l'inéquation 2.5 suit un modèle linéaire de la forme implicite :

$$x_i(t) \leq \bigwedge_{j=1}^n x_j(t - T_j) \otimes m_j. \quad (2.10)$$

Si nous considérons un fonctionnement au plus tôt du GET, le système 2.10 se met sous la forme :

$$x_i(t) = \bigwedge_{j=1}^n x_j(t - T_j) \otimes m_j. \quad (2.11)$$

#### 2.4.6 Modèle de type compteur de GET dans l'algèbre conventionnelle

En utilisant le compteur  $x_i(t)$  qui représente le nombre d'activations de la transition  $x_i$  survenues avant ou à la date  $t$ . A partir de l'équation 2.4

$$-x_j(t - T_l) + x_i(t) \leq +m_l. \quad (2.12)$$

Le coefficient 1 de  $x_i(t)$  (respectivement, 1 de  $x_j(t - T_l)$ ) est le poids en valeur absolue de l'arc allant de la place  $p_l$  vers la transition  $x_i$  (respectivement, l'arc allant de la transition  $x_j$  vers la place  $p_l$ ).

##### Forme générale dans l'algèbre standard.

Soit  $M_0$  le vecteur colonne du marquage initial, l'ensemble des inégalités 2.12, peut être exprimé avec la forme suivante :

$$F \times \begin{pmatrix} x(t - \Delta_T) \\ x(t - \Delta_T + 1) \\ \vdots \\ x(t) \end{pmatrix} \leq M_0, \quad (2.13)$$

où  $\Delta_T$  représente la temporisation maximale des places du GET.

La matrice  $F$  est définie  $\forall (p, x) \in P \times Tr$  comme suit :  $F = (F_{\Delta_T}, F_{\Delta_T-1}, \dots, F_0)$ .

Chaque place  $p_l$  correspond à une ligne  $l$  de la matrice  $F$  qui contient les poids de ses arcs entrants et sortants. En particulier, on a :

1. La matrice  $F_i$  contient le poids avec le signe moins, de l'arc entrant de chaque place dont la transition  $x_i$  est située en amont de la place  $p_l$ .
2. En plus des poids définis ci-dessus, la matrice  $F_0$  contient les poids, avec le signe plus, des arcs sortant des places dont la transition  $x_i$  est située en aval de la place  $p_l$ .

Comme la matrice  $G$ , la matrice  $F$  d'un graphe d'événements permet une description synthétique d'un système. Elle est en quelque sorte l'analogue de la matrice d'incidence arcs-sommets en théorie des graphes.

## 2.5 Graphes d'événements temporels

Les graphes d'événements temporels sont une sous-classe des réseaux de Petri temporels, introduits par Merlin [Merlin 1974]. Les GE temporels, l'extension temporelle s'exprime sous la forme d'un intervalle associé principalement aux transitions (t-temporel), ou aux places (p-temporel). En ce qui concerne l'expressivité des graphes d'événements p-temporels et t-temporels, Khansa a montré [Khansa 1996] que ces deux modèles sont distinctes.

### 2.5.1 Graphes d'événements P-temporels

Développés dans [Khansa 1996] et [Khansa 1997], les réseaux de Petri p-temporels ont pour objectif de modéliser des systèmes dans lesquels le temps prend un aspect non déterministe. En effet, le temps de séjour associé à chaque place  $p_i$  n'est plus considéré déterminé comme pour les GET, mais appartient à un intervalle de temps  $IS_i$ . Nous consacrons la première partie de cette section au rappel des définitions et notations relatives aux réseaux de Petri p-temporels. Comme domaines d'applications, on peut citer les lignes de galvanoplastie, les chaînes alimentaires, les manufactures comportant un four, les traitements thermiques, etc.

**Définition 2.13.** *Un graphe d'événements p-temporel est défini par le doublet  $\langle R, IS \rangle$  où :  $R$  est un réseau de Petri marqué (places-transitions)*

$IS : P \longrightarrow (\mathbb{R}^+ \cup \{0\}) \times (\mathbb{R}^+ \cup \{0, +\infty\}),$   
 $p_i \longrightarrow IS_i = [a_i, b_i] \text{ avec } 0 \leq a_i \leq b_i,$   
 où  $[a_i, b_i]$  définit l'intervalle statique de temps de séjour d'une marque dans la place  $p_i$ .

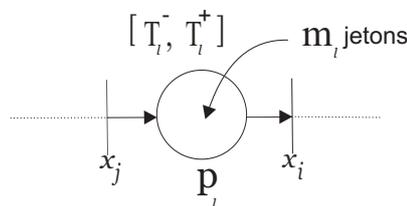
Le temps de séjour d'un jeton est compris dans un intervalle de temps. Le jeton qui arrive dans une place est indisponible (il ne peut pas permettre le franchissement de la transition située en aval) pendant un certain temps. Il doit passer un temps de séjour minimum correspondant à la borne minimale de l'intervalle. Le jeton atteint un âge de maturité (qui lui permet de franchir une transition) après  $a_i$  unités de temps après son arrivée dans la place. Il reste dans cet état de disponibilité durant  $b_i - a_i$  unités de temps. Une des spécificités des réseaux de Petri p-temporels est la possibilité de la mort de jetons, le mode de fonctionnement est le suivant. Après un séjour de  $b_i$  (borne max de l'intervalle) unités de temps dans la place, le jeton se trouve dans l'obligation de quitter cette dernière, sinon, il se retrouve dans un état de mort. Autrement dit, le jeton ne peut plus participer aux franchissements, et cela peut générer d'éventuels dysfonctionnements du système dans le futur si, par exemple, le jeton représente une ressource importante. Par conséquent, ce modèle va nécessiter un contrôle des tirs des transitions afin d'éviter la mort des jetons.

**Définition 2.14. (Fonctionnement admissible)**

Un fonctionnement est admissible pour un réseau de Petri p-temporel, lorsque son évolution dynamique préserve la vivacité des jetons. En conséquence, il ne mène pas le système à un état de blocage.

### 2.5.2 Premières inéquations d'un graphe d'événements P-temporel

De la même manière que pour les graphes d'événements temporisés, nous pouvons décrire le système d'inégalités pour les graphes d'événements p-temporels, à partir de chaque place  $p_l$  où  $(x_j, x_i) = (\bullet p_l, p_l^\bullet)$ .



**Fig. 2.9 :** Graphe d'événements p-temporel.

Les places sont supposées fonctionner en FIFO. L'évolution peut être alors représentée par les inégalités suivantes lesquelles décrivent les relation entre les dates de tir.

Pour la borne inférieure  $T_l^-$ , on a :

$$T_l^- + x_j(k - m_l) \leq x_i(k). \quad (2.14)$$

Le coefficient 1 de  $x_j(k - m_l)$  (respectivement, 1 de  $x_i(k)$ ) est le poids en valeur absolue de l'arc allant de la transition  $x_j$  vers la place  $p_l$  (respectivement, l'arc allant de la place  $p_l$  vers la transition  $x_i$ ).

De même en utilisant la borne supérieure  $T_l^+$ , nous avons :

$$x_i(k) \leq T_l^+ + x_j(k - m_l). \quad (2.15)$$

Si  $x_i$  est une transition de synchronisation (voir la figure 2.10b),  $x_i(k)$  doit satisfaire les deux systèmes d'inéquations suivants :

$$\left\{ \begin{array}{l} x_i(k) \geq x_1(k - m_1) + T_1^- \\ x_i(k) \geq x_2(k - m_2) + T_2^- \\ \vdots \\ x_i(k) \geq x_n(k - m_n) + T_n^- \end{array} \right. \text{ et } \left\{ \begin{array}{l} x_i(k) \leq x_1(k - m_1) + T_1^+ \\ x_i(k) \leq x_2(k - m_2) + T_2^+ \\ \vdots \\ x_i(k) \leq x_n(k - m_n) + T_n^+ \end{array} \right. .$$

### 2.5.3 Modèles de graphe d'événements P-temporels

#### 2.5.3.1 Modèle de GE P-temporels dans l'algèbre (Max,+)

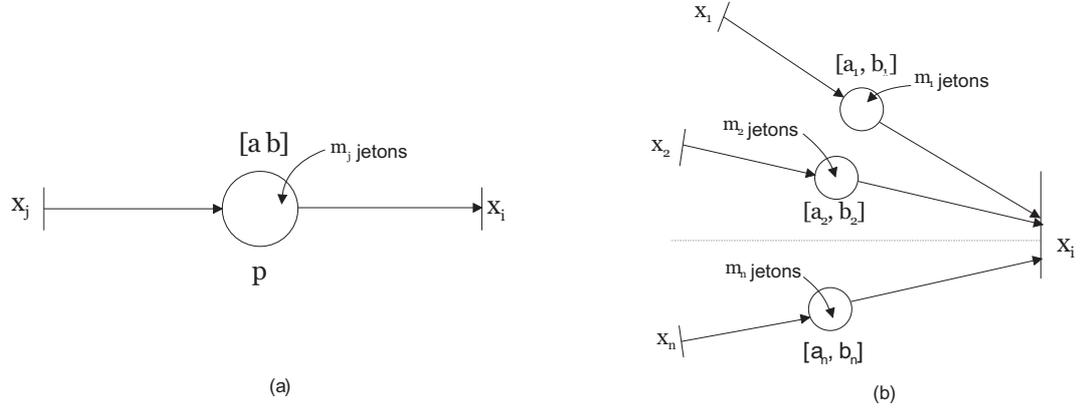
Dans l'algèbre des dioïdes, nous adoptons les notations suivantes :  $\oplus$  opérateur maximum,  $\wedge$  opérateur minimum et  $\otimes$  addition dans  $\overline{\mathbb{R}}_{max}$ .

La proposition suivante donne les conditions que doivent satisfaire les dates de tirs des transitions pour garder un fonctionnement admissible. Ces conditions se présentent sous la forme d'un modèle d'intervalle.

**Proposition 2.15.** *Pour qu'un graphe d'événements p-temporel respecte un fonctionnement admissible, la modélisation correspondant à chacune de ses transitions, non transition source, est donnée par l'expression suivante :*

$$\bigoplus_{p_j \in \bullet x_i} (x_j(k - m_j) + T_j^-) \leq x_i(k) \leq \bigwedge_{p_j \in \bullet x_i} (x_j(k - m_j) + T_j^+), \quad (2.16)$$

avec  $\bullet x_i$  l'ensemble des places d'entrée de la transition  $x_i$  et  $m_j$  le marquage initial de chacune de ces places.



**Fig. 2.10** : Parties d'un graphe d'événements p-temporel.

Une autre reformulation du modèle (2.16) a été proposée dans [Didi-Alaoui 2005]. L'objectif est de trouver une nouvelle forme du modèle représentant le fonctionnement d'un graphe d'événements p-temporel. Cette nouvelle forme ne comporte que des termes  $(\max, +)$ . Nous adoptons ici les notations suivantes :

$\leftarrow x_i = \{x_j \text{ tel que } x_j \in \bullet(\bullet x_i)\}$  est l'ensemble des transitions  $x_j$  qui se trouvent en amont des places situées en amont de la transition  $x_i$ .

$x_i^{\rightarrow} = \{x_j \text{ tel que } x_j \in (x_i^{\bullet})\bullet\}$  est l'ensemble des transitions  $x_j$  qui se situent en aval des places situées en aval de la transition  $x_i$ .

$m_{ij}$  (respectivement,  $m_{ji}$ ) le marquage de la place entre  $x_j$  et  $x_i$ , à laquelle est associé l'intervalle de temporisation  $[a_{ij}, b_{ij}]$  avec  $x_j \in \leftarrow x_i$  (respectivement,  $x_j \in x_i^{\rightarrow}$ ).

D'après le système 2.16, on en déduit que :

$$\forall x_j \in \leftarrow x_i, x_i(k) \geq x_j(k - m_{ij}) + a_{ij} \text{ et } \forall x_j \in \leftarrow x_i, x_i(k) \leq x_j(k - m_{ij}) + b_{ij} \iff \\ \forall x_j \in x_i^{\rightarrow}, x_j(k + m_{ji}) - b_{ji} \leq x_i(k).$$

Par conséquent, on obtient le modèle suivant décrit uniquement par des fonctions  $(\max, +)$  :

$$\begin{cases} x_i(k) \geq \bigoplus_{x_j \in \leftarrow x_i} x_j(k - m_{ij}) \otimes a_{ij}, \\ x_i(k) \geq \bigoplus_{x_j \in x_i^{\rightarrow}} x_j(k - m_{ji}) \otimes (-b_{ji}), \end{cases}$$

soit

$$x_i(k) \geq \bigoplus_{x_j \in \leftarrow x_i} a_{ij}^- \otimes x_j(k - m_{ij}) \oplus \bigoplus_{x_j \in x_i^{\rightarrow}} a_{ij}^+ \otimes x_j(k + m_{ji}), \quad (2.17)$$

avec  $a_{ij}^- = a_{ij}$  et  $a_{ij}^+ = -b_{ji}$ ,  $a_{ij}^- \in \mathbb{R}^+$ ,  $a_{ij}^+ \in \mathbb{R}^-$ .

Ce modèle uniquement de type  $(\max, +)$  donné par 2.17 est équivalent au modèle d'intervalle initial de type  $((\max, +), (\min, +))$  2.16 est qui représenté par le système suivant :

$$\bigoplus_{j=1}^n a_j \otimes x_j(k - m_{ij}) \leq x_i(k) \leq \bigwedge_{j=1}^n T_j^+ \otimes x_j(k - m_{ij}).$$

Dans le dioïde  $\overline{\mathbb{R}}_{\max}$ , le premier système est :  $x_i(k) \geq \bigoplus_{j=1}^n T_j^- \otimes x_j(k - m_{ij})$ ,

dans le dioïde  $\overline{\mathbb{Z}}_{\min}$ , le deuxième système est :  $x_i(k) \leq \bigwedge_{j=1}^n T_j^+ \otimes x_j(k - m_{ij})$ .

### 2.5.3.2 Modèle de type dateur de GE P-temporels dans l'algèbre conventionnelle

Rappelons que  $\Delta$  est la plus grande composante de  $M_0$ . Ci-dessous, les deux vecteurs colonne  $-T^-$  et  $T^+$  sont des vecteurs de temporisations où  $[T_l^-, T_l^+]$  est l'intervalle de temps de la place  $p_l$ .

L'ensemble des inégalités précédentes (2.14) et (2.15), lesquelles décrivent un graphe d'événements P-temporel, peut être exprimé avec la forme matricielle suivante :

$$\begin{pmatrix} G^- \\ G^+ \end{pmatrix} \cdot \begin{pmatrix} x(k - \Delta) \\ x(k - \Delta + 1) \\ \vdots \\ x(k - 1) \\ x(k) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}, \quad (2.18)$$

avec :

$$G^- = [G_{\Delta}^- \ G_{\Delta-1}^- \ G_{\Delta-2}^- \ \cdots \ G_1^- \ G_0^-] \text{ et } G^+ = [G_{\Delta}^+ \ G_{\Delta-1}^+ \ G_{\Delta-2}^+ \ \cdots \ G_1^+ \ G_0^+].$$

Chaque place correspond à une ligne de  $G^-$  qui contient les poids, de ses arcs entrants et sortants. En particulier, on a :

1. La matrice  $G_i^-$  contient les poids, avec le signe plus, des arcs entrant des places dont le marquage initial est  $i$ , pour  $i \in [0, \Delta]$ .
2. De plus, la matrice  $G_0^-$  contient les poids, avec le signe moins, de l'arc sortant de chaque place dont le marquage initial est nul (habituellement exprimé par  $-W^-$ ).

On a donc  $G_i^- \geq 0$  pour  $i \in [1, \Delta]$  et  $G_0^- \geq -W^-$ . Remarquons qu'un coefficient non nul de la matrice  $G_0^-$  peut être négatif ou positif.

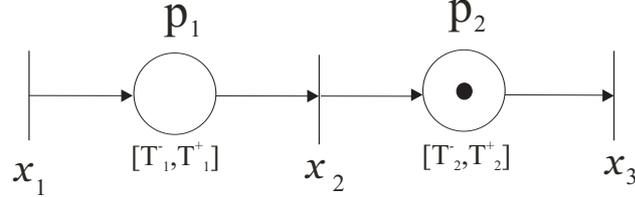
De la description ci-dessus sur le poids des arcs, nous pouvons déduire la relation suivante avec la matrice d'incidence  $W$  :

$$W = \sum_{i=0}^{i=\Delta} G_i.$$

Notons que chaque place présentant une temporisation supérieure égale à l'infinie correspond à une contrainte  $x_i(k) - x_j(k - m_l) \leq T_l^+$  avec  $T_l^+ = +\infty$ . N'ayant pas d'effet sur l'évolution du système, chacune de ces contraintes peut donc être retirée. Une autre possibilité est de supposer que chaque borne infinie est remplacée par une grandeur finie arbitrairement grande, ce qui neutralise également la contrainte. Afin de présenter simplement le modèle, nous choisissons cette deuxième possibilité. Ainsi, on a  $G^+ = -G^-$ , et l'interprétation de la matrice  $G^+$  est la

même que  $G^-$  au changement de signe près des coefficients. La dimension de  $G^-$  et  $G^+$  est égale à  $|P| \cdot (m+1) \cdot |Tr|$ .

**Exemple 2.16** Soit le graphe d'événements P-temporel dans la Figure (2.11). L'ensemble des



**Fig. 2.11** : Graphe d'événements P-temporel.

inégalités qui décrivent ce graphe est exprimé par le système d'inégalités suivant :

$$\begin{cases} x_1(k) - x_2(k) \leq -T_1^-, \\ x_2(k-1) - x_3(k-1) \leq -T_2^-, \\ -x_1(k) + x_2(k) \leq T_1^+, \\ -x_2(k-1) + x_3(k) \leq T_2^+, \end{cases}$$

la forme matricielle correspondante est comme suit :

$$\begin{pmatrix} 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} \leq \begin{pmatrix} -T_1^- \\ -T_2^- \\ T_1^+ \\ T_2^+ \end{pmatrix}.$$

Les matrices sont donc :  $G^- = [G_1^- G_0^-]$  et  $G^+ = [G_1^+ G_0^+]$ ,

$$\text{où } G_1^- = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, G_0^- = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, G_1^+ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \text{ et } G_0^+ = \begin{pmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

$$\text{Notons que } -W^- = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

◇

### Modèle algébrique sur un horizon réduit

Afin de simplifier les expressions présentées dans ce document, le système d'inégalités (2.18) va être exprimé sur un horizon réduit. La technique est d'établir un modèle équivalent tel que chaque place du graphe contienne au plus un, ou zéro, jeton. L'idée est de dupliquer chaque place en plusieurs places, chacune contenant un seul jeton. Chaque place contenant au maximum  $\Delta$

jetons, cette opération se fera au maximum  $\Delta$  fois. La procédure est alors d'introduire la nouvelle variable  $X$ , définie comme suit :

$X(k) = \left( X_0(k) \quad \dots \quad X_i(k) \quad \dots \quad X_{\Delta-1}(k) \right)^t$  avec  $X_i(k) = x(k - \Delta + i + 1)$ . Par construction, on obtient :  $X_{\Delta-1}(k) = x(k)$  et  $X_i(k) = X_{i+1}(k - 1)$  pour  $i \in [0, \Delta - 2]$ .

Donc, le système (2.18) devient :

$$\begin{pmatrix} G'_1 & G'_0 \\ -G'_1 & -G'_0 \end{pmatrix} \times \begin{pmatrix} X(k-1) \\ X(k) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix},$$

où :

$$G'_1 = \begin{pmatrix} G_{\Delta} & 0 & \dots & \dots & 0 \end{pmatrix} \text{ et } G'_0 = \begin{pmatrix} G_{\Delta-1} & G_{\Delta-2} & \dots & G_1 & G_0 \end{pmatrix}.$$

Il est nécessaire de compléter le système avec  $2(\Delta - 1) \times |Tr|$  relations :

$$\begin{cases} X_i(k) - X_{i+1}(k-1) \leq 0, \\ -X_i(k) + X_{i+1}(k-1) \leq 0, \end{cases}$$

pour  $i = 0$  à  $\Delta - 2$ .

On peut alors écrire le système de la manière suivante :

$$\begin{pmatrix} V_{01} & V_{00} \\ -V_{01} & -V_{00} \end{pmatrix} \times \begin{pmatrix} X(k-1) \\ X(k) \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

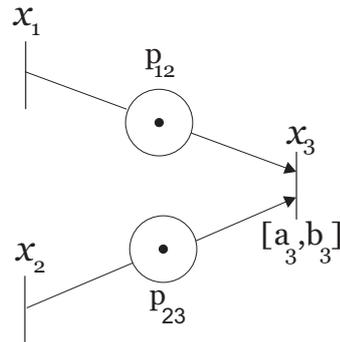
où la matrice  $V_{00}$  est la matrice identité, tandis que  $V_{01}$  est une matrice de dimension  $((\Delta - 1) \times |Tr|, \Delta)$  composée d'une sur-diagonale de coefficients  $-1$ . Finalement, la concaténation des deux systèmes donne la forme algébrique suivante :

$$\begin{pmatrix} H \\ -H \end{pmatrix} \times \begin{pmatrix} X(k-1) \\ X(k) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ 0 \\ T^+ \\ 0 \end{pmatrix}, \quad (2.19)$$

$$\text{où } H = \begin{pmatrix} G'_1 & G'_0 \\ V_{01} & V_{00} \end{pmatrix}.$$

#### 2.5.4 Graphes d'événements T-temporels

Dans les graphes d'événements T-temporels, les activités sont liées aux transitions. Ils sont capables de représenter par exemple des systèmes informatiques ayant des temps de réponse variables [Berthomieu 1991]. Un autre problème pratique pour lequel les graphes d'événements t-temporels sont un outil de modélisation intéressant est le système du "chien de garde", ou alarme. Un des objectifs de ce système est de surveiller la date de production d'une action. Suivant cette date et un paramètre donné, l'alarme se déclenche, ou se désactive [Boyer 2001]. La figure 2.12 donne un exemple d'un graphe d'événements t-temporel.



**Fig. 2.12** : Graphe d'événements t-temporel.

**Définition 2.17.** *Un graphe d'événements t-temporel est un doublet  $\langle \text{GE}, IS_x \rangle$  où GE est un graphe d'événements*

$$\begin{cases} IS_x : T \longrightarrow (\mathbb{R}^+ \cup \{0\}) \times (\mathbb{R}^+ \cup \{0, +\infty\}), \\ x_i \longrightarrow IS_i = [a_i, b_i] \text{ avec } 0 \leq a_i \leq b_i. \end{cases}$$

*Chaque transition  $x_i$  est associée à un intervalle statique  $[a_i, b_i]$ , les durées  $a_i, b_i$  représentent respectivement les dates de tir au plus tôt et au plus tard de la transition  $x_i$ .*

Le tir des transitions présente un aspect logique et un aspect temporel. L'intervalle de tir d'une transition est relatif à la date où la transition devient sensibilisée, c'est-à-dire, que le nombre de jetons situés dans les places en amont de la transition est suffisant : il n'y a donc pas de notion de temps à ce niveau. Pour qu'une transition  $x_i$  soit franchie, il faut qu'elle reste sensibilisée durant au moins  $a_i$  unités de temps et au plus  $b_i$  unités de temps. Si, par exemple, une transition  $x_j$  est validée à l'instant  $\tau$ , sachant que  $[a_j, b_j]$  est l'intervalle statique associé à  $x_j$ . Alors,  $x_j$  peut être tirée entre  $\tau + a_j$  et  $\tau + b_j$ .

**Exemple 2.18** (Un système producteur-consommateur simple)

Le système se compose de deux nœuds, **un producteur** et **un consommateur**. **Le producteur** prend 1 à 7 unités de temps pour produire un produit, puis prend de 1 à 5 unités de temps pour déposer le produit pour le consommateur, puis répétez le processus de production. **Le consommateur** prend de 2 à 6 unités de temps pour entrer le produit, 2 à 4 unités de temps pour le consommer, et puis attend un nouveau produit. La figure 2.13 montre le GE T-temporel d'un système producteur-consommateur simple. Le tableau suivant donne la description des places et des transitions pour un seul produit.

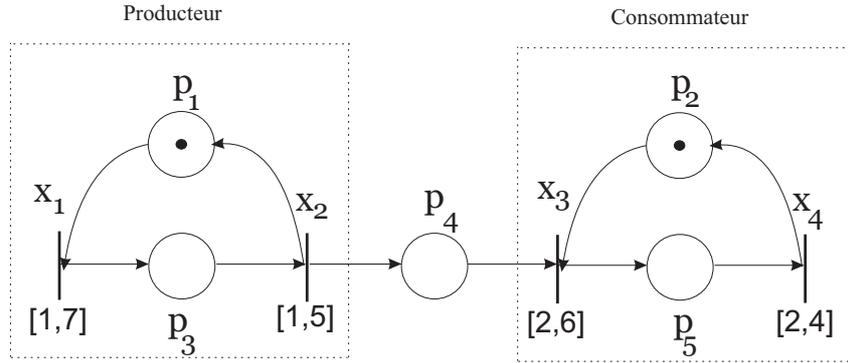


Fig. 2.13 : GE T-temporel d'un simple système producteur-consommateur.

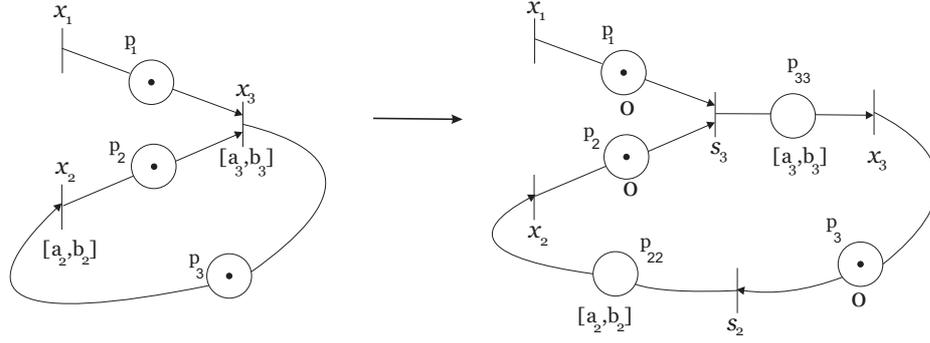
Place	Description de l'état
$p_1$	Le producteur est disponible ou en cours de production ( $x_1$ est sensibilisée).
$p_3$	Le produit est prêt pour être déposé ou en phase de dépôt ( $x_3$ est sensibilisée).
$p_2$	Le consommateur est disponible ou en cours de consommation.
$p_5$	Le consommateur est en cours de récupération du produit.
$p_4$	Le produit est disponible pour le consommateur.
Transition	Description de la tâche
$x_1$	Le producteur produit le produit.
$x_2$	Le producteur dépose le produit.
$x_3$	Le consommateur récupère le produit.
$x_4$	Le consommateur consomme le produit.

◇

#### 2.5.4.1 Modèle des GE T-temporels dans l'algèbre (Max,+)

Nous étudions maintenant le cas d'une transition de synchronisation comme illustrée dans la figure 2.12. Nous constatons tout d'abord que le déclenchement de l'activation de l'horloge (on dit aussi la condition temporelle, c'est-à-dire, la date du commencement de la sensibilisation du  $x_3$ ) pour la transition  $x_3$  est lié directement à la condition logique (il traduit la condition qui doit être satisfaite au niveau du marquage). Par conséquent, c'est le dernier jeton arrivant dans l'une des places ( $p_{12}$  et  $p_{23}$ ) qui marquera le début de la sensibilisation de la transition  $x_3$ . A partir de ce moment, la sensibilisation doit être maintenue continuellement pendant  $a_3$  unités de temps, et cela peut aller jusqu'à  $b_3$  unités de temps. Le franchissement peut avoir lieu entre deux dates représentant la date de tir au plus tôt, et la date de tir au plus tard. Ceci nous amène au système d'inéquations suivant pour l'exemple de la figure 2.12 :

$$\begin{cases} (x_1(k-1) + a_3) \oplus (x_2(k-1) + a_3) \leq x_3(k) \\ x_3(k) \leq (x_1(k-1) + b_3) \oplus (x_2(k-1) + b_3) \end{cases} \quad (2.20)$$



**Fig. 2.14 :** Transformation du GE T-temporel en un nouveau GE temporel.

Plus généralement, le modèle d'état correspondant est le suivant :

$$\bigoplus_{j=1}^n a_j \otimes x_j(k-1) \leq x_i(k) \leq \bigoplus_{j=1}^n b_j \otimes x_j(k-1). \quad (2.21)$$

Avec  $n$  le nombre des transitions en amont avec la transition  $x_i$ . Ce modèle est aussi présenté dans la thèse de Didi-Alaoui [Didi-Alaoui 2005].

#### 2.5.4.2 Modèle relaxé de type dateur des GE T-temporels dans l'algèbre conventionnelle

L'objectif de cette partie est d'obtenir une écriture algébrique de la forme  $Ax \leq b$  du GE T-temporels comme celle des GE P-temporels. Dans l'algèbre conventionnelle, le fonctionnement de graphe de la figure 2.12 se traduit comme suit :

$$a_3 + \underset{i \in \bullet(\bullet x_3)}{\text{Max}} x_i(k-1) \leq x_3(k) \leq b_3 + \underset{i \in \bullet(\bullet x_3)}{\text{Max}} x_i(k-1). \quad (2.22)$$

avec  $\bullet(\bullet x_3) = \{x_1, x_2\}$

Sous forme générale, on obtient :

$$a_i + \underset{j \in \bullet(\bullet x_i)}{\text{Max}} x_j(k - m_j) \leq x_i(k) \leq b_i + \underset{j \in \bullet(\bullet x_i)}{\text{Max}} x_j(k - m_j). \quad (2.23)$$

avec  $m_j$  est le marquage de la place  $p_j$ .

A partir de l'écriture (2.23), on remarque que nous ne pouvons pas déduire l'écriture polyédrale de la forme  $Ax \leq b$  pour les GE T-temporels comme le cas des GET et P-temporels à partir des inégalités en dateur.

Pour cela, on présente une technique qui permet d'écrire une forme polyédrale relaxée du GE T-temporel en utilisant la date logique représentée par la transition  $s_i$  qui représente la date du commencement de la sensibilisation de la transition  $x_i$ . Cette technique utilise une transformation du GE T-temporel en un GE particulier en rajoutant une place  $p_{ii}$  entre la transition  $s_i$  et  $x_i$

qui représente la condition temporelle. La figure 2.14 montre cette transformation. Ce nouveau graphe contient les places  $p_1, p_2$  et  $p_3$  avec une temporisation nulle correspondant à un GE temporisé,  $p_{22}$  et  $p_{33}$  correspondant à des places de GE p-temporel.

Dans notre nouveau GE, on a deux transitions logiques  $s_2$  et  $s_3$ , et deux places  $p_{33}$  et  $p_{22}$  qui représentent la condition temporelle. L'évolution du graphe de la figure (2.14) est donnée comme suit :

$$\left\{ \begin{array}{l} s_3(k) = \max(x_1(k-1), x_2(k-1)) \\ s_2(k) = x_3(k-1) \\ a_3 + s_3(k) \leq x_3(k) \leq b_3 + s_3(k) \\ a_2 + s_2(k) \leq x_2(k) \leq b_2 + s_2(k) \end{array} \right. .$$

On obtient donc un nouveau système dont les places sont celles :

- d'un GET fonctionnant au plus tôt pour la date de sensibilisation.
- d'un GE p-temporel mais sans mort de jetons.

Réécrivons le système sous la forme  $Ax \leq b$ . Une forme relaxé est la suivante :

$$\left\{ \begin{array}{l} s_3(k) \geq x_1(k-1) \\ s_3(k) \geq x_2(k-1) \\ s_2(k) \geq x_3(k-1) \\ -x_3(k) + s_3(k) \leq -a_3 \quad , \\ x_3(k) - s_3(k) \leq b_3 \\ -x_2(k) + s_2(k) \leq -a_2 \\ x_2(k) - s_2(k) \leq b_2 \end{array} \right.$$

on a :

$$\left\{ \begin{array}{l} x_1(k-1) - s_3(k) \leq 0 \\ x_2(k-1) - s_3(k) \leq 0 \\ x_3(k-1) - s_2(k) \leq 0 \\ -x_2(k) + s_2(k) \leq -a_2 \quad , \\ -x_3(k) + s_3(k) \leq -a_3 \\ x_2(k) - s_2(k) \leq b_2 \\ x_3(k) - s_3(k) \leq b_3 \end{array} \right.$$

alors l'écriture polyédrale est comme suit :

$$\left( \begin{array}{c} \underbrace{x^{(k-1)}} \\ x_1 \quad x_2 \quad x_3 \\ \underbrace{x^{(k)}} \\ x_1 \quad x_2 \quad x_3 \\ \underbrace{s^{(k)}} \\ s_1 \quad s_2 \quad s_3 \\ p_1 \quad 1 \quad 0 \quad 1 \\ p_2 \quad 0 \quad 1 \quad 0 \quad 1 \\ p_3 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \\ p_{11} \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \\ p_{22} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \\ p_{11} \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0 \quad 0 \\ p_{22} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad -1 \quad 0 \end{array} \right) \cdot \begin{pmatrix} x^{(k-1)} \\ x^{(k)} \\ s^{(k)} \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ -a_2 \\ -a_3 \\ b_2 \\ b_3 \end{pmatrix}. \quad (2.24)$$

Pour une forme plus générale, on obtient :

$$A \cdot \begin{pmatrix} x^{(k-\Delta)} \\ x^{(k-\Delta-1)} \\ \vdots \\ x^{(k)} \\ s^{(k)} \end{pmatrix} \leq \begin{pmatrix} T_{s \bullet} \\ -T_{\bullet x}^- \\ T_{\bullet x}^+ \end{pmatrix}. \quad (2.25)$$

Rappelons que  $\Delta$  est la plus grande composante de  $M_0$ . Les deux vecteurs colonne  $-T_{\bullet x}^-$  et  $T_{\bullet x}^+$  sont des vecteurs de temporisations, où  $[-T_{\bullet x_i}^-, T_{\bullet x_i}^+]$  est l'intervalle de temps de la nouvelle place  $p_{ii}$  située en amont de la transition  $x_i$ . Le vecteur  $T_{s \bullet}$  est le vecteur colonne de temporisations des places situées en aval de la transition de logique  $s_i$ . Ce vecteur est égal au vecteur nul. On obtient au final une écriture relaxée de la forme  $A \cdot x \leq b$  pour le GE T-temporel. Dans cette partie, nous avons présentés un lien entre les modèles p-temporels et t-temporels et non une équivalence entre ces deux modèles. Cette étude est une perspective qui sera développée dans de futures travaux.

## 2.6 Synthèse et comparative des modèles des graphes d'événements temporisés et temporels

Dans la thèse de Didi-Alaoui [Didi-Alaoui 2005], une étude propose des comparaisons formelles entre plusieurs modèle de RdPs temporels<sup>3</sup> utilisés pour représenter ces exemples. Cette étude est basée sur l'analyse de leurs qualités attendues. Une étude comparative de différents modèles de RdPs est aussi proposée dans [Boyer 2001] sous un point de vue pratique.

3. Le terme temporels ici recouvre les RdPs temporels et temporisés.

Ci-dessous, nous nous basons sur l'écriture algébrique dans l'algèbre conventionnelle des différentes sous classes des RdPs présentées au long de ce chapitre pour réaliser une classification.

Si nous reprenons les modèles mathématiques obtenus lors de la modélisation de différents graphes d'événements de type intervalle, nous pouvons constater qu'ils ont la forme commune suivante :

$$f^-(x) \leq x \leq f^+(x), \quad (2.26)$$

avec  $f^-$  et  $f^+$  deux fonctions définies sur  $\mathbb{R}$ , ces deux fonctions sont de deux types, soit *max* (pour signifier le maximum) ou *min* (pour signifier le minimum). L'introduction du temps sur les graphes d'événements a généré plusieurs sous-classes, que l'on peut scinder en deux sous-grandes familles selon que le comportement soit borné par dessus et dessous ou uniquement par dessous. Les modèles dont l'évolution est bornée par valeur inférieure, sont en général analysés et exploités dans un fonctionnement déterministe en raison d'un choix de fonctionnement au plus tôt local. Ainsi, pour la forme dateur, les graphes d'événements temporisés présentent un modèle du type  $f(x) \leq x$  et l'égalité  $f(x) = x$  vient de l'hypothèse du fonctionnement au plus tôt : c'est généralement ce dernier modèle avec un fonctionnement déterministe qui est utilisé dans la majorité des travaux réalisés jusqu'à présent.

Le tableau de la figure 2.15 donne une classification de tous les modèles vus dans ce chapitre à partir du type de fonctions  $f^-()$  et  $f^+()$  :

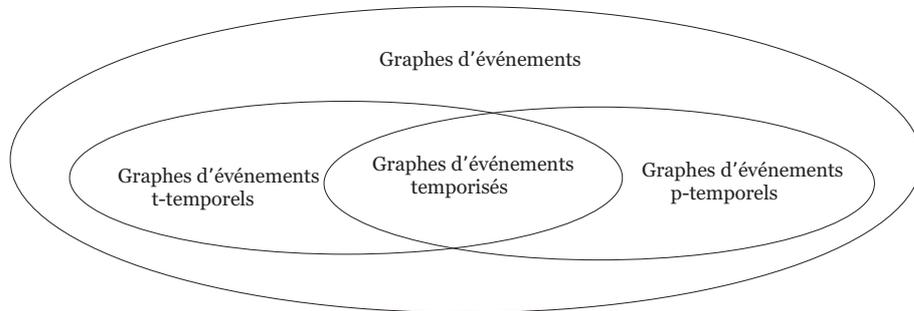
		$f^+(x)$ est de type		
		$f^+(x) = +\infty$	(Min ,+)	(Max ,+)
$f^-(x)$ est de type	$f^-(x) = -\infty$	Pas de contraintes	Graphes d'événements temporisés (équation aux compteurs)	
	(Max ,+)	Graphes d'événements p-temporisés (équation aux dateurs)	Graphes d'événements p-temporels	Graphes d'événements t-temporels

Fig. 2.15 : Synthèse des modèles algébriques des graphes d'événements.

### Remarque 2.19

La figure 2.16 [Didi-Alaoui 2005] donne une autre présentation des différents liens existant entre ces modèles. Ainsi, les graphes d'événements temporisés sont un cas particulier de graphe d'évé-

nements temporels. ◇



**Fig. 2.16 :** Classification des graphes d'événements temporisés et temporels.

## 2.7 Conclusion

Les Graphes d'événements temporisés et temporels sont généralement modélisés dans l'algèbre  $(\max, +)$  à l'aide de "dateurs" et dans l'algèbre  $(\min, +)$  à l'aide de "compteurs". Dans ce chapitre, nous utilisons également les dateurs (et les compteurs pour les GET) mais dans l'algèbre standard, pour présenter une modélisation décrite par des inéquations dans l'algèbre standard afin de décrire ces modèles sous la forme  $Ax \leq b$ . La matrice  $A$  représente les matrices d'incidence du graphe d'événements considéré.

Dans ce chapitre, une classification algébrique des modèles a été donnée, afin de montrer le fonctionnement de chacun des modèles, les uns par rapport les autres. Cette classification se base sur le type de fonctions obtenues pour chacune de ces sous-classes.



# Chapitre 3

## Taux de production

### 3.1 Introduction

Le développement des systèmes fabriqués par l'homme exige que les exigences fonctionnelles et certaines performances soient respectées. Un réseau de Petri ordinaire n'introduit pas le concept de temps. Avec cette classe de réseaux, il est possible de décrire la structure logique et le comportement du système modélisé, mais pas son évolution au fil du temps. Il est nécessaire d'introduire le temps pour l'analyse de performances des systèmes à événements discrets. Le temps est introduit dans une variété de réseaux de Petri. Dans ce chapitre, on considère les graphes d'événements.

L'objectif de ce chapitre est l'évaluation de la performance des graphes d'événements. Ce choix a été motivé principalement par la reconnaissance de l'importance de l'utilisation des techniques analytiques pour l'évaluation de performance. Il est organisé comme suit : nous commençons par un rappel sur les éléments de la théorie des graphes, et les propriétés spectrales. Avant d'exposer notre contribution dans ce domaine, un état de l'art sur les travaux existants est proposé. Une approche pour calculer le taux de production utilise les propriétés spectrales de l'algèbre  $(\max, +)$ . Une autre approche, classique pour les GET, se base sur l'analyse des circuits (en prenant le maximum des rapports poids en temporisation sur poids en jetons [Gaubert 1992]). Ce théorème met en valeur les structures fondamentales du graphe d'événements, mais doit énumérer tous les circuits : elle n'aboutit pas à une résolution numériquement efficace pour des systèmes de taille moyenne.

Le point de départ de notre contribution est le modèle décrivant le comportement du graphe d'événements P-temporel dans l'algèbre standard (Chapitre 2, §2.5.3.2). On considère un comportement 1-périodique. Dans ce chapitre, deux parties pour analyser le temps de cycle  $\lambda$  seront présentées. La première partie est l'application du *lemme de Farkas* pour déterminer les bornes encadrant le temps de cycle [Declerck 2007]. Les expressions des bornes obtenues dépendent uniquement des caractéristiques du graphe d'événements P-temporel. Rappelons que l'inverse du temps de cycle est le taux de production. De ce fait, l'existence du temps de cycle est analysée. La deuxième partie consiste à proposer des techniques efficaces permettant de calculer les bornes

minimale et maximale du temps de cycle. A partir du modèle algébrique, nous montrons que les temps de cycle minimal et maximal peuvent simplement être obtenus moyennant l'utilisation d'algorithmes classiques de la programmation linéaire, comme le Simplexe. La transposition de cette étude en théorie des graphes est l'analyse des circuits, non pas du graphe d'événements, mais d'un graphe associé au graphe d'événements P-temporel, ce dernier est défini dans [Khansa 1997]. Notons que la simulation ne permet pas de répondre de manière satisfaisante à cette question car les graphes d'événements P-temporels nécessitent une anticipation de l'évolution du système sous peine de morts de jetons et de blocages. A notre connaissance, les seuls travaux portant sur le calcul du temps de cycle des graphes d'événements P-temporels sont [Khansa 1997].

### 3.2 Propriétés spectrales des matrices de $\mathbb{R}_{max}^{n \times n}$

Cette partie rappelle les principales propriétés spectrales des matrices à coefficients dans le dioïde  $\mathbb{R}_{max}^{n \times n}$ , pour les matrices irréductibles (voir Annexe B.1). On rappelle comment ces résultats peuvent être appliqués à l'évaluation de performances des GET.

**Définition 3.1.** Soit  $A$  une matrice de dimension  $n \times n$  à coefficients dans un dioïde  $\mathcal{D}$ . Si un couple  $(\lambda, \vartheta) \in \mathbb{R}_{max} \times (\mathbb{R}_{max}^n - \{\varepsilon\}^n)$ , vérifie :

$$A \otimes \vartheta = \lambda \otimes \vartheta \quad (3.1)$$

on dit que  $\lambda$  (respectivement,  $\vartheta$ ) est valeur propre (respectivement, vecteur propre) de  $A$ .

On note  $\rho_{max}(A)$  le rayon spectral de  $A$ , i.e, la plus grande des valeurs propres de  $A$ .

La matrice irréductible ne possède qu'une seule valeur propre donnée par le théorème de Perron-Frobenius, soit :

$$\rho_{max}(A) = \max_{1 \leq k \leq n} \max_{i_1, \dots, i_n} \frac{A_{i_1 i_2} + A_{i_2 i_3} + \dots + A_{i_k i_1}}{k}. \quad (3.2)$$

#### A. Le calcul de la valeur propre en utilisant la théorie des graphes

La matrice  $A$  de dimension  $n \times n$  dans un dioïde  $\mathcal{D}$  est une matrice de précédence, l'élément  $A_{ij}$  désigne le poids  $w_{ij}$  de l'arc  $j \rightarrow i$ . Le poids d'un arc absent du graphe est égal à  $\varepsilon$ . A toute matrice carrée  $A$ , on peut associer un graphe de précédence, noté  $\mathcal{G}(A)$ .

La matrice est dite irréductible si pour toute paire  $(i; j)$ , il existe un entier  $k$  tel que  $(A^k)_{ij} \neq \varepsilon$ . En terme des graphes, le graphe  $\mathcal{G}(A)$  est fortement connexe. Le théorème suivant permet de calculer la valeur propre d'une matrice irréductible.

**Théorème 3.2.** [Gondran 1979] Une matrice  $A \in \mathbb{R}_{max}^{n \times n}$  irréductible admet une unique valeur propre, notée  $\lambda$ , égale au rayon spectral de la matrice  $A$ , soit :

$$\lambda = \rho_{max}(A) = \bigoplus_{k=1}^n (\text{trace} A^k)^{1/k}. \quad (3.3)$$

Où  $\text{trace} A^k = \bigoplus_{i=1}^n (A^k)_{ii}$   $A^k$  est une matrice carrée de dimension  $n \times n$  à éléments dans  $\mathbb{R}$ , l'éléments  $(A^k)_{ij}$  représente le poids maximal des chemins de longueur  $k$  allant du nœud  $j$  au nœud  $i$ .

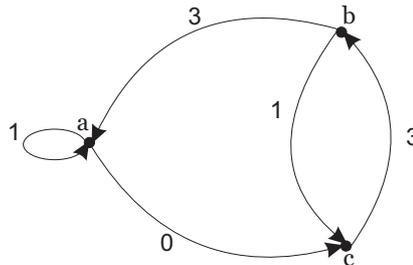
Pour  $k = 2$ , on a ainsi :

$$(A^2)_{ij} = \max_{l=1..n} \{A_{il} + A_{lj}\}.$$

D'après ce théorème, la valeur propre  $\lambda = \rho_{max}(A)$  correspond, dans le graphe  $G(A)$ , au poids moyen du circuit de poids moyen maximal (le poids d'un circuit étant la somme des poids des arcs qui le composent). Par analogie avec la notion de chemin critique, un tel circuit est appelé *circuit critique* du système. Ce sont les circuits qui limitent les performances du système et la valeur  $\frac{1}{\lambda}$  est le taux de production maximal qui peut être obtenu. Par ailleurs, les vecteurs propres associés à la valeur propre  $\lambda$  fournissent les différentes conditions d'initialisation possibles permettant d'obtenir du système ces performances maximales.

### Exemple 3.3

Soit le graphe de précedence suivant :



**Fig. 3.1 :** Graphe de précedence fortement connexe.

Sa matrice irréductible dans le dioïde  $\mathbb{R}_{max}^{3 \times 3}$  :

$$A = \begin{pmatrix} 1 & 3 & \varepsilon \\ \varepsilon & \varepsilon & 3 \\ 0 & 1 & \varepsilon \end{pmatrix},$$

la valeur propre associée à la matrice  $A$  peut être calculée en utilisant le théorème 3.2. On a :

$$A^2 = \begin{pmatrix} \varepsilon & 4 & 6 \\ 3 & 4 & \varepsilon \\ 1 & 3 & 4 \end{pmatrix}, \quad A^3 = \begin{pmatrix} 6 & 7 & 7 \\ 4 & 6 & \varepsilon \\ \varepsilon & 5 & 6 \end{pmatrix}. \text{ D'où :}$$

$$\begin{aligned}
- \text{trace}(A) &= \bigoplus_{k=1}^3 A_{ii} = 1, \\
- \text{trace}(A^2) &= \bigoplus_{k=1}^3 A_{ii}^2 = 4, \\
- \text{trace}(A^3) &= \bigoplus_{k=1}^3 A_{ii}^3 = 6,
\end{aligned}$$

alors, on a :

$$\lambda = \rho_{max}(A) = \bigoplus_{k=1}^n (\text{trace} A^k)^{1/k} = tr(A) \oplus tr(A^2)^{1/2} \oplus tr(A^3)^{1/3} = 2.$$

◇

## B. Le calcul de la valeur propre en utilisant l'algorithme de Karp

Le rayon spectral de la matrice  $A$  peut se calculer directement comme ci-dessus. L'algorithme consiste à évaluer la formule (3.2), ce qui suppose le calcul des puissances successives de la matrice  $A$  jusqu'à l'ordre  $n$  (= taille de  $A$ ). Pour une matrice pleine, on a ainsi un algorithme de complexité en  $O(n^4)$ . Une méthode plus efficace (en temps  $O(n^3)$ ) permettant de calculer la valeur propre d'une matrice consiste à appliquer une formule due à Karp [Karp 1978]. Pour d'autres détails concernant l'application de Karp pour le calcul de la valeur propre, on pourra se référer au document [Gaubert 1995].

## 3.3 Comportement périodique d'un graphe d'événements temporisé

L'établissement de la valeur propre de  $A$  permet d'évaluer le comportement périodique du GET correspondant en régime autonome. Le théorème suivant [Baccelli 1992] précise le comportement du graphe pour une condition initiale quelconque.

**Théorème 3.4.** *Pour une matrice irréductible  $A \in \mathbb{R}^{n \times n}$  de valeur propre  $\lambda$ , il existe un entier  $c$  et une propriété de cyclicité de la forme suivante :*

$$n \geq N, A^{n+c} = \lambda^c \otimes A^n, \quad (3.4)$$

où  $c$  est appelé la cyclicité de  $A$  et  $n$  est supérieur à un entier  $N$ .

Cette cyclicité définit un certain fonctionnement périodique en régime permanent, après un régime transitoire.

Un graphe d'événements temporisé fonctionnant en régime autonome, dès l'état initial. Un tel graphe se représente par l'équation récurrente.

$$x(k+c) = A \otimes x(k-1).$$

Lorsque  $A$  est irréductible, le théorème de cyclicité entraîne que le comportement est périodique, *i.e.*,

$$x(k+c) = \lambda^c \otimes x(k),$$

pour  $k$  assez grand. En revenant à l'algèbre usuelle, cela s'écrit :

$$x(k+c) = c \times \lambda + x(k),$$

où  $\lambda$  est la valeur propre de  $A$  et le vecteur d'état initial  $x(0)$  est un vecteur propre de  $A$ .

Après un régime transitoire fini, le régime devient *c-périodique* de période égale à  $c$ . Autrement dit,  $c$  événements se produisent tous les  $c\lambda$  unités de temps, et  $\lambda$  est le temps moyen de production d'une pièce (l'inverse du taux de production). Ainsi, un graphe d'événements temporisé fortement connexe fonctionnant en régime autonome atteint après un régime transitoire fini un régime périodique, le taux de production étant l'inverse de la valeur propre de la matrice  $A$ .

### 3.4 Evaluation de performances des graphes d'événements dans l'algèbre standard

Dans cette section, on introduit les techniques d'analyse pour analyser la performance (calcul du temps de cycle) des GET et les graphes P-temporels en particulier dans l'algèbre standard.

**Définition 3.5.** *Le temps de cycle noté  $\lambda$ , d'un graphe d'événements, est le temps moyen séparant deux tirs consécutifs d'une transition. Pour un système de production, il correspond au temps moyen séparant deux pièces produites.*

#### A. Evaluation de performances des GET dans l'algèbre standard

Dans la section précédente, on a associé à un graphe d'événements donné une matrice  $A \in \mathbb{R}_{max}^{n \times n}$  et montré que le temps de cycle était égal au rayon spectral de  $A$ . On peut aussi caractériser directement le temps de cycle à partir du graphe sans avoir à construire la matrice  $A$ . Le résultat donne le théorème suivant.

**Théorème 3.6.** [*Ramamoorthy 1980*] *Le temps de cycle  $\lambda$  d'un graphe d'événements temporisé fortement connexe est le maximum des temps de cycles de tous les circuits élémentaires du graphe.*

$$\lambda = \max_{c \text{ circuit élémentaire}} \frac{T(c)}{M(c)}, \quad (3.5)$$

où  $M(c)$  correspond au nombre total de jetons du circuit  $c$  (équivalent au poids du circuit du graphe de précedence associé) et  $T(c)$  correspond à la somme des temporisations des places du circuit  $c$  (équivalent à la longueur du circuit du graphe de précedence associé).

Dans les travaux proposés par P. Chrétienne [Chrétienne 1983] et Ramamoorthy [Ramamoorthy 1980], le nombre des circuits augmente d'une manière exponentielle avec la taille du graphe d'événements, mesurée par le nombre de ces places et transitions. Donc l'énumération de tous les circuits est non polynomial. C'est la raison pour laquelle d'autres contributions sont nécessaires. Parmi ces contribution, certaines convertissent le calcul du temps de cycle en un problème de programmation linéaire [J. Campos 1992], [Magott 1984] et [Giua 2000].

Selon [J. Campos 1992], le programme linéaire pour calculer le temps de cycle est comme suit :

$$\lambda = \text{Max } y^t f,$$

$$\text{tel que } : y^t W \leq 0, y^t m_0 \leq 1, y \geq 0,$$

où  $y$  est un vecteur de dimension  $n$  ( $n$  est le nombre de places),  $f$  est une fonction (depend de la matrice d'incidence arrière  $W^+$  et le vecteur des délais moyens du franchissemnt des transitions  $x_i$ ),  $W$  est la matrice d'incidence, et  $m_0$  est le marquage initial. Ce problème a  $n$  variables (correspond au nombre de places) et  $n + s + 1$  contraintes, où  $s$  est le nombre de transitions. Le problème de programmation linéaire peut être réduit à un autre problème linéaire de  $n - s + 1$  variables et  $n$  contraintes, cette transformation est prouvée dans [Yamada 1994], ou par un problème avec  $s + 1$  variables et  $n + 1$  contraintes [Magott 1984].

## B. Evaluation de performances des GE P-temporels dans l'algèbre standard

Les GET constituent une sous classe des graphes d'événements temporels. Ils ont été introduits plus tôt que les graphes d'événements temporels. Aussi la littérature scientifique qui les concerne est plus abondante. Pour cette raison, un grand nombre de propriétés des graphes d'événements temporels ont été présentées comme une généralisation de propriétés préalablement établies pour les GET.

Les définitions fondamentales établies pour les GET étant présentées, nous allons énoncer leur généralisation pour les graphes d'événements P-temporels. Le théorème suivant permet de calculer le temps de cycle dans un graphe p-temporels, ce qui représente une extension des travaux de Ramamoorthy [Ramamoorthy 1980] réalisés sur les GET.

**Théorème 3.7.** [Zhou 1999] Soit un graphe d'événements P-temporel fortement connexe, ses performances ( $[\lambda'_{min}, \lambda'_{max}]$ ) peuvent être évaluées en résolvant les deux problèmes linéaires suivants :

$$\lambda'_{min} = \text{Max } \left( \frac{T_{i,min}}{M_i} \right),$$

$$\lambda'_{max} = \text{Max } \left( \frac{T_{i,max}}{M_i} \right),$$

où :  $T_{i,min}$  (respectivement,  $T_{i,max}$ ) est la durée totale du  $i^{me}$  circuit basé sur  $T_{min}$  (respectivement,  $T_{max}$ ) pour toute les temporisations. Notons que chaque place  $p_i$  est associée à une temporisation  $[T_{i,min}, T_{i,max}]$ .

Nous verrons par la suite que ces bornes ne constituent qu'un encadrement de  $\lambda'_{min}$  et  $\lambda'_{max}$ .  
où :  $[\lambda_{min}, \lambda_{max}] \subset [\lambda'_{min}, \lambda'_{max}]$ ,  
avec  $\lambda_{min}$  et  $\lambda_{max}$  les bornes du temps de cycle calculées par notre approche.

### 3.5 Approche proposée pour l'évaluation de performances des GE P-temporels

Notre approche proposée pour l'évaluation de performances est basée sur deux points. Le premier objectif est la détermination de l'existence du temps de cycle  $\lambda$  en utilisant le modèle algébrique proposé dans (Chapitre 2, §2.5.3.2), le deuxième point est le calcul des deux bornes (minimale et maximale) du temps de cycle en utilisant le théorème de la programmation linéaire 1.34. Nous supposons dans cette partie que le graphe d'événements P-temporel a un comportement 1-périodique et son marquage initial satisfait  $0 \leq m_{ij} \leq 1$ .

Le comportement 1-périodique est défini par  $x(k+1) = \lambda \times u + x(k)$  avec  $u = (1, 1, \dots, 1)^t$  et  $\lambda$  le temps de cycle. A partir de l'écriture algébrique de notre système, nous pouvons écrire :

$$\begin{pmatrix} G^- \\ G^+ \end{pmatrix} \times \begin{pmatrix} x(k) \\ x(k+1) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}, \quad (3.6)$$

avec  $\begin{pmatrix} G^- \\ G^+ \end{pmatrix} = \begin{pmatrix} G_1^- & G_0^- \\ G_1^+ & G_0^+ \end{pmatrix}$ . En remplaçant  $x(k+1)$  par  $\lambda \times u + x(k)$ , on obtient :

$$\begin{pmatrix} G_1^- & G_0^- \\ G_1^+ & G_0^+ \end{pmatrix} \times \begin{pmatrix} x(k) \\ \lambda \times u + x(k) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix},$$

soit

$$\begin{pmatrix} G_1^- + G_0^- \\ G_1^+ + G_0^+ \end{pmatrix} \times x(k) + \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times \lambda \times u \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}.$$

Comme

$$W = \sum_{i=0}^{i=m} G_i^- \quad \text{et} \quad -W = \sum_{i=0}^{i=m} G_i^+, \quad (3.7)$$

on obtient

$$\begin{pmatrix} W \\ -W \end{pmatrix} \times x(k) + \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times \lambda \times u \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}. \quad (3.8)$$

### 3.5.1 Condition d'existence du temps de cycle dans un GE P-temporel

Après l'obtention d'un système d'inégalités linéaires du type  $A \times x \leq b$ , nous analysons l'existence du temps de cycle  $\lambda$  en utilisant le lemme de Farkas 1.28. On obtient le théorème suivant.

**Théorème 3.8.** *Le système (3.8) peut suivre un comportement 1-périodique pour un temps de cycle  $\lambda$  donné, si et seulement si, pour chaque vecteur ligne du vecteur  $y \geq 0$  avec*

$$y \times \begin{pmatrix} W \\ -W \end{pmatrix} = 0, \quad (3.9)$$

on a :

$$\frac{y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}}{y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u} \geq \lambda, \quad \text{si} \quad y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u > 0, \quad (3.10)$$

$$\frac{y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}}{y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u} \leq \lambda, \quad \text{si} \quad y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u < 0, \quad (3.11)$$

$$y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} \geq 0, \quad \text{si} \quad y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u = 0. \quad (3.12)$$

**Preuve :**

On a :

$$\begin{pmatrix} G_1^- & G_0^- \\ G_1^+ & G_0^+ \end{pmatrix} \times \begin{pmatrix} x(k) \\ \lambda \times u + x(k) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix},$$

soit

$$\begin{cases} G_1^- \times x(k) + G_0^- \times (\lambda \times u + x(k)) \leq -T^- \\ G_1^+ \times x(k) + G_0^+ \times (\lambda \times u + x(k)) \leq T^+ \end{cases},$$

soit

$$\begin{cases} (G_1^- + G_0^-) \times x(k) \leq -T^- - G_0^- \times (\lambda \times u) \\ (G_1^+ + G_0^+) \times x(k) \leq T^+ - G_0^+ \times (\lambda \times u) \end{cases},$$

soit

$$\begin{pmatrix} G_1^- + G_0^- \\ G_1^+ + G_0^+ \end{pmatrix} \times x(k) \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} - \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times \lambda \times u. \quad (3.13)$$

En utilisant (3.7), on obtient :

$$\begin{pmatrix} W \\ -W \end{pmatrix} \times x(k) \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} - \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times \lambda \times u. \quad (3.14)$$

A partir du lemme de Farkas, on peut déduire que le système (3.14) d'inégalités linéaires a une solution  $x$ , si et seulement si,  $y \times \left( \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} - \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times \lambda \times u \right) \geq 0$  pour chaque vecteur ligne du vecteur  $y \geq 0$  avec  $y \times \begin{pmatrix} W \\ -W \end{pmatrix} = 0$ .

$$\text{Donc, } y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} - y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times (\lambda \times u) \geq 0,$$

soit

$$y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} \geq y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times (\lambda \times u) = \lambda \times y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u.$$

Dans cette relation, le produit par  $u$  donne une addition avec toutes les colonnes de  $\begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix}$ .

A partir du signe de  $y \times \begin{pmatrix} G_0^- \\ G_0^+ \end{pmatrix} \times u$ , les deux cas (3.11) et (3.10) et les conditions nécessaires et suffisantes de l'existence de  $x$  peuvent être déduites.

□

### Remarque 3.9

Le vecteur  $y$  n'est pas le vecteur p-invariant décrit dans la définition 2.4, puisque les deux vecteurs n'ont pas la même taille.  $|y| > |\theta|$  ◇

### Exemple 3.10

On considère le graphe d'événements P-temporel suivant :

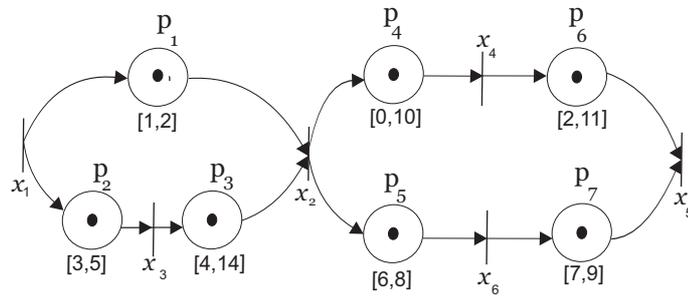


Fig. 3.2 : Graphe d'événements P-temporel.

Le modèle algébrique est donné comme suit :

$$\begin{pmatrix} W^+ & -W^- \\ -W^+ & W^- \end{pmatrix} \cdot \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix},$$

avec  $x(k) = (x_1(k) \ x_2(k) \ x_3(k) \ x_4(k) \ x_5(k) \ x_6(k))^t$ ,

$$W^+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, W^- = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, -T^- = \begin{pmatrix} -1 \\ -3 \\ -4 \\ 0 \\ -6 \\ -2 \\ -7 \end{pmatrix}, T^+ = \begin{pmatrix} 2 \\ 5 \\ 14 \\ 10 \\ 8 \\ 11 \\ 9 \end{pmatrix}.$$

On a :  $W = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}.$

**Existence du temps de cycle :**

Une matrice entière  $Y \geq 0$  avec  $Y \cdot \begin{pmatrix} W \\ -W \end{pmatrix} = 0$  est comme suit :

$$Y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$

$$\begin{pmatrix} -W^- \\ W^- \end{pmatrix} \times u = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^t,$$

$$Y \cdot \begin{pmatrix} -W^- \\ W^- \end{pmatrix} \times u = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \end{pmatrix}^t,$$

$$Y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix} = \begin{pmatrix} +1 & +2 & +10 & +10 & +2 & +9 & +2 & +18 & -5 & +15 & +10 \end{pmatrix}^t.$$

Pour  $\frac{Y \times \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}}{Y \cdot \begin{pmatrix} -W^- \\ W^- \end{pmatrix} \times u}$ , on obtient la borne supérieure  $\lambda_{max} = \frac{+18}{+1} = 18$  et la borne inférieure  $\lambda_{min} = \frac{-5}{-1} = 5$ .

◇

### Remarque 3.11

Dans un graphe d'événements P-temporel fortement connexe, si nous appliquons le théorème 3.7, nous obtenons les bornes  $\lambda_{max}$  et  $\lambda_{min}$  qui encadrent les valeurs obtenues par notre approche. En effet, l'approche ne considère que les temporisations inférieures pour la borne inférieure et les temporisations supérieures pour la borne supérieure, le théorème que nous proposons utilise les temporisations supérieures et inférieures pour calculer chacune des bornes.

◇

## 3.5.2 Le calcul du temps de cycle d'un graphe d'événements P-temporel

Après avoir prouvé l'existence du temps de cycle dans un graphe d'événements P-temporel (chaque place de ce graphe a au plus un jeton) avec un comportement 1-périodique, nous proposons des techniques permettant de calculer les bornes minimale et maximale du temps de cycle. Contrairement à de nombreuses publications considérant les GET, le point de départ de l'approche n'est pas un théorème basé sur l'analyse des circuits (qui compare des rapports de poids en jetons et en temporisation), mais directement en considérant le modèle décrivant le comportement du système.

A partir du modèle décrivant le comportement du système, nous montrons que les temps de cycle minimal  $\lambda_{min}$  et maximal  $\lambda_{max}$  peuvent être obtenus moyennant l'utilisation de la théorie de la programmation linéaire. Dans cette partie, nous proposons deux approches qui permettent de déterminer le temps de cycle  $\lambda$  et les dates de tir des transitions.

### 3.5.2.1 La forme primale

A partir de l'écriture (3.14), pour une raison de simplification, on remplace  $x(k) = x$  et  $u = (1, 1, \dots, 1)^t$ . On obtient l'écriture suivante :

$$\begin{pmatrix} W & G_0^- \cdot u \\ -W & G_0^+ \cdot u \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}. \quad (3.15)$$

Cette écriture peut être simplifiée car nous remarquons que dans le cas de graphe d'événements avec un marquage initial unitaire, on a :  $G_0^- = -W^-$  et  $G_1^- = W^+$ . Ainsi,  $G_0^- \cdot u = -W^- \cdot u = -M_0$  (ce qui correspond à la somme des colonnes de  $-W^-$ ), ce qui représente le

marquage initial. Ce résultat est également vrai lorsque chaque place a initialement un jeton au plus. En effet, chaque place sans jeton est représentée par une ligne de  $G_0^-$  contenant les poids de l'arc entrant 1 et de l'arc sortant -1. De même, on a :  $G_0^+ . u = W^- . u = M_0$ .

Alors on peut écrire (3.15) comme suit :

$$\begin{pmatrix} W & -M_0 \\ -W & M_0 \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \end{pmatrix} \leq \begin{pmatrix} -T^- \\ T^+ \end{pmatrix}.$$

On notera que chaque ligne de la contrainte contient 3 variables par inégalité :  $\lambda$  et 2 variables  $x_i$ . En appliquant un algorithme de la programmation linéaire, on pourra ainsi déterminer la valeur optimale de  $\lambda$ . Pour la bonne application de l'algorithme, il est cependant nécessaire que la solution optimale, qui est un point extrême (ou une surface extrême) du polyèdre, soit finie. De même, si on utilise un algorithme comme le Simplexe, alors tous les points extrêmes qui conduisent à la solution optimale devront être également finis. Le plus petit et le plus grand  $\lambda$  sont fixés par les caractéristiques du graphe d'événements P-temporel. Ce n'est cependant pas le cas pour l'instant pour le plus petit ou le plus grand vecteur  $x$ . La solution est de borner par dessous (respectivement, par dessus) par un minorant (respectivement, un majorant) fini le problème de minimisation (respectivement, maximisation). Le minorant de l'état  $x$  (respectivement, majorant) est noté  $L$  (respectivement, noté  $U$ ). Sous condition de compatibilité du système de contraintes et qu'il existe un plus petit (respectivement, grand) temps de cycle, les propositions suivantes présentent une solution dans  $\mathbb{R}$ .

*Min*  $\lambda$

sous la contrainte :

$$\begin{pmatrix} -I & 0 \\ W & -M_0 \\ -W & M_0 \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \end{pmatrix} \leq \begin{pmatrix} -L \\ -T^- \\ T^+ \end{pmatrix}. \quad (3.16)$$

*Max*  $\lambda$

sous la contrainte :

$$\begin{pmatrix} I & 0 \\ W & -M_0 \\ -W & M_0 \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \end{pmatrix} \leq \begin{pmatrix} U \\ -T^- \\ T^+ \end{pmatrix}. \quad (3.17)$$

- Remarquons que les bornes  $L$  et  $U$  peuvent décrire des dates de tir désirées au plus tôt ou au plus tard. Les dates de tir calculées donneront des dates suivant au plus près ces limites.

- La non-décroissance des trajectoires peut être imposée dans le modèle algébrique avec l'écriture  $x(k+1) \geq x(k)$  avec pour conséquence  $\lambda \geq 0$ . En effet, cela donne  $x(k-1) - x(k) \leq 0$  et donc,  $m=1$ ,  $G_1^- = W^+ = I$ ,  $G_0^- = -W^- = -I$  et  $-T^- = 0$ .  
On obtient  $W = 0$  et  $-M_0 = -u$ . Ainsi,

$$\begin{pmatrix} W & -M_0 \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \end{pmatrix} \leq \begin{pmatrix} -T^- \end{pmatrix}$$

se réduit à  $-u \cdot \lambda \leq 0$ , soit  $\lambda \geq 0$ .

### 3.5.2.2 Technique de résolution

Le programme linéaire (3.16) de minimisation de  $\lambda$  peut être transformé pour ne comporter que des équations, en ajoutant une variable  $s_i$ , appelée variable d'écart, pour chaque contrainte, le programme linéaire s'écrit alors comme suit :

*Min*  $\lambda$   
sous la contrainte :

$$\begin{pmatrix} -I & 0 & I \\ W & -M_0 & I \\ -W & M_0 & I \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \\ s \end{pmatrix} = \begin{pmatrix} -L \\ -T^- \\ T^+ \end{pmatrix}. \quad (3.18)$$

En appliquant la même transformation pour le problème de maximisation, on obtient :

*Max*  $\lambda$   
sous la contrainte :

$$\begin{pmatrix} I & 0 & I \\ W & -M_0 & I \\ -W & M_0 & I \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \\ s \end{pmatrix} = \begin{pmatrix} U \\ -T^- \\ T^+ \end{pmatrix}. \quad (3.19)$$

$I$  est la matrice identité  $m \times m$ , et  $s$  est le vecteur des variables d'écarts. Notons que les deux programmes linéaires sont équivalents. On considérera dorénavant que les programmes linéaires sont écrits sous cette forme. La matrice  $A$  définissant les contraintes du programme linéaire est désormais de dimension  $m \times (m+n)$ , la dimension du vecteur coût  $c$  est égale à  $m+n$  et les  $m$  derniers éléments de  $c$  sont nuls.

L'algorithme choisi pour le calcul du taux de production, en utilisant la forme primale, est l'algorithme du Simplexe (voir Annexe B.2). Pour cette étude, on utilise **LPSolve IDE** (Integrated Development Interface), qui est un environnement connu de résolution des programmes

linéaires sous Windows.

A partir des formulations (3.19) et (3.18), le **LPSolve IDE** construit tout d'abord une solution réalisable, qui est un sommet du polytope, puis se déplace selon les arêtes du polytope pour atteindre des sommets pour lesquels la valeur de l'objectif est de plus en plus grande, jusqu'à atteindre l'optimum. Bien que cet algorithme soit efficace en pratique et qu'il soit assuré de trouver l'optimum, son comportement dans le pire cas peut être mauvais. Il est ainsi possible de construire un PL pour lequel la méthode du Simplexe requiert un nombre d'étapes exponentiel en la taille du problème.

### 3.5.2.3 La forme duale

La forme primale ne donne pas d'information sur la partie du système qui est à l'origine de ce plus grand temps de cycle. L'approche suivante apporte une réponse à cette question. En appliquant le principe de la dualité, nous allons nous affranchir de ces variables, tout en introduisant une nouvelle variable que nous notons  $y^t$  (les raisons de cette notation seront données plus tard). Ce type d'approche a déjà été appliqué dans [J. Campos 1992] et [Giua 2000], mais pour les GET. De plus, nous choisissons un raisonnement inverse, ne partant pas du théorème bien connu du calcul de temps de cycle à partir des circuits élémentaires. Il en résulte une démarche plus simple.

Rappelons d'abord les problèmes duaux

$$\left\{ \begin{array}{l} \text{Min}_{y \in \mathbb{R}_+^n} y.b \text{ sous } y.A = c \text{ et } y \geq 0, \\ \text{Max}_{x \in \mathbb{R}^m} c.x \text{ sous } A.x \leq b \text{ avec } x \text{ quelconque,} \end{array} \right. \begin{array}{l} \text{Problème (P);} \\ \text{Problème (D).} \end{array}$$

Notre problème correspond à (D) en raison de l'inégalité (3.16). Considérons, pour simplifier la présentation, la minimisation de  $\lambda$ . Ainsi, on a :

$$A = \begin{pmatrix} -I & 0 \\ W & -M_0 \\ -W & M_0 \end{pmatrix}, b = \begin{pmatrix} -L \\ -T^- \\ T^+ \end{pmatrix} \text{ et } c = \begin{pmatrix} 0 \dots 0 & -1 \end{pmatrix} \text{ comme}$$

$$\text{Min } \lambda = \text{Max}(-\lambda) = \text{Max} \begin{pmatrix} 0 \dots 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix}.$$

Nous obtenons ainsi le problème dual suivant (P) qui est également un problème de programmation linéaire.

Min  $\begin{pmatrix} -L^t & -(T^-)^t & (T^+)^t \end{pmatrix} \cdot y^t$   
 sous la contrainte

$$y \cdot \begin{pmatrix} -I & 0 \\ W & -M_0 \\ -W & M_0 \end{pmatrix} = \begin{pmatrix} 0 \dots 0 & -1 \end{pmatrix}, \quad \text{avec } y \geq 0.$$

On introduit la notation  $X^{ad} = \{x \in \mathbb{R}^m | A \cdot x \leq b\}$  qui désigne l'ensemble des  $x$  admissibles pour le problème dual. De même, soit  $Y^{ad} = \{y \in \mathbb{R}_+^n | y \cdot A = c\}$ .

Ces formes duales sont pleinement liées car nous pouvons appliquer le théorème de la Dualité 1.32, les matrices  $A$ ,  $b$  et  $c$  étant dans  $\mathbb{R}$ . De plus, si dans l'approche 1, les solutions sont dans  $\mathbb{R}$ , alors l'approche 2 donnera également des solutions correspondantes finies et inversement.

Le théorème suivant fait le lien avec l'approche en utilisant la forme primale. L'approche duale ne donne pas la date où chaque transition doit initier son premier tir, ni *a priori*, le temps de cycle  $\lambda$  mais un certain vecteur  $y$  pour l'instant, sans sens physique. L'application du théorème de la dualité va montrer que la solution optimale du problème (P) ci-dessus donne également le temps de cycle minimal. Il en est de même pour le temps de cycle maximal. On note  $b_1^t = \begin{pmatrix} -L^t & -(T^-)^t & (T^+)^t \end{pmatrix}$  et  $b_2^t = \begin{pmatrix} U^t & -(T^-)^t & (T^+)^t \end{pmatrix}$  les deux vecteurs présents dans les problèmes (3.16) et (3.17). Soient  $\begin{pmatrix} x \\ \lambda \end{pmatrix}$  et  $y^t$  appartenant respectivement aux ensembles admissibles des problèmes (D) et (P), définies ci-dessus pour le problème (3.16) et,  $\begin{pmatrix} x_{min} \\ \lambda_{min} \end{pmatrix}$  et  $y_{min}^t$  les solutions optimales correspondantes. Symétriquement, pour le problème (3.17), soient  $\begin{pmatrix} x_{max} \\ \lambda_{max} \end{pmatrix}$  et  $y_{max}^t$  les solutions optimales correspondantes. Alors, nous avons le résultat suivant.

**Théorème 3.12.** *Pour les problèmes (3.16) et (3.17), nous avons respectivement*

$$-y \cdot b_1 \leq \lambda \quad \text{et} \quad \lambda \leq y \cdot b_2,$$

ainsi que

$$\lambda_{min} = -y_{min} \cdot b_1 \quad \text{et} \quad \lambda_{max} = y_{max} \cdot b_2.$$

**Preuve :**

Le point 1 du théorème de la dualité indique que :  $y \cdot b \geq c \cdot x$  d'où, pour le problème (3.16),  $\begin{pmatrix} -L^t & -(T^-)^t & (T^+)^t \end{pmatrix} \cdot y^t \geq \begin{pmatrix} 0 \dots 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = -\lambda$ . De plus, les valeurs optimales

associées sont égales (point 2) si les deux problèmes admettent une solution optimale finie. Le raisonnement est identique pour le problème (3.17).

□

### 3.5.2.4 La forme duale simplifiée

Si nous posons  $L = 0$  (respectivement,  $U = 0$ ), nous pouvons simplifier l'approche en utilisant la forme duale. Cette hypothèse effectue une modification des premières dates de tir des transitions, mais n'a pas d'effet sur le calcul des bornes du temps de cycle qui sont des constantes du système, indépendante de ces dates : les formes précédentes (la forme primal et duale) montrent que le calcul des bornes ne dépend que des paramètres du graphe d'événements P-temporel.

On note  $y^t = \begin{pmatrix} y_1^t \\ y_2^t \end{pmatrix}$ , où  $y_1^t$  a une dimension égale à celle de  $x$ . Le problème ( $P$ ) peut être réécrit sous la forme suivante :

$$\text{Min} \left( -(T^-)^t \quad (T^+)^t \right) \cdot y_2^t$$

sous contraintes :

$$\left\{ \begin{array}{l} y_2 \cdot \begin{pmatrix} W \\ -W \end{pmatrix} \geq 0, \\ y_2 \cdot \begin{pmatrix} M_0 \\ -M_0 \end{pmatrix} = 1. \\ y_2 \geq 0. \end{array} \right. \quad (3.20)$$

**Preuve :**

En effet, le problème ( $P$ ) devient :

$$y_1 \cdot \begin{pmatrix} -I & 0 \end{pmatrix} + y_2 \cdot \begin{pmatrix} W & -M_0 \\ -W & M_0 \end{pmatrix} = \begin{pmatrix} 0 \dots 0 & -1 \end{pmatrix}.$$

On en déduit directement que  $y_2 \cdot \begin{pmatrix} M_0 \\ -M_0 \end{pmatrix} = 1$ .

Le deuxième résultat découle de  $y_1 \geq 0$ .

□

Ainsi, chaque composante non-nulle du vecteur  $y_2$  va sélectionner une ligne des matrices  $\begin{pmatrix} W \\ -W \end{pmatrix}$  et  $\begin{pmatrix} M_0 \\ -M_0 \end{pmatrix}$ , ce qui correspond à une place et son marquage initial. Le vecteur

$y_2$  répondant au problème de minimisation (respectivement, maximisation) définira une structure critique du graphe d'événements P-temporel limitant la vitesse la plus rapide par dessous (respectivement, limitant la vitesse la plus lente par dessus) du système.

Le théorème suivant va permettre d'analyser la première relation de (3.20).

**Théorème 3.13.** (Stiemke 1915, [Schrijver 1987]). *Pour une matrice  $A$ , les cas suivants s'excluent mutuellement.*

- cas 1.  $Ax = 0$ ,  $x > 0$  a une solution  $x$ .
- cas 2.  $y.A \geq 0$  et  $y.A \neq 0$  a une solution  $y$ .

Soit  $A = \begin{pmatrix} W \\ -W \end{pmatrix}$ . Comme dans un graphe d'événements, chaque ligne de  $W$  a exactement un coefficient 1 et -1, il existe  $x > 0$  tel que  $A.x = 0$ , ce qui correspond au cas 1. On peut en déduire que le graphe d'événements est "répétitif" (*consistent* en anglais), c'est-à-dire, qu'il existe une séquence de tir et un marquage initial permettant de retrouver ce même marquage sous la condition que chaque transition soit tirée au moins une fois (voir [Murata 1989], page 567).

L'application du théorème de Stiemke entraîne que le cas 2 est sans solution : le système  $y.A \geq 0$  et  $y.A \neq 0$  n'a pas de solution  $y$ . D'autre part, la condition  $y.A \geq 0$  dans (3.20) doit être vérifiée afin de pouvoir calculer le temps de cycle. La seule possibilité pour que le système du cas 2 soit sans solution est  $y.A = 0$ , ou sous une forme développée :

$$y_2 \cdot \begin{pmatrix} W \\ -W \end{pmatrix} = 0.$$

#### Exemple 3.14

Suite du graphe d'événements P-temporel dans (3.2). Le calcul du temps de cycle en utilisant les deux approches est comme suit :

##### A. La forme primale

Pour une fonction objective  $f : \lambda_{min}$  le résultat est comme suit :

$$\begin{pmatrix} x(k) \\ \lambda \end{pmatrix} = \begin{pmatrix} 3 & 0 & 1 & 0 & 3 & 1 & 5 & 5 \end{pmatrix}^t \text{ avec } f = 5.$$

La fonction objective  $f = \lambda_{max}$

$$\begin{pmatrix} x(k) \\ \lambda \end{pmatrix} = \begin{pmatrix} 36 & 19 & 23 & 7 & 0 & 9 & 18 & 18 \end{pmatrix}^t \text{ avec } f = -18.$$

##### B. La forme duale

Les valeurs correspondant à la temporisation du graphe d'événements sont les 14 dernières valeurs.

$$c^t = ( 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000 \ -1 \ -3 \ -4 \ 0 \ -6 \\ -2 \ -7 \ 2 \ 5 \ 14 \ 10 \ 8 \ 11 \ 9 ),$$

$$\begin{aligned}
y &= ( 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \\
&0 \ 0 ), \\
&\text{et } g = -5, \\
y &= ( 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \\
&0 \ 0 ), \text{ et } g = 18.
\end{aligned}$$

◇

### 3.6 Illustration

Dans cette partie, on illustre notre approche en présentant une application relatif à un processus de panification. Pour avoir plus de détails sur le fonctionnement de ce processus, le lecteur pourra consulter les documents [Gros 2000] [R. Jeantet 2006].

#### a. Processus

On considère une boulangerie semi-industrielle qui fabrique deux types de pains, Pain 1 (qualité supérieure) et Pain 2 (qualité ordinaire). Le procédé de fabrication est représenté par le graphe d'événements P-temporel de la Figure (5.2). Les deux types de pains doivent suivre la même séquence de fabrication sur les quatre machines disponibles : le pétrin (M1), la diviseuse (M2), la façonneuse (M3) et le four (M4).

$$\text{Pains 1 et 2 : } M1 \rightarrow M2 \rightarrow M3 \rightarrow M4.$$

Les durées différentes entre les deux gammes, ainsi qu'une opération de pointage supplémentaire (place  $P21$ ) pour la gamme 2, sont à l'origine d'une qualité différente. Les opérations sont les suivantes :

**Pétrissage.** Le pétrissage commence par un mélange de tous les ingrédients dans un pétrin, cette opération durant de 15 à 27 minutes. Cette étape est représentée par les places  $P11$  et  $P20$ .

**Pointage.** C'est la première fermentation. La pâte est laissée au repos de 10 à 30 minutes pour permettre à la levure, ou le levain, d'assurer une fermentation suffisante. Cette étape ne concerne que le Pain 2 et elle est représentée par  $P21$  dans le graphe.

**Division.** Il s'agit de découper la pâte, en pâtons plus petits avec la diviseuse. Dans le graphe, cela est représenté par  $P12$  pour le Pain 1 et  $P22$  pour le Pain 2. Deux diviseuses sont disponibles.

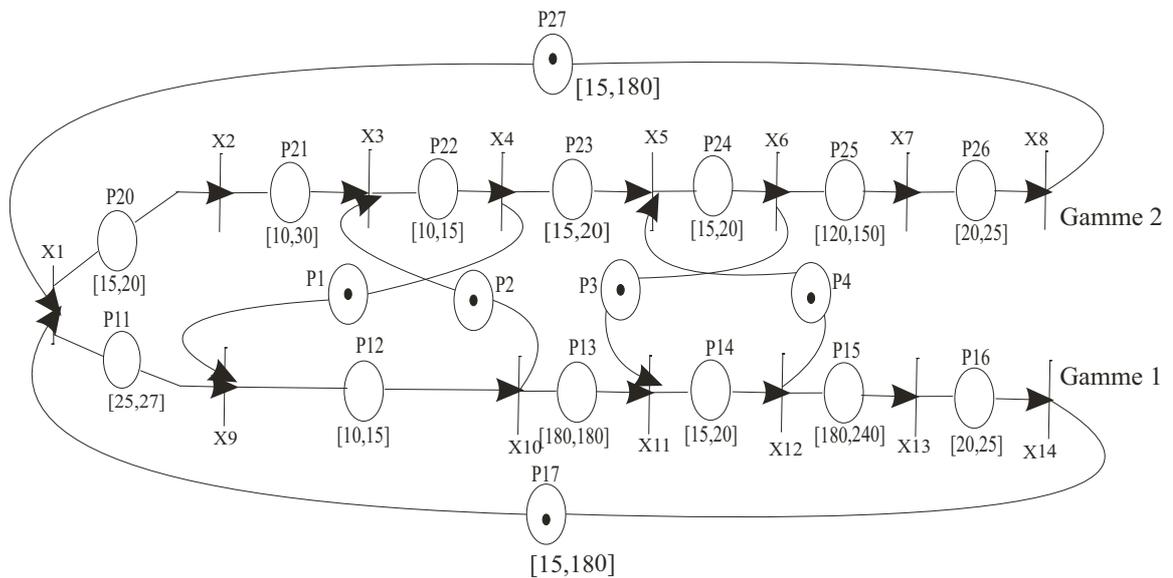
**Détente.** Les pâtons sont laissés au repos une nouvelle fois. Cette étape dure 180 minutes pour le Pain 1 ( $P13$ ) et de 15 à 20 minutes pour le Pain 2 ( $P23$ ).

**Façonnage.** Chaque pâton est retravaillé pour lui donner la forme voulue. On utilise pour cela une façonneuse. Dans le graphe, cela est représenté par  $P14$  pour le Pain 1 et  $P24$  pour le Pain 2. Deux façonneuses sont disponibles.

**Apprêt.** Les pâtons sont disposés sur une toile de lin pour la deuxième fermentation. L'apprêt (repos) dépend du type de Pain et est représenté par les places  $P15$  et  $P25$ .

**Cuisson.** Les pains sont enfournés à  $250 - 280^{\circ}\text{C}$  dans le four (supposé suffisamment grand). La cuisson dure de 20 à 25 minutes et est identique pour les deux gammes. Cette opération est représentée par  $P16$  pour le Pain 1 et  $P26$  pour le Pain 2.

**Sortie des chariots.** Elle est représentée par  $P17$  et  $P27$ .



**Fig. 3.3 :** Système de production du pain.

## b. Modèle

Les matrices d'incidence  $G_1$ ,  $G_0$ , du graphe d'événements p-temporel de taille  $(19 \times 14)$  sont comme suit :





également proposée sous une forme plus réduite. Si la notion de structure critique du temps de cycle minimal est bien connue pour les GET, cette notion est généralisée pour les graphes d'événements P-temporels.

## Chapitre 4

# Standardisation et trajectoire au plus tôt

### 4.1 Introduction

Les parties (2.4.4) et (2.5.3.2) du chapitre 2 proposent des modèles des GET et graphes P-temporels dans l'algèbre conventionnelle. Le résultat est un modèle dynamique défini par un polyèdre  $A.x \leq b$ . Les objectifs de cette partie sont les suivants :

1. On souhaite compléter le modèle initial en introduisant des transitions d'entrée et de sortie du GET. De plus, on introduit une technique de standardisation du modèle telle que chaque place interne contienne seulement un jeton. L'objectif est d'avoir un modèle dont le fonctionnement est proche de celui de l'équation d'état de l'automatique classique. Cette partie a naturellement déjà été développée dans l'algèbre  $(\max,+)$  avec l'étoile de Kleene [Baccelli 1992].
2. On présente le fonctionnement au plus tôt du GET où les entrées sont connues, on discute alors de la technique de résolution. On montre que la résolution peut être réalisée dans l'algèbre standard de manière analogue à l'algèbre  $(\max,+)$  avec la résiduation [Baccelli 1992],[Cottenceau 1999].

Le chapitre suit le plan suivant. On complète la partie 2.5.3.2 du chapitre 2, en présentant le modèle d'entrées/sorties et l'approche de standardisation du modèle. Ensuite, on applique le modèle pour calculer la trajectoire au plus tôt du GET. Un exemple simple représentant une chaîne de production illustre l'approche.

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

Pour les points 1 et 2 cités au dessus, il est bien connu que cela peut être réalisé en utilisant l'algèbre  $(\max,+)$ . Rappelons que les GET peuvent être modélisés sous forme de systèmes

linéaires stationnaires dans différents dioïdes. Plus précisément, le fonctionnement d'un GET peut être décrit par des équations linéaires dans les dioïdes  $\mathbb{R}_{max}$  ou  $\mathbb{Z}_{min}$ , lesquelles peuvent se mettre sous la forme du modèle d'état suivant :

$$\begin{cases} x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \\ y(k) = C \otimes x(k) \end{cases} .$$

Notons que ce modèle présente des entrées/sorties, que la matrice  $A$  correspond à un marquage unitaire, l'égalité correspondant au fonctionnement au plus tôt. Nous désirons à présent obtenir le même résultat mais dans l'algèbre standard.

### 4.3 Le modèle entrées/sorties pour un graphe d'événements temporisé

Nous partons de l'équation (2.9) où l'équation a été réécrite sur un horizon réduit (la technique utilisée est détaillée dans (chapitre 2 §2.4.4)). L'objectif de cette section est de réécrire le modèle initial 2.8 sur un horizon réduit dont la forme est proche de celle de l'équation d'état de l'automatique classique. Ce modèle est appelé *modèle entrées/sorties*.

A partir du GET, on définit trois types de transitions : les transitions internes, d'entrées et de sorties. Ces transitions, sont notées  $x$ ,  $u$  et  $y$  respectivement. Les ensembles correspondant à ces transitions sont notés comme suit :  $Tr_x$ ,  $Tr_u$  et  $Tr_y$ .

L'ensemble des places  $P$  est l'union de trois ensembles, à savoir :  $P = P_{u \rightarrow x} \cup P_{x \rightarrow x} \cup P_{x \rightarrow y}$  où :

- $P_{u \rightarrow x}$  est l'ensemble des places entre les transitions d'entrées et celles internes ;
- $P_{x \rightarrow x}$  est l'ensemble des places entre les transitions internes ;
- $P_{x \rightarrow y}$  est l'ensemble des places entre les transitions internes et celles de sorties.

Dans un temps premier, on commence par les inégalités telles que les valeurs de la commande  $u(k)$  et l'état antérieur  $x(k-1)$  peuvent modifier l'état actuel  $x(k)$ . L'écriture matricielle des inégalités internes est donnée comme suit :

$$\begin{pmatrix} A_{.,1} & A_{.,0} \end{pmatrix} \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq -T^A, \quad (4.1)$$

où la notation  $A_{.,1}$  (respectivement  $A_{.,0}$ ) associe les colonnes correspondant aux composantes de  $x(k-1)$  (respectivement,  $x(k)$ ) sans restriction sur les lignes.

Ensuite, pour un marquage initial de  $P_{u \rightarrow x}$  nul, les inégalités d'entrée sont les suivantes :

$$\begin{pmatrix} B_1 & B_0 \end{pmatrix} \begin{pmatrix} u(k) \\ x(k) \end{pmatrix} \leq -T^B. \quad (4.2)$$

Enfin, pour un marquage initial de  $P_{x \rightarrow y}$  nul aussi, les inégalités qui déterminent la sortie à partir de l'état sont :

$$\begin{pmatrix} C_1 & C_0 \end{pmatrix} \begin{pmatrix} x(k) \\ y(k) \end{pmatrix} \leq -T^C. \quad (4.3)$$

Les coefficients des matrices  $A = \begin{pmatrix} A_{.,1} & A_{.,0} \end{pmatrix}$ ,  $B = \begin{pmatrix} B_1 & B_0 \end{pmatrix}$  et  $C = \begin{pmatrix} C_1 & C_0 \end{pmatrix}$  sont :  $\{0, 1, -1\}$ .

Dans le but de simplifier la présentation, on suppose qu'il n'y a pas de rapport direct entre les entrées et les sorties, ce qui correspond à :

$$\begin{pmatrix} D_1 & D_0 \end{pmatrix} \begin{pmatrix} u(k) \\ y(k) \end{pmatrix} \leq -T^D. \quad (4.4)$$

Ce cas correspond à l'existence d'une place entre une transition d'entrée et de sortie. Une simple introduction d'une transition interne donne le GET souhaité, aussi l'inégalité ci-dessus peut être réécrite sous les formes (4.1),(4.2) et (4.3).

Ci-dessous, nous réécrivons les relations (4.1) dans l'ordre numérique du marquage initial des places. La ligne supérieure correspond à un marquage initial nul. Le premier indice de  $A_{M_0, \cdot}$  : correspond au marquage initial  $M_0$  (dans ce cas, soit 0, soit 1) tandis que le second indice de  $A_{\cdot, \Delta}$  correspond au numéro du déplacement  $\Delta$  (en fonction du marquage, soit 0, soit 1) qui donne  $x(k - \Delta)$ . On a :

$$\begin{pmatrix} A_{0,1} & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix} \cdot \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq \begin{pmatrix} -T_0^A \\ -T_1^A \end{pmatrix}, \quad (4.5)$$

avec  $A_{0,1} = 0$ .

## 4.4 Modèle entrées/sorties standardisé pour un GET

L'objectif de cette section est d'obtenir un nouveau modèle à partir du modèle entrées/sorties initial (4.1),(4.2) et (4.3) avec les caractéristiques suivantes :

- Un graphe d'événement peut être déduit directement à partir des matrices d'incidences d'entrées et de sorties du nouveau modèle. Chaque place interne contient un jeton (et un seul).
- Par conséquent, ce nouveau modèle permet un calcul efficace de l'état, connaissant l'état passé et la commande : elle évite la répétition des mêmes calculs dans le calcul itératif de l'état.

Sans restriction de généralité, nous supposons que le marquage initial des places  $P_{u \rightarrow x}$  et  $P_{x \rightarrow y}$  est nul. Cela peut s'obtenir simplement en ajoutant une place de temporisation et de marquage initial nuls aux transitions d'entrées et de sorties le nécessitant. Nous écrivons ci-dessous les inégalités à obtenir et qui expriment comme ci-dessus les relations entre les dates de tir des transitions internes, d'entrées et de sorties pour  $P_{x \rightarrow x}$ ,  $P_{u \rightarrow x}$  et  $P_{x \rightarrow y}$ , respectivement.

La relation entre  $x(k-1)$  et  $x(k)$  sera de la forme :

$$\begin{pmatrix} W_{x \rightarrow x}^+ & -W_{x \rightarrow x}^- \end{pmatrix} \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq (-T_{x \rightarrow x}). \quad (4.6)$$

La relation entre  $u(k)$  et  $x(k)$  sera de la forme :

$$\begin{pmatrix} W_{u \rightarrow x}^+ & -W_{u \rightarrow x}^- \end{pmatrix} \begin{pmatrix} u(k) \\ x(k) \end{pmatrix} \leq (-T_{u \rightarrow x}), \quad (4.7)$$

et la relation entre  $y(k)$  et  $x(k)$  sera de la forme :

$$\begin{pmatrix} W_{x \rightarrow y}^+ & -W_{x \rightarrow y}^- \end{pmatrix} \begin{pmatrix} x(k) \\ y(k) \end{pmatrix} \leq (-T_{x \rightarrow y}), \quad (4.8)$$

où les dimensions des matrices d'incidence sont :

- $\dim(W_{x \rightarrow x}^+) = \dim(W_{x \rightarrow x}^-) = |P_{x \rightarrow x}| \cdot |Tr_x|$ ,
- $\dim(W_{u \rightarrow x}^+) = |P_{u \rightarrow x}| \cdot |Tr_u|$ ,
- $\dim(W_{u \rightarrow x}^-) = |P_{u \rightarrow x}| \cdot |Tr_x|$ ,
- $\dim(W_{x \rightarrow y}^+) = |P_{x \rightarrow y}| \cdot |Tr_x|$ ,
- $\dim(W_{x \rightarrow y}^-) = |P_{x \rightarrow y}| \cdot |Tr_y|$ .

#### Remarque 4.1

Aucune ligne des matrices d'incidence est nulle. De même, aucune colonne de  $W_{u \rightarrow x}^+$  et  $W_{x \rightarrow y}^-$  est nulle, sachant que chaque transition d'entrée et sortie est supposée reliée au graphe.  $\diamond$

Le système d'inégalités (4.6) correspond à un marquage initial égal à un, et les systèmes (4.7) et (4.8) à un marquage initial nul. Ainsi, les variables sont reliées par des matrices d'incidences d'entrées et de sorties qui permettent un calcul simple. Le GET correspondant pourra être construit, les matrices d'incidence de l'équation fondamentale de marquage sont :

$$W^+ = \begin{pmatrix} W_{u \rightarrow x}^+ & 0 & 0 \\ 0 & W_{x \rightarrow x}^+ & 0 \\ 0 & W_{x \rightarrow y}^+ & 0 \end{pmatrix} \text{ et } W^- = \begin{pmatrix} 0 & W_{u \rightarrow x}^- & 0 \\ 0 & W_{x \rightarrow x}^- & 0 \\ 0 & 0 & W_{x \rightarrow y}^- \end{pmatrix},$$

pour le vecteur des transitions  $\begin{pmatrix} u^t & x^t & y^t \end{pmatrix}^t$ . Les temporisations sont  $T_{u \rightarrow x}$ ,  $T_{x \rightarrow x}$  et  $T_{x \rightarrow y}$  et chaque place interne (respectivement, d'entrée et de sortie) présente un marquage initial égal à un (respectivement, égal à zéro).

Comme le marquage initial des places d'entrées/sorties est nul, la détermination du système (4.7) et (4.8) est immédiate. Comme le système (4.3) exprime les places reliant les transitions internes aux transitions de sortie, les matrices  $C_0$  et  $C_1$  ont la forme de la matrice d'incidence qui facilite le calcul de  $y(k)$ , car chaque ligne de  $C_0$  et  $C_1$  a un coefficient unique non-nul. On a  $C_1 = W_{x \rightarrow y}^+$ ,  $C_0 = -W_{x \rightarrow y}^-$  et  $T^C = T_{x \rightarrow y}$ .

L'établissement du système (4.6) est plus difficile puisque le marquage initial de chaque place interne du système (4.5) peut être égal à zéro ou un. En particulier, le calcul itératif de  $x(k)$  doit

être fait sans tenir compte des relations liant les composantes du vecteur d'état  $x(k)$  pour un  $k$  donné.

$$A_{0,0}.x(k) \leq -T_0^A. \quad (4.9)$$

La détermination algébrique des matrices d'incidence et les temporisations du système (4.6) à partir du système composé des inéquations (4.1), (4.2 et (4.3) est le but de la partie suivante.

#### 4.4.1 Technique en utilisant la programmation linéaire

Soit la transition  $x_i$  et les systèmes (4.2) et (4.5). Considérons les effets possibles sur la date  $x_i(k)$  produite par :

- Les dates de tir de la transition de commande  $u_j(k)$  (**cas a**)
- Les dates de tir des transitions en amont des places dont le marquage initial égal à un correspond à  $x_j(k-1)$  (**cas b**).

Nous nous concentrons sur le comportement au plus tôt.

- **Cas a**) : l'effet minimal est la différence minimale  $x_i(k) - u_j(k)$ .

Cette différence correspond à la fonction objectif suivante :  $\min(c'x)$ , où  $c'$  est un vecteur ligne nul sauf  $c'_i = 1$  et  $c'_j = -1$  ( $i$  et  $j$  sont respectivement l'indice de la transition interne  $x_i$  et la transition d'entrée  $u_j$ ) pour les contraintes suivantes :

$$\begin{pmatrix} 0 & A_{0,0} \\ B_1 & B_0 \end{pmatrix} \begin{pmatrix} u(k) \\ x(k) \end{pmatrix} \leq \begin{pmatrix} -T_0^A \\ -T^B \end{pmatrix}, \quad (4.10)$$

ce qui est un problème classique de programmation linéaire.

Pour chaque couple  $(x_i; u_j)$ , la résolution de ce problème donne une différence minimale  $\Delta T$  (précisément,  $\Delta T_{x_i(k), u_j(k)}$ ). Nous pouvons écrire la relation  $x_i(k) - u_j(k) \geq \Delta T$ , où l'expression plus habituelle  $x_i(k) \geq u_j(k) + \Delta T$ . Dans le nouveau graphe, elle correspond à une place entre la transition entrante  $u_j$  et celui de la transition sortante  $x_i$  avec une temporisation  $\Delta T$  et un marquage initial nul.

- **Cas b**) l'effet minimal est la différence minimale  $x_i(k) - x_j(k-1)$ .

Cette différence correspond à la fonction objectif suivante :  $\min(c'x)$ , où  $c'$  est un vecteur ligne nul sauf  $c'_i = 1$  et  $c'_j = -1$  ( $i$  et  $j$  sont les indices de de la transition sortante  $x_i$  et celui de la transition entrante  $x_j$ ) pour les contraintes suivantes :

$$\begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix} \cdot \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq \begin{pmatrix} -T_0^A \\ -T_1^A \end{pmatrix}, \quad (4.11)$$

qui est également un problème classique de programmation linéaire.

Pour chaque couple  $(x_i, x_j)$ , la résolution de ce problème donne une différence minimale  $\Delta T$  (précisément,  $\Delta T_{x_i(k), x_j(k-1)}$ ). Nous pouvons écrire la relation  $x_i(k) - x_j(k-1) \geq \Delta T$ , où l'expression plus habituelle  $x_i(k) \geq x_j(k-1) + \Delta T$ . Dans le nouveau graphe, elle correspond à

une place entre la transition d'entrée  $x_j$  et la transition sortante  $x_i$  avec une temporisation  $\Delta T$  et un marquage initial égal à un.

Les deux procédures permettent la considération de tous les effets sur chaque transition  $i$  produites par la commande  $u(k)$  et une évolution du passé exprimée par  $x(k-1)$ . L'existence d'un effet peut être expliqué graphiquement par **l'existence d'un chemin entre les transitions**. Si les deux procédures sont répétées pour chaque transition  $x_i$ , les relations (4.1) et (4.2) peuvent être remplacées par le nouveau système (4.6) et (4.7). En particulier, le système (4.9), soit  $A_{0,0}.x(k) \leq -T_0^A$ , peut désormais être écarté, car les effets possibles sur les dates de tirs de  $x_i$  sont considérés dans (4.6) et (4.7).

Pour résumer, l'approche génère un nouveau GET qui ne contient pas de places internes dont le marquage initial est nul. Il contient un ensemble de places dont le nombre est inférieur à celui de  $card(x)(card(x) + card(u))$ .

#### 4.4.2 Technique duale

Dans cette section, nous proposons une deuxième résolution algébrique du problème basée sur la dualité (voir 1.32).

- Dans le cas b) de la partie précédente, nous cherchons la différence minimale  $x_i - x_j$  ou, plus formellement, nous désirons résoudre le problème défini comme suit :

$min(c'.x)$ , où  $c'$  est un vecteur ligne nul, sauf  $c'_i = 1$  et  $c'_j = -1$ , avec les contraintes :

$$\begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix} \cdot \begin{pmatrix} x(k-1) \\ x(k) \end{pmatrix} \leq \begin{pmatrix} -T_0^A \\ -T_1^A \end{pmatrix}. \quad (4.12)$$

Comme la minimisation de  $c'.x$  correspond à la maximisation de  $-c'.x$ , ce problème correspond à (D) où :

$$A = \begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix}, b = \begin{pmatrix} -T_0^A \\ -T_1^A \end{pmatrix} \text{ et } c = -c'.$$

Par conséquent, le problème primal est le suivant :

$$\min y \cdot \begin{pmatrix} -T_0^A \\ -T_1^A \end{pmatrix} \text{ avec } y \in R_+^n \text{ et } y \cdot \begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix} = -c'.$$

Ce problème est la maximisation du produit de  $y$  par un vecteur de temporisation  $\begin{pmatrix} T_0^A \\ T_1^A \end{pmatrix}$

sous la contrainte  $y \cdot \begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix} = -c'$ ,

où  $c'$  est un vecteur ligne nul, sauf  $c'_i = 1$  et  $c'_j = -1$  ( $i$  et  $j$  sont respectivement les indices de la transition sortante  $x_i$  et la transition entrante  $x_j$ ).

- Symétriquement, le problème primal correspond le cas a) est comme suit :

$$\max y \cdot \begin{pmatrix} T_{A,m=0} \\ T_{u \rightarrow x} \end{pmatrix} \text{ avec } y \in R_+^m \text{ et}$$

$$y \cdot \begin{pmatrix} 0 & A_{0,0} \\ B_1 & B_0 \end{pmatrix}, \quad (4.13)$$

où  $c'$  est un vecteur ligne nul sauf  $c'_i = 1$  et  $c'_j = -1$  ( $i$  et  $j$  sont respectivement les indices de la transition interne  $x_i$  et la transition d'entrée  $u_j$ ).

**Analyse de la solution** Le théorème suivant relatif à la matrice d'incidence permet l'analyse de la solution optimale  $y_{opt}$ .

**Théorème 4.2.** ([Gondran 1979] §chapitre 1) Une matrice d'incidence est totalement unimodulaire, c'est-à-dire, que toute sous-matrice carrée extraite de cette matrice a un déterminant égal à  $-1, 0$  ou  $1$ .

**Corollaire 4.3.** La matrice d'incidence  $\begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix}$  est totalement unimodulaire. ■

Nous pouvons maintenant analyser la solution  $y_{opt}$ .

**Théorème 4.4.** (Hoffman-Kruskal 1956 (Annexe 2 dans [Gondran 1979])) La solution optimale d'un programme linéaire ( $P$ ) est du type entier pour tout vecteur  $c$  à coefficients entiers et tout vecteur coût  $b$  si, et seulement si, la matrice  $A$  est totalement unimodulaire.

**Corollaire 4.5.** La solution optimale  $y_{opt}$  d'un programme linéaire ( $P$ ) et la fonction objective

$y_{opt} \cdot \begin{pmatrix} T_0^A \\ T_1^A \end{pmatrix}$  sont entiers.

En outre, rappelons que les coefficients de  $y_{opt}$  sont positifs :  $y_{opt} \geq 0$ .

#### 4.4.3 Lien avec la théorie des graphes

Nous allons montrer que l'approche duale correspond à la détermination du plus grand chemin dans la théorie des graphes. Nous avons  $y \cdot AW = -c'$  avec  $AW = \begin{pmatrix} 0 & A_{0,0} \\ A_{1,1} & A_{1,0} \end{pmatrix}$ ,

où  $c'$  un vecteur ligne nul sauf  $c'_e = 1$  ( $e$  pour *end*) et  $c'_s = -1$  ( $s$  pour *start*) et,  $e$  et  $s$  sont respectivement les indices de la transition sortante  $x_e$  et la transition entrante  $x_s$ .

Nous considérons maintenant l'existence d'un chemin allant de  $s$  à  $e$ . Pour chaque produit  $y \cdot AW$ , chaque  $y_i \neq 0$  sélectionne une ligne  $i$  de  $AW$  et la place appropriée. Nous savons maintenant que  $y$  est entier. Considérons la colonne  $x_s$ . Le produit  $y \cdot AW_{.,s} = -c'_s = 1$  montre qu'il y a une place  $i_1$  telle que  $AW_{i_1,s} = 1$  : il correspond à un coefficient positif ( $y_{i_1} \geq 1$  tel que  $y$  est un entier). Graphiquement, il correspond à la place  $i_1$  (avec en amont la transition  $x_s$ ) qui est la première place du chemin.

Considérons la transition sortante  $j_1$  de la place  $i_1$  et la colonne correspondante  $x_{j_1}$ . Cette transition existe comme le graphe est un GET. Alors,  $AW_{i_1, j_1} = -1$ . Si le produit  $y \cdot AW_{\cdot, j_1} = 0$ , alors il y a au moins un coefficient positif dans la colonne  $j_1$  et une place en aval  $i_2$  de la transition  $j_1$ . Dans cette étape, un couple de coefficients négatif et positif de cette colonne est sélectionné. Graphiquement, cette étape correspond à la transition  $j_1$  avec la place  $i_1$  en amont et une place  $i_2$  en aval. Comme  $y_{i_1} \geq 1$ , il y a  $y_{i_2} \geq 1$ .

Comme ci-dessus, la place  $i_2$  a une transition aval  $j_2$ , ainsi on peut considérer la colonne correspondante et répéter la procédure qui génère une nouvelle place si le produit  $y \cdot AW_{\cdot, j_2}$  est nul. Ce dernier s'arrête lorsqu'il y produit.  $y \cdot W_{\cdot, e} = -1$ , ce qui correspond à la dernière transition  $x_e$ . Ainsi, les coefficients de  $y$  sont positifs ou nuls, et décrivent un chemin allant de  $s$  à  $e$ .

Nous allons montrer que le chemin est unique. Le vecteur  $y$  peut choisir un ensemble de places incluant les places du chemin. Cependant, d'autres places peuvent être choisies. Prenons la colonne de la transition  $x_1$ , où  $y \cdot AW_{\cdot, 1} = 0$ . Supposons que  $y$  sélectionne une place en aval de la transition  $x_1$ , notée  $k_1$  différente de  $i_1$  qui n'appartient pas au chemin décrit au-dessus. La condition :  $y \cdot AW_{\cdot, 1} = 0$  ne peut pas être satisfaite comme  $y_{i_1} \geq 1$  et  $y_{k_1} \geq 1$ . La seule possibilité est l'existence d'une place en amont de la transition  $x_1$ . La structure du GET implique qu'une transition entrante est nécessaire. Comme les produits correspondants doivent être nuls, de nouvelles places sont nécessaires et la procédure ne peut que créer un circuit mais il n'est pas possible que le problème primal recherche le poids maximal.

Par conséquent, le vecteur d'entiers  $y$  peut seulement choisir un unique chemin de la transition  $x_s$  vers la transition  $x_e$  et ses coefficients sont égaux à zéro ou un. Pour résumer, l'algorithme de programmation linéaire choisit le plus grand chemin parmi les chemins possibles allant de  $s$  vers  $e$ .

#### 4.4.4 Illustration

Cet exemple décrit un processus de la fabrication de draps. Utilisant l'approche PERT, le tableau suivant présente l'ensemble des tâches :

Label	Description de la tâche	Durée	tâche précédente
A	impression	10	-
B	pliage	8	A
C	emballage	3	B

Une autre description basée sur les transitions est choisie dans l'approche de Roy [Gondran 1979]. Le franchissement de chaque transition correspond au début de la tâche. Le tableau correspondant est comme suit :

Label	Description de l'événement	Durée de la tâche commencée	Événement suivant
$u$	introduire la matière première dans le processus	0	$x_1$
$x_1$	début de A	10	$x_1(k+1), x_2$
$x_2$	début de B	8	$x_2(k+1), x_3$
$x_3$	début de C	3	$x_3(k+1), x_4$
$y$	fin d'un processus	-	-

A partir de la description des transitions du tableau précédent, on obtient le GET décrit dans la figure 4.1. :

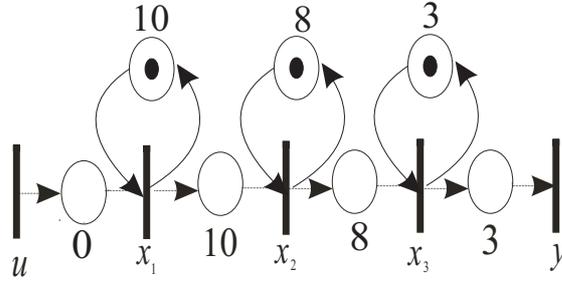


Fig. 4.1 : Processus du fabrication de draps.

Notons que le graphe d'événements déduit est plus petit que le graphe d'événement déduit à partir du premier tableau des tâches. Les matrices d'incidences associées sont comme suit :

$$A_{0,1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_{0,0} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}, A_{1,1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, A_{1,0} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

$$B_1 = 1, B_0 = \begin{pmatrix} -1 & 0 & 0 \end{pmatrix}, C_1 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \text{ et } C_0 = -1. T_0^A = \begin{pmatrix} 10 \\ 8 \end{pmatrix}, T_1^A = \begin{pmatrix} 10 \\ 8 \\ 3 \end{pmatrix},$$

$T^B = 0$  et  $T^C = 3$ .

a) la différence minimale

$$x_i(k) - u_j(k) \text{ est } \begin{pmatrix} 0 \\ 10 \\ 18 \end{pmatrix},$$

alors,

$$W_{u \rightarrow x}^+ = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, -W_{u \rightarrow x}^- = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \text{ et } -T_{u \rightarrow x} = \begin{pmatrix} 0 \\ -10 \\ -18 \end{pmatrix}.$$

b) la différence minimale  $x_i(k) - x_j(k-1)$  est  $\begin{pmatrix} 10 & -\infty & -\infty \\ 20 & 8 & -\infty \\ 28 & 16 & 3 \end{pmatrix}$ , alors,  $W_{x \rightarrow x}^+ = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ ,

$$-W_{x \rightarrow x}^- = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -1 \end{pmatrix} \text{ et } -T_{x \rightarrow x} = \begin{pmatrix} -10 \\ -20 \\ -28 \\ -8 \\ -16 \\ -3 \end{pmatrix}.$$

Comme conséquence, les inégalités d'entrée et internes peuvent être remplacées par des relations réduites. Les inégalités de sortie sont conservées. Par conséquent, le nouveau GET est entièrement caractérisé.

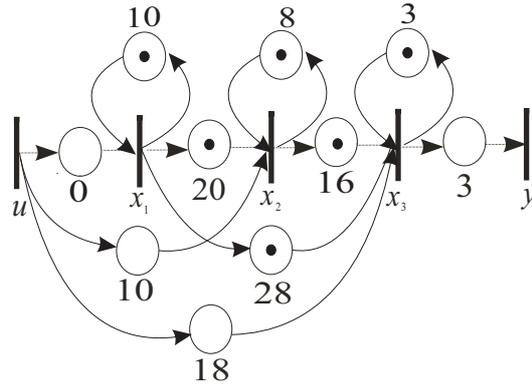


Fig. 4.2 : GET final.

## 4.5 Trajectoire au plus tôt d'un GET :

Cette partie traite de la trajectoire au plus tôt en utilisant le modèle d'entrées/sorties proposé dans la partie 4.3 et certains concepts relatifs aux systèmes monotones : suite à l'ordre composante par composante, la trajectoire au plus tôt est la plus petite des trajectoires possibles. Dans l'algèbre  $(\max, +)$ , il est bien connu que le GET a une trajectoire au plus tôt, exprimée par une égalité. Nous donnons deux techniques de résolution de ce problème.

Les relations suivantes sont déduites à partir des inégalités d'entrée (4.6) (4.7). La deuxième ligne de (4.14) exprime la caractéristique non-décroissante de l'état.

Pour  $k \geq 1$ , on a :

$$\begin{pmatrix} W_{u \rightarrow x}^- \\ I \\ W_{x \rightarrow x}^- \end{pmatrix} .x(k) \geq \begin{pmatrix} T_{u \rightarrow x} \\ 0 \\ T_{x \rightarrow x} \end{pmatrix} + \begin{pmatrix} 0 & W_{u \rightarrow x}^+ \\ I & 0 \\ W_{x \rightarrow x}^+ & 0 \end{pmatrix} \begin{pmatrix} x(k-1) \\ u(k) \end{pmatrix}, \quad (4.14)$$

avec l'état initial  $x(0) = x_0$  dans  $\mathbb{R}$ . Le système linéaire d'inégalités (4.14) est exprimé sous la forme  $A^{fw} .x(k) \geq b^{fw}(x(k-1), u(k))$  (abréviation *fw* pour *ForWard*) avec :

$$A^{fw} = \begin{pmatrix} W_{u \rightarrow x}^- \\ I \\ W_{x \rightarrow x}^- \end{pmatrix},$$

$$b^{fw}(x(k-1), u(k)) = \begin{pmatrix} T_{u \rightarrow x} \\ 0 \\ T_{x \rightarrow x} \end{pmatrix} + \begin{pmatrix} 0 & W_{u \rightarrow x}^+ \\ I & 0 \\ W_{x \rightarrow x}^+ & 0 \end{pmatrix} \begin{pmatrix} x(k-1) \\ u(k) \end{pmatrix} \forall k \in k \geq 1.$$

En outre, l'inégalité de sortie (4.8) doit être prise en compte. Le système correspond est  $A^{fw'} .y(k) \geq b^{fw'}(x(k))$  avec

$$A^{fw'} = W_{x \rightarrow y}^- \text{ et } b^{fw'}(x(k)) = T_{x \rightarrow y} + W_{x \rightarrow y}^+ .x(k) \text{ pour } k \geq 1. \quad (4.15)$$

#### 4.5.1 La technique algébrique

Connaissant l'état initial  $x(0) = x_0$  et la trajectoire de commande  $u$ , le résultat suivant prouve l'existence de la trajectoire minimum d'état et donne l'expression des trajectoires d'état et de sortie. Les matrices  $A^{fw}$ ,  $A^{fw'}$  et les vecteurs  $b^{fw}$ ,  $b^{fw'}$  sont définis au-dessus.

**Théorème 4.6.** - *Connaissant la commande  $u$ , il y a un état minimum, noté  $x^-$ , et une sortie minimale, notée  $y^-$ , pour l'état initial  $x(0) = x_0$ .*

- *La trajectoire d'état au plus tôt est définie par*

$$x_i^-(k) = \bigvee_{j|A_{j,i}^{fw}=1} b_j^{fw}(x^-(k-1), u(k)), \quad \text{pour } k \geq 1.$$

- *La trajectoire de sortie au plus tôt est définie par*

$$y_i^-(k) = \bigvee_{j|A_{j,i}^{fw'}=1} b_j^{fw'}(x^-(k)), \quad \text{pour } k \geq 1.$$

**Preuve :**

- Le système  $A^{fw} .x \geq b^{fw}(x(k-1), u(k))$  est inf-monotone puisque chaque ligne de la matrice  $A^{fw}$  a un coefficient strictement positif.

- Cet ensemble est non-vide : une valeur arbitraire  $x(k)$  dans  $\mathbb{R}^n$  pour tout état initial et de commande peut facilement être calculée puisque chaque ligne de la matrice  $A^{fw}$  contient seulement un coefficient strictement positif  $A_{j,i}^{fw} = 1$ . En effet,  $A_{j,i}^{fw} \cdot x_i(k) \geq b_j^{fw}(x(k-1), u(k))$  pour tout  $A_{j,i}^{fw} \neq 0$  et,  $x_i(k) \geq \bigvee_{A_{j,i}=1} b_j^{fw}(x(k-1), u(k))$ .

- De plus, la trajectoire arbitraire de l'état a un minorant. En effet, le vecteur  $x_0$  est un minorant de  $x(1)$  puisque la trajectoire d'état est non-décroissante : cette propriété est exprimée dans les inégalités d'état.

Finalement, le théorème (1.49) montre que l'espace d'état a une trajectoire d'état minimum  $x^-(k)$  pour  $k \geq 1$ . C'est-à-dire, le GET a une trajectoire d'état au plus tôt définie par  $x_i^-(k) = \bigvee_{j|A_{j,i}=1} b_j^{fw}(x^-(k-1), u(k))$  pour  $k \geq 1$ . Cette trajectoire d'état au plus tôt est de plus unique.

Le même raisonnement peut être appliqué pour l'inégalité de sortie  $A^{fw'} \cdot y(k) \geq b^{fw'}(x^-(k))$  comme  $x^-(k)$  est connue. Ce système est inf-monotone, et non vide puisque une trajectoire arbitraire  $y(k)$ , pour tout  $k$ , peut facilement être calculée et a un minorant puisque la trajectoire d'état a un minorant ( $b^{fw'}(x(k)) \geq T_{x \rightarrow y} + W_{x \rightarrow y}^+ \cdot x_0$ ). Alors le graphe d'événement temporisé a une unique trajectoire de sortie au plus tôt  $y^-$ . En conséquence, l'état et sortie minimum existent connaissant l'état initial  $x_0$  et la commande  $u$ .

□

#### Exemple 4.7

Soit le graphe d'événements temporisé :

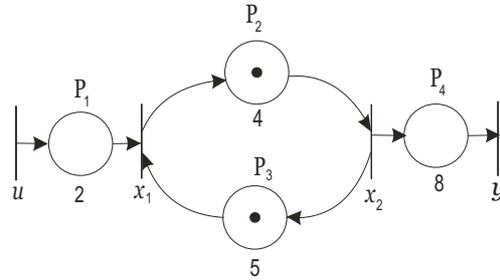


Fig. 4.3 : Graphe d'événements temporisé.

$P = P_{u \rightarrow x} \cup P_{x \rightarrow x} \cup P_{x \rightarrow y}$  où  $P_{u \rightarrow x} = \{p_1\}$ ,  $P_{x \rightarrow x} = \{p_2, p_3\}$  et  $P_{x \rightarrow y} = \{p_4\}$ .

$TR = Tr_u \cup Tr_x \cup Tr_y$  où  $Tr_u = \{u\}$ ,  $Tr_x = \{x_1, x_2\}$  et  $Tr_y = \{y\}$ .

Pour un marquage initial  $M_0 = (0, 1, 1, 0)^t$ , le modèle entrées/sorties est comme suit. Avec

l'état  $x(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix}$ ,

$$W_{x \rightarrow x}^+ = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, W_{x \rightarrow x}^- = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ et } T_{x \rightarrow x} = \begin{pmatrix} 4 \\ 5 \end{pmatrix},$$

$$W_{u \rightarrow x}^+ = 1, W_{u \rightarrow x}^- = \begin{pmatrix} 1 & 0 \end{pmatrix} \text{ et } T_{u \rightarrow x} = 2,$$

$W_{x \rightarrow y}^+ = \begin{pmatrix} 0 & 1 \end{pmatrix}$ ,  $W_{x \rightarrow y}^- = 1$  et  $T_{x \rightarrow y} = 8$ .  
 Comme  $\begin{pmatrix} T_{u \rightarrow x} \\ 0 \\ T_{x \rightarrow x} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 4 \\ 5 \end{pmatrix}$  et  $\begin{pmatrix} 0 & W_{u \rightarrow x}^+ \\ I & 0 \\ W_{x \rightarrow x}^+ & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ ,  $b^{fw}(x(k-1), u(k)) =$   
 $\begin{pmatrix} 2 + u(k) \\ x_1(k-1) \\ x_2(k-1) \\ 4 + x_1(k-1) \\ 5 + x_2(k-1) \end{pmatrix}$ . Comme  $A^{fw} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$ , le vecteur d'état au plus tôt  $\begin{pmatrix} x_1^-(k) \\ x_2^-(k) \end{pmatrix}$  est  
 $\begin{pmatrix} (2 + u(k)) \vee x_1^-(k-1) \vee (5 + x_2^-(k-1)) \\ x_2^-(k-1) \vee (4 + x_1^-(k-1)) \end{pmatrix}$ .  
 Comme  $A^{fw'} = 1$  et  $b^{fw'}(x(k)) = 8 + \begin{pmatrix} 0 & 1 \end{pmatrix}$ , on a :  $\begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} = 8 + x_2(k)$ , et la sortie  
 au plus tôt  $y^-(k)$  est  $8 + x_2^-(k)$ .

◇

#### Exemple 4.8

Le modèle initial et le modèle standard de l'illustration précédente (fabrication de draps) donnant la même trajectoire au plus tôt :

k	0	1	2	3
x	0	10	20	35
	0	20	30	45
	0	28	38	53
u	-	0	0	35
y	3	31	41	56

◇

#### 4.5.2 Technique utilisant la fonction objectif

Les résultats de la partie 4.5 montre qu'il est possible de déterminer la trajectoire au plus tôt d'un GET en utilisant une forme  $A \cdot x \leq b$  comme dans la programmation linéaire. Cependant, la fonction objectif n'est pas définie. Le résultat suivant fait le lien entre la fonction objectif de la programmation linéaire et l'ordre composante par composante, utilisé dans le précédente partie.

En fait, la solution optimale du problème de calcul de la trajectoire au plus tôt est aussi une solution d'un problème particulier de la programmation linéaire, comme indiqué par le résultat

suisant brièvement déclaré dans [Cottle 1972] [Gabbay 1976]. Cette affirmation est complétée par la preuve suivante.

**Théorème 4.9.** *Les propositions suivantes sont équivalentes :*

1. L'ensemble  $\Gamma = \{x \in R^n : Ax \leq b\}$  a un élément maximum  $x_0$  ;
2.  $x_0$  est optimal pour le problème  $\max\{cx, \text{ tel que } Ax \leq b \text{ pour tout } c > 0\}$  .

En outre, l'équivalence précédente entre 1 et 2 est vérifiée, si les mots maximum et max sont remplacés par minimum et min respectivement.

**Preuve :** - Soit  $x_0$  l'élément maximum de l'ensemble  $\Gamma$  (proposition 1). Alors  $\forall x \in \Gamma, x \leq x_0 \Rightarrow c.x \leq c.x_0$  comme  $c > 0$  alors,  $x_0$  est aussi la solution optimale pour la proposition 2.

- L'inverse est vérifiée avec une preuve par contradiction. Soit  $x_0$  la solution optimale du problème 2, supposons que la proposition 1 soit fausse. Comme  $x_0$  n'est pas l'élément maximum de l'ensemble  $\Gamma$ , il y a  $x \in \Gamma$  tel que  $\exists i$  avec  $x_i > (x_0)_i$ . Alors on a :  $x \not\leq x_0$ . Comme le système sup-monotone est un sup demi-treillis (Théorème 1.48), le maximum  $x \vee x_0$  appartient à  $\Gamma$ . Donc,  $x_0 \leq x \vee x_0$  et  $x_0 \neq x \vee x_0$ . Une conséquence est d'avoir  $c.x_0 < c.(x \vee x_0)$  comme  $c > 0$ , où  $x_0$  n'est pas la solution optimale du problème 2, ce qui conduit à une contradiction. Donc, l'inverse est prouvé. Notons que la condition  $c \geq 0$  n'est pas suffisante : si  $x_i$  est le seul élément tel que  $x_i > (x_0)_i$ ,  $c_i = 0$  conduit à  $c.x_0 = c.(x \vee x_0)$ .

□

Alors, les approches utilisant la programmation linéaire pour  $c > 0$  obéissent à une relation d'ordre si le système a un maximum ou un minimum. En conséquence, les algorithmes de la programmation linéaire peuvent aussi résoudre le problème à l'aide de la relation d'ordre et notamment le problème de la partie 4.5 qui peut être écrit comme suit.

**Corollaire 4.10.** *Le programme linéaire  $P_{fw}$ , pour calculer la trajectoire au plus tôt, est donné par les deux sous programmes linéaires suivant :*

- $P_{x^-} : x^-(k) = \min\{c.x(k), \text{ tel que } A^{fw}.x(k) \geq b^{fw}(x^-(k-1), u(k)) \text{ pour tout } c > 0\}$  ;
- $P_{y^-} : y^-(k) = \min\{c.y(k), \text{ tel que } A^{fw}.y(k) \geq b^{fw}(x^-(k)) \text{ pour tout } c > 0\}$ . ■

L'algorithme choisi pour la résolution du problème de calcul de la trajectoire au plus tôt  $P_{fw}$ , est l'algorithme du Simplexe (voir Annexe B.2), puisque ce dernier résout efficacement notre formulation. Pour la résolution, on utilise l'environnement **LPSolve IDE** (Integrated Development Interface). La complexité de notre résolution est en  $O(n^2)$  dans le pire des cas. L'ensemble des inégalités du programme linéaire  $P_{fw}$  ont une forme particulière : chaque inégalité a deux variables au plus. Certaines études [Hochbaum 1993] montrent que l'élimination de variables introduite par Fourier [Fourier 1827] est efficace pour ce type de formulation, et la complexité en temps est en  $O(mn^2 \log m)$ .

### La technique duale

L'ensemble des contraintes du programme linéaire  $P_{fw}$  augmente avec le nombre de places, aussi pour minimiser la complexité de l'algorithme de résolution, on a intérêt à minimiser le nombre de contraintes. Pour cela, il faut considérer le problème dual. La relation entre le dual et le primal est donnée par le théorème 1.34. Le problème dual  $D_{fw}$  est comme suit :

Le programme linéaire  $P_{x^-}$ , est nommé programme linéaire primal, on associe un programme linéaire dual  $D_{\alpha^-}$ , dont lequel on doit trouver un vecteur  $\alpha(k) \in \mathbb{R}^m$  ( $\dim(\alpha(k)) = \dim(b^{fw})$ )

$$D_{\alpha^-} : \alpha^-(k) = \{\max (\alpha(k) \times b^{fw}(x^-(k-1), u(k))) \quad : \quad \alpha(k)A^{fw^t} = c, \quad \alpha(k) \geq 0\}. \quad (4.16)$$

De la même manière, on obtient le problème dual  $D_{\beta^-}$  du primal  $P_{y^-}$  comme suit :

$$D_{\beta^-} : \beta^-(k) = \{\max (\beta(k) \times b^{fw'} x^-(k) \quad : \quad \beta(k)A^{fw'^t} = c, \quad \beta(k) \geq 0\}, \quad (4.17)$$

avec  $\dim(\beta(k)) = \dim(b^{fw'})$ .

## 4.6 Conclusion

Dans ce chapitre, nous avons décrit un modèle entrées/sorties sous la forme d'un système d'inégalités dans l'algèbre standard correspondant à un GET. Des matrices d'incidences particulières liées aux transitions d'entrées, internes et de sorties ont été introduites. On a en particulier présenté une méthode de standardisation du modèle telle que initialement chaque place interne (externe, respectivement) contienne un jeton (aucun jeton, respectivement). De ce fait, les itérations du modèle, en régime autonome, seront analogues à celles du modèle d'état de l'automatique classique. Elles dépendront exclusivement d'un sous-ensemble des inéquations liées aux transitions internes. Dans la partie 4.5, nous avons montré que la détermination de la trajectoire au plus tôt conduit à un système inf-monotone.

Pour la résolution de notre problème, on utilise un algorithme de programmation linéaire, qui dans notre cas est la méthode du Simplexe.



## Chapitre 5

# Commande des systèmes à événements discrets

### 5.1 Introduction

Dans le domaine des systèmes à événements discrets, un objectif souvent considéré est la commande de processus définie de la manière suivante : certains événements sont considérés comme commandables, ou contrôlables, c'est-à-dire, que le franchissement de la transition (d'entrée) peut être retardé jusqu'à un certain temps défini par le superviseur. De façon classique, la *commande* d'un système à événements discrets est le pilotage de ses entrées en vue d'obtenir un comportement spécifié pour le système. Depuis une quinzaine d'années, plusieurs approches complémentaires ont été proposées. Avant de présenter dans le chapitre suivant notre contribution sur le sujet, nous proposons dans la section suivante un bref état de l'art sur la commande des systèmes à événements discrets.

Dans la section suivante, nous donnons le modèle algébrique matriciel considéré dans cette étude et la notion d'inégalité monotone est appliquée pour la commande en juste-à-temps du GET. Finalement, nous donnons la commande effectuant une poursuite de trajectoire sur un horizon glissant.

### 5.2 Bref état de l'art de la commande des systèmes à événements discrets

Les stratégies de la commande pour les systèmes à événements discrets se résument en deux stratégies majeures :

1. Commande avec correcteur ;
2. Commande prédictive.

### 5.2.1 Commande avec correcteur

L'objectif de ce problème de commande est de synthétiser un correcteur modifiant le comportement d'un système nominal. En termes de GET, la correction va consister à ajouter des arcs au système nominal et par conséquent à en modifier la dynamique. Les performances attendues par le système corrigé sont spécifiées sous forme de modèle dit de référence.

#### 5.2.1.1 Commande avec spécifications

L'objectif de cette commande est de synthétiser un correcteur modifiant le comportement dynamique du système en vue d'obtenir pour le système corrigé (ensemble système nominal + correcteur) des performances spécifiées sous la forme d'un modèle dit de *référence*. Le correcteur  $C$  peut être placé en amont (précompensateur). La figure (5.1) présente une commande avec spécifications : correction du système  $H$  par un correcteur  $C$ .

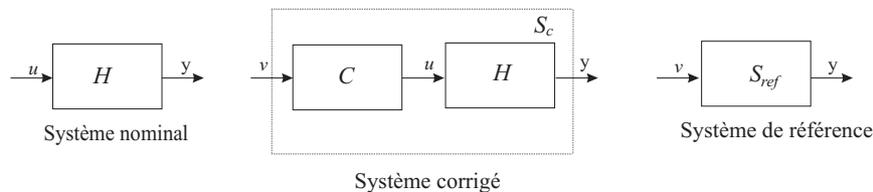


Fig. 5.1 : Commande avec spécifications.

#### 5.2.1.2 Commande avec retour d'information

Pour fournir une bonne robustesse vis-a-vis des incertitudes et des perturbations, on utilise la commande avec retour d'information. On considère toujours un système donné par son équation d'état et un ensemble de contrôles. On introduit alors une loi de commande qui permet d'exprimer la fonction de contrôle précisément en fonction de cet état. L'état du système est pris en compte à chaque instant pour déterminer "en temps réel" la commande. Le contrôle est de type feedback.

La commande avec retour d'information consiste à modifier la dynamique d'un système  $S$  par l'ajout d'un correcteur  $C$  (situé entre la sortie et l'entrée de  $S$ ). L'objectif de la commande est d'imposer au système bouclé la dynamique décrite par un modèle de référence  $G_{ref}$ . Dans la théorie des systèmes linéaires, ce problème de commande est similaire au problème classique de poursuite de modèle (*model matching problem* [Wang 1972]). Différentes structures de commande avec retour d'informations sont utilisées pour les systèmes à événements discret (max,+) linéaire (retour de sortie [Cottenceau 1999] et retour d'état [Cottenceau 2001] [Lhommeau 2003]).

Dans [Lahaye 2004], les auteurs ont proposé un correcteur pour que le système corrigé atteigne le modèle de référence.

Une autre étude sur la commande sous contrainte de temps a été développée dans [Amari 2005]. Cette synthèse de commande permet de valider un ensemble de contraintes temporelles dans

un GET. La loi de commande est finalement définie par une équation aux différences  $(\min,+)$  linéaires ou  $(\max,+)$  linéaires, qui présente un nombre fini de retards. Cette équation correspond elle-même à un graphe d'événements temporisé.

### 5.2.2 Commande prédictive

La commande prédictive, avec une connaissance du modèle, envisage le comportement du procédé dans le futur en fonction des objectifs à atteindre. La stratégie de mise en oeuvre de la commande prédictive s'articule autour d'un critère traduisant l'objectif de fonctionnement souhaité du GET. Dans cette section, nous présentons les différents objectifs de fonctionnement et leurs critères.

#### 5.2.2.1 La poursuite de trajectoire désirée

La définition formelle du concept de poursuite peut être établie en conformité avec la définition donnée par [Fossen 2003].

**Définition 5.1.** *Quand l'objectif est de forcer la sortie du système  $y(t) \in \mathbb{R}^m$  pour le suivi d'une sortie désirée  $y_d(t) \in \mathbb{R}^m$ , il sera défini comme un problème de suivi.*

La sortie désirée représente la trajectoire de référence, également appelée consigne. Le problème de commander la sortie d'un système par rapport à une référence donnée, peut être traité de différentes manières. En automatique classique, une des approches est celle du problème de la stabilisation en un point, ou le problème de stationnement [Frezza 1999], qui consiste à faire diriger un système vers un point de configuration fixe, peu importe le comportement du système entre l'état initial et l'état final.

Dans l'algèbre des dioïdes, la poursuite de trajectoire consiste à établir une trajectoire d'entrée (commande) du système telle que la réponse en sortie **suive au mieux** une trajectoire de consigne correspondant au comportement désiré en sortie du système, ce qui est analogue à une inversion du modèle. Dans ce domaine, l'approche proposée dans [Cohen 1989] permet d'établir la plus grande trajectoire d'entrée, notée  $u$ , telle que la réponse, notée  $y$ , soit inférieure ou égale à une trajectoire de consigne, notée  $z$ . Plus précisément, pour un système  $\Gamma$ , on cherche le plus grand élément de l'ensemble suivant :

$$\{u \in \mathcal{D}^{\mathbb{Z}} \mid y = \Gamma(u) \preceq z\},$$

où  $\preceq$  est l'ordre défini dans le dioïde  $\mathcal{D}^{\mathbb{Z}}$ .

Les premiers travaux concernant le pilotage des systèmes à événements discrets sont basés sur le critère de *juste-à-temps*. L'objectif de la commande revient à calculer une trajectoire d'entrée de sorte que celle de la sortie soit plus petite ou égale à une trajectoire de référence. Autrement

dit, dans le cas du critère de *juste-à-temps*, le problème de poursuite de trajectoire consiste à rechercher la plus grande commande  $u$  telle que  $y \leq z$  où  $z$  est la consigne et  $y$  la sortie du système. Cette commande est notée  $u_{opt}$ , et se définit par :

$$u_{opt} = \bigoplus_{\Gamma(u) \leq z} u.$$

Le critère de *juste-à-temps* vise à effectuer le nombre minimum de tirs des transitions d'entrée, et à retarder au maximum le tir de ces transitions, tout en satisfaisant les objectifs de commande.

Différents travaux ont permis d'étendre la commande en juste-à-temps des systèmes linéaires sans spécifications dans les cas souvent :

- le modèle ne reflète pas le comportement réel du système (l'existence d'une désadaptation entre le système et son modèle) [Boimond 1996] , [Menguy 000a] et ([Menguy 1997], §6) ;
- des entrées sont non maîtrisables [Menguy 000b] et ([Menguy 1997], §5.4.2) ;
- le système est linéaire, mais non stationnaire [Lahaye 2000]

Une autre structure de commande en juste-à-temps des systèmes linéaires avec spécifications :

- un modèle a été proposée dans [Declerck 2010]. Elle est basée sur l'utilisation simultanée de la poursuite de trajectoire et celle de modèle. Les spécifications se présentent sous la forme d'intervalle décrit par des fonctions (min,max,+). Ce modèle utilise la technique du point fixe ;
- une commande en juste-à-temps de systèmes (*max, +*)-linéaire sous contraintes [Houssin 2007]

Le principe de cette commande est que le processus vérifie des conditions initiales et finales données, sachant que les variables d'état sont soumises à certaines contraintes.

Parallèlement à ces travaux, un critère différent a été considéré notamment dans [De Schutter 2000]. Le critère de la commande prédictive se décompose en une somme pondérée de deux sous-critères qui correspondent, d'une part, à l'erreur quadratique entre une trajectoire de référence connue et la sortie prédite du système, et d'autre part, à l'amplitude des variations de la commande (en vue de prendre en compte la sollicitation des actionneurs). Ce dernier sous-critère a donc été substitué par une fonction coût qui correspond à la somme des dates de tirs de toutes les entrées, soit :

$$J_{in} = \sum_{k=k_s+1}^{k_f} \sum_{i=1}^{\dim(u)} u_i(k),$$

où  $[k_s, k_f]$  représente l'horizon de prédiction.

Le but de ce sous critère est de maximiser la somme des dates d'activations des événements d'entrées (en ce sens, ce sous critère se rapproche de celui du juste-à-temps exposé précédemment). Un critère global considéré par [De Schutter 2000] à minimiser est donné par :

$$J = J_{out} - \lambda \cdot J_{in},$$

où  $J_{out}$  représente le critère de poursuite de trajectoire et  $\lambda$  est une variable scalaire utilisée pour pondérer le critère de commande

$$J_{out} = J + \lambda \cdot J_{in}.$$

### 5.2.2.2 Commande avec retour d'information

Un second objectif est l'extension de ce premier problème, en effectuant une poursuite de trajectoire sur un horizon glissant. L'approche est une forme de commande prédictive dans laquelle la commande courante est obtenue par la résolution en ligne, d'un problème de commande optimale en boucle ouverte, ceci à chaque instant "d'échantillonnage". La procédure peut être répétée indéfiniment, en utilisant un glissement de l'horizon fini considéré, c'est-à-dire, la fenêtre de temps considérée pour les calculs est décalée à chaque itération. Une approche a été proposée par [Declerck 2009], elle est basée sur le concept de la sortie au plus tôt désirée. La mise à jour associée à la sortie désirée est effectuée par une technique *forward*, tandis que la commande est donnée par une approche *backward*.

## 5.3 Commande au plus tard d'un GET

Dans la suite, on s'intéresse uniquement au problème de commande par poursuite de trajectoire. Nous montrons que la synthèse de cette commande se généralise pour des systèmes linéaires admettant une écriture polyédrale de la forme  $A \cdot x \leq b$  dans l'algèbre conventionnelle.

### 5.3.1 Propriétés des GET

**Définition 5.2.** *Un graphe d'événements est observable si toute composante connexe est reliée à une sortie par au moins un chemin dans le réseau. Il est contrôlable si toute composante connexe est reliée à au moins une entrée par un chemin dans le réseau.*

**Définition 5.3.** [Baccelli 1992] *Un graphe d'événements est structurellement observable si, pour toutes les transitions internes, il existe un chemin d'accès à une transition de sortie au moins. Le nombre minimal des jetons de ces chemins dans le marquage initial  $x_i$  est noté  $\pi_{i,out}$ .*

Dans les réseaux de Petri, une autre définition est souvent utilisée : une transition est dite observable si les dates des tirs sont connues.

Un effet possible sur les dates de transitions internes ne peut être fait seulement s'il y a un chemin entre une transition d'entrée et chaque transition interne.

**Définition 5.4.** [Baccelli 1992] *Un graphe d'événements est structurellement contrôlable si, chaque transition interne peut être atteinte par un chemin d'une transition d'entrée au moins.*

Cette partie est consacrée à la détermination d'une commande en boucle ouverte pour les SED. Cette commande est optimale vis-à-vis du critère de juste-à-temps : la commande doit retarder autant possible le système, de sorte que la sortie se produise avant une trajectoire de référence  $z$  définie sur un horizon  $[k_s, k_f]$ .

**Objectif :**

L'objectif du problème que nous abordons ici est formulé de la manière suivante. Étant donnée une trajectoire de sortie désirée  $z(k)$ , on cherche à calculer la plus grande commande  $u(k)$  telle que la réponse du graphe à cette entrée produise une sortie  $y(k)$  vérifiant la contrainte suivante :

$$y(k) \leq z(k). \quad (5.1)$$

En raison de la propriété d'isotonie matricielle, obtenir le plus grand  $u(k)$  entraîne le calcul du plus grand état et de la plus grande sortie *via*, respectivement, l'équation d'état et l'équation relative à la sortie.

Le problème est la détermination de la plus grande commande  $u(k)$  de telle façon que  $y(k) \leq z(k)$  sur l'horizon  $[k_s + 1, k_f]$ .

Pour le problème de commande traité ici, la séquence de production désirée  $z(k)$  est supposée connue, tandis que la commande  $u(k)$  et l'état  $x(k)$  sont les variables à déterminer.

Respectant l'ordre composant par composant (voir la section 1.4.4), la plus grande commande est le maximum des solutions possibles. L'état initial est  $x(k_s) = x_{k_s}$  et l'hypothèse  $k_s = 0$  peut éventuellement être prise. Une solution connue de ce problème existe dans l'algèbre  $(max, +)$  et est décrite dans la section 5.6 de [Baccelli 1992].

De manière similaire à la section 4.5, le système composé de (4.6) et (4.8) est réécrit sous la forme  $Ax \leq b$ . A partir de l'inégalité (4.6), on obtient le système suivant. Pour  $k \geq k_s + 1$ ,

$$\begin{pmatrix} I \\ W_{x \rightarrow x}^+ \end{pmatrix} x(k-1) \leq \begin{pmatrix} 0 \\ -T_{x \rightarrow x} \end{pmatrix} + \begin{pmatrix} I \\ W_{x \rightarrow x}^- \end{pmatrix} x(k),$$

l'inégalité de sortie (4.8) et la contrainte  $y(k) \leq z(k)$  pour  $k \geq k_s + 1$  impliquent

$$W_{x \rightarrow y}^+ \cdot x(k) \leq -T_{x \rightarrow y} + W_{x \rightarrow y}^- \cdot y(k) \leq -T_{x \rightarrow y} + W_{x \rightarrow y}^- \cdot z(k).$$

Finalement, le système d'inégalités est exprimé sous la forme :

$$A^{bw} \cdot x(k) \leq b^{bw}(z(k), x(k+1)), \quad (5.2)$$

(abréviation *bw* pour *BackWard*) avec :

$$A^{bw} = \begin{pmatrix} W_{x \rightarrow y}^+ \\ I \\ W_{x \rightarrow x}^+ \end{pmatrix}$$

et

$$b^{bw}(z(k), x(k+1)) = \begin{pmatrix} -T_{x \rightarrow y} \\ 0 \\ -T_{x \rightarrow x} \end{pmatrix} + \begin{pmatrix} W_{x \rightarrow y}^- & 0 \\ 0 & I \\ 0 & W_{x \rightarrow x}^- \end{pmatrix} \begin{pmatrix} z(k) \\ x(k+1) \end{pmatrix},$$

pour  $k \geq k_s + 1$ .

Le problème considère un horizon fini  $[k_s + 1, k_f]$  et le système (5.3.1) doit être limité à  $k \leq k_f$ . Ce qui signifie qu'il n'y a pas d'information sur la production après l'événement  $k_f$ . La sortie désirée  $z(k)$  et l'état estimé  $x(k)$  doivent être réglés à la valeur maximale possible, c'est-à-dire,  $+\infty$  pour  $k > k_f$ .

En plus, l'inégalité de sortie (4.7) doit également être prise en considération. Le système est :

$$A^{bw'} \cdot u(k) \leq b^{bw'}(x(k)), \quad (5.3)$$

avec

$$A^{bw'} = W_{u \rightarrow x}^+ \text{ et } b^{bw'}(x(k)) = -T_{u \rightarrow x} + W_{u \rightarrow x}^- \cdot x(k), \text{ pour } k \geq k_s + 1. \quad (5.4)$$

### 5.3.2 La technique algébrique

L'hypothèse de l'observabilité structurelle est utilisée dans le théorème suivant. Dans l'algèbre  $(\max, +)$ , l'observabilité structurelle ([Baccelli 1992] §partie 6.6.2) donne une condition pour observer *potentiellement* l'effet de la date de tir d'une transition interne sur au moins une sortie. Dans cette définition, les dates de franchissement de sortie ne sont pas particulièrement connues.

Nous supposons que :

1. chaque sortie est observable,
2. le graphe d'événements est structurellement observable et structurellement contrôlable.

Dans un graphe d'événements structurellement observable, chaque transition interne a une influence structurelle sur au moins une transition de sortie. Cette influence structurelle peut être vue si chaque sortie est observable. Par conséquent, à une date arbitrairement grande de franchissement de cette transition interne correspond une date arbitrairement grande de franchissement d'une transition de sortie.

Le théorème suivant utilise des matrices  $A^{bw}$ ,  $A^{bw'}$  et les vecteurs  $b^{bw}$ ,  $b^{bw'}$  définis ci-dessus. La notation  $\pi_i^{out}$  est étendue à la transition d'entrée.

**Théorème 5.5.** - *Pour chaque transition interne  $i$ , il y a un état maximum, noté  $x_i^+(k)$ , satisfaisant (5.3.1) sur l'horizon  $[k_s, k_f - \pi_i^{out}]$ . Pour chaque transition d'entrée  $i$ , il existe une commande maximale, notée  $u_i^+(k)$ , satisfaisant (5.4) sur l'horizon  $[k_s, k_f - \pi_i^{out}]$ .*

- La trajectoire d'état au plus tard est définie par  $x_i^+(k) = \bigwedge_{j|A_{j,i}^{bw}=1} b_j^{bw}(z(k), x^+(k+1))$ , pour  $k \in [k_s, k_f]$ .

- La trajectoire d'entrée au plus tard est définie par  $u_i^+(k) = \bigwedge_{j|A_{j,i}^{bw'}=1} b_j^{bw'}(x^+(k))$ , pour  $k \in [k_s + 1, k_f]$ .

**Preuve :**

- Le système linéaire d'inégalités  $A^{bw} \cdot x(k) \leq b^{bw}(z(k), x(k+1))$  est sup-monotone puisque chaque ligne de la matrice  $A^{bw}$  a un coefficient strictement positif.
- La résolution du système suivant donne un majorant :  $W_{x \rightarrow y}^+ x(k_f) \leq -T_{x \rightarrow y} + W_{x \rightarrow y}^- z(k_f)$  et  $x(k) \leq x(k+1)$  pour  $k \leq k_f$ .
- D'une manière générale, certaines colonnes de  $W_{x \rightarrow y}^+$  peuvent être nulles si la transition interne en question n'est pas directement liée à une transition de sortie : seules certaines composantes de  $x(k_f)$  ont un majorant fini si  $z(k_f)$  est fini. Si l'on suppose que le système est structurellement observable, alors il existe pour chaque transition interne  $x_i$  au moins un chemin vers une transition de sortie  $y_j$  dans le graphe d'événements. La résolution suit la direction opposée : il y a une relation reliant les valeurs d'une transition de sortie  $y_j$  et une transition interne  $x_i$ , mais avec un changement dans la numérotation des événements. Pour chaque transition interne  $i$ , une valeur  $\pi_i^{out}$  minimale existe puisque le graphe d'événements est structurellement observable.

La valeur finie  $z(k)$  génère un majorant de  $x_i(k - \pi_i^{out})$ , les valeurs correspondantes aux transitions internes peuvent être considérées dans l'approche.

- L'ensemble réduit des variables avec majorant est non-vide et la solution peut être calculée facilement. En effet, puisque chaque ligne  $j$  de la matrice  $A^{bw}$  contient seulement un coefficient strictement positif  $A_{i,j}^{bw} = 1, A_{i,j}^{bw} x_i(k) \leq b_j^{bw}(z(k), x(k+1))$  pour tout  $A_{i,j}^{bw} \neq 0$ . Enfin, le théorème (1.49) montre que l'ensemble des solutions a un élément maximal  $x^+$ . En outre, le système  $A^{bw} u(k) \leq b^{bw'}(x(k))$  est également sup-monotone. Le même raisonnement peut être appliqué pour les entrées : comme chaque transition d'entrée est connectée au système, un majorant des majorants de l'état peut être déduit : le passage de la numérotation  $\pi_i^{out}$  peut être étendue à une transition d'entrée. Avec cette restriction, la plus grande trajectoire existe dans  $\mathbb{R}$  et peut être déduite à partir de l'inégalité suivante : comme chaque ligne  $j$  de la matrice  $A^{bw'}$  ne contient qu'un seul coefficient strictement positif  $A_{i,j}^{bw'} = 1, A_{i,j}^{bw'} u_i(k) \leq b_j^{bw'}(x(k))$ , pour chaque  $A_{j,i}^{bw'} \neq 0$ .

□

A partir de  $k_f$ , la détermination suit une itération en arrière de  $k_f$  à  $k_s$ . Rappelons que  $x^+(k_f + 1)$  est égal à l'infini (ou une valeur arbitrairement grande) lors de la première étape.

Si le graphe d'événements est structurellement contrôlable, un effet possible d'une transition d'entrée sur une transition interne  $x_i$  est fait avec un changement de la numérotation des événements. Sa valeur minimale est noté  $\pi_i^{in}$ . Pour résumer, l'hypothèse de l'observabilité d'un graphe d'événements donne un couple  $(\pi_i^{in}, \pi_i^{out})$  pour chaque transition interne  $x_i$ .

**Exemple 5.6** (suite de l'exemple 4.7)

On a  $(\pi_1^{in}, \pi_1^{out}) = (0, 1)$  et  $(\pi_2^{in}, \pi_2^{out}) = (1, 0)$ . L'application du théorème 5.5 donne les relations d'état suivantes :

$$\text{Comme } \begin{pmatrix} -T_{x \rightarrow y} \\ 0 \\ -T_{x \rightarrow x} \end{pmatrix} = \begin{pmatrix} -8 \\ 0 \\ 0 \\ -4 \\ -5 \end{pmatrix} \text{ et } \begin{pmatrix} W_{x \rightarrow y}^- & 0 \\ 0 & I \\ 0 & W_{x \rightarrow x}^- \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

$$b^{bw}(z(k), x(k+1)) = \begin{pmatrix} -8 + z(k) \\ x_1(k+1) \\ x_2(k+1) \\ -4 + x_2(k+1) \\ -5 + x_1(k+1) \end{pmatrix} \text{ et } A^{bw} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \text{ l'état le plus grand}$$

$$\begin{pmatrix} x_1^+(k) \\ x_2^+(k) \end{pmatrix} \text{ est :}$$

$$\begin{pmatrix} x_1^+(k+1) \wedge (-4 + x_2^+(k+1)) \\ (-8 + z(k)) \wedge x_2^+(k+1) \wedge (-5 + x_1^+(k+1)) \end{pmatrix}. \quad \diamond$$

### 5.3.3 Technique utilisant la fonction objectif

Avec la forme  $Ax \leq b$ , la détermination de la trajectoire au plus tard peut aussi utiliser l'approche de la programmation linéaire. De la même façon que le calcul de la trajectoire au plus tôt, le corollaire suivant est le résultat du théorème (4.9) :

**Corollaire 5.7.** *Le programme linéaire  $P_{bw}$  (abréviation bw pour Backward) pour calculer la trajectoire au plus tard est donné par les deux sous programmes linéaires suivant :*

- $P_{x^+} : x^+(k) = \max\{c \cdot x(k), \text{ tel que } A^{bw} \cdot x(k) \leq b^{bw}(z(k), x^+(k+1)) \text{ pour tout } c > 0\}$ ;
- $P_{u^+} : u^+(k) = \max\{c \cdot u(k), \text{ tel que } A^{bw'} \cdot u(k) \leq b^{bw'}(x^+(k)) \text{ pour } c > 0\}$ .

■

On applique l'environnement **LPSolve IDE**, qui consiste à parcourir le polyèdre des points réalisables de sommet en sommet jusqu'à ce qu'on ne puisse plus améliorer la solution. Au départ, la fonction objectif est nulle, et il s'agit de l'augmenter. Si certains de ses coefficients sont positifs, il apparaît clairement qu'en augmentant l'une des variables correspondant à un coefficients positifs, on augmente cette fonction objectif. On a donc un critère d'obtention de l'optimum. La complexité de l'algorithme est en  $O(n^2)$  dans le pire des cas.

#### Technique duale

On utilise le Simplexe comme algorithme de résolution comme le cas du calcul de la trajectoire

au plus tôt, et pour minimiser la complexité on utilise le programme dual du problème initial 1.34. L'écriture du problème dual  $D_{bw}$  est comme suit :

Le programme dual  $D_{\alpha^+}$  du programme primal  $P_{x^+}$  est donné par :

$$D_{\alpha^+} : \alpha^+(k) = \{\min \alpha(k) \times b^{bw}(z(k), x^+(k+1)) \quad : \quad A^{bw^t} \alpha(k) \geq c, \quad x^*(k) \geq 0\}. \quad (5.5)$$

Le programme dual  $D_{\beta^+}$  du programme primal  $P_{u^+}$  est donné par :

$$D_{\beta^+} = \beta^+(k) = \{\min \beta(k) \times b^{bw^t}(x^+(k)) \quad : \quad A^{bw^t} \beta(k) \geq c, \quad \beta(k) \geq 0\}. \quad (5.6)$$

## 5.4 Commande à horizon glissant

Parmi les stratégies de commande existantes, l'approche prédictive est souvent employée dans le milieu industriel. Cette stratégie, apparue il y a une quarantaine d'années [Propoi 1963] et [Rafal 1968], consiste à optimiser, à partir des entrées/sorties d'un système, le comportement futur prédit du système considéré. La prédiction est faite à partir d'un modèle interne du système sur un intervalle de temps fini, appelé horizon de prédiction. La solution du problème d'optimisation est un vecteur de commande dont la première entrée de la séquence optimale est injectée au système. Le problème est à nouveau résolu sur l'intervalle de temps suivant en utilisant les données du système mises à jour.

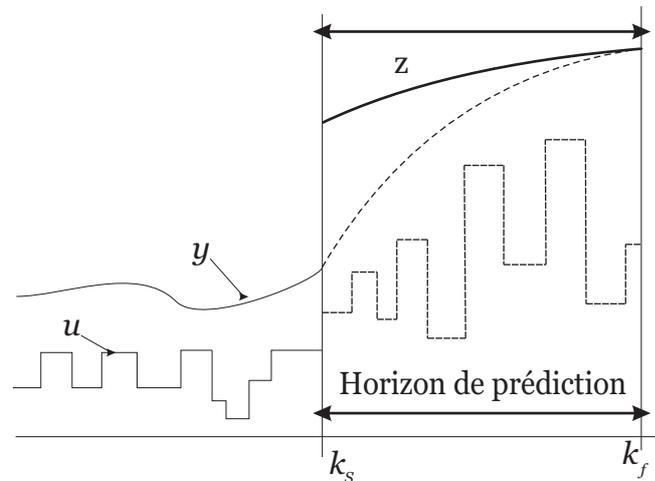
La première génération de commande prédictive appliquée en milieu industriel a été initiée par Richalet [Richalet J. 1978], sous le nom de MPHPC (*Model Predictive Heuristic Control*), qui fut connue plus tard sous le nom de commande algorithmique (*MAC - Model Algorithmic Control*) [Verbruggen 1984]. Dans ces approches, l'objectif est de poursuivre une référence, mais les contraintes ne sont pas prises en compte. Ces algorithmes ont permis de définir l'essence même des stratégies prédictives, sachant que les applications nombreuses dans le milieu industriel ont assurées leur pérennité.

La deuxième génération apparues au début des années 1980, permet en plus la prise en compte de contraintes sur les entrées et les sorties en posant un problème d'optimisation quadratique. Le lecteur trouvera dans [Allgöwer 1999] un état de l'art complet sur les stratégies prédictives.

L'objectif de cette partie est d'effectuer une poursuite de trajectoire sur un horizon glissant. L'approche est une forme de commande prédictive dans laquelle la commande courante est obtenue par la résolution en ligne, d'un problème de commande optimale en boucle ouverte, ceci à chaque instant "d'échantillonnage". La procédure peut être répétée à l'infini, en utilisant un glissement de l'horizon fini considéré. Dans les paragraphes suivants, le principe de cette stratégie est développé.

### 5.4.1 Principe

La commande prédictive inclue une famille de méthodes de commande qui utilise explicitement un modèle du processus étudié afin d'obtenir le signal de commande en minimisant une



**Fig. 5.2** : Modèle prédictif pour les systèmes continues ( $u$  est numériser).

fonction objectif (critère). Toute commande prédictive est basée sur les trois éléments suivants. L'utilisation d'un modèle afin de prédire (calculer) la sortie du processus lors des instants futurs ; le calcul d'un signal de commande qui minimise la fonction objectif ; l'utilisation du principe de glissement de façon à ce que, après chaque étape de calcul, l'horizon soit décalé dans le futur.

La méthodologie des commandes appartenant à la famille des modèles prédictives est caractérisée par les trois principes suivants.

### 5.4.2 Modèle de prédiction

La sortie future pour un horizon déterminé  $h$ , appelé horizon de prédiction, est prédite à chaque instant  $t$  en utilisant le modèle du processus, ou du système étudié, et comparée à la sortie réelle obtenue pour ce système. Ces sorties prédites, notées  $y(t+k)$  pour  $k = 1 \dots N$ , dépendent des valeurs connues jusqu'à l'instant  $t$  (entrées et sorties passées) et des valeurs du signal de commande future, notées  $u(t+k)$ ,  $k = 0 \dots N-1$ . Ces dernières vont être appliquées au système et doivent alors être calculées.

### 5.4.3 Fonction objectif

L'ensemble des valeurs futures du signal de commande (entrées) est calculé en optimisant une fonction objectif (ou un critère coût) déterminée dans le but de garder le processus aussi proche que possible de la trajectoire de référence. Les différents algorithmes des modèles prédictifs proposent différentes fonctions de coût pour obtenir la loi de commande. Le but général est que la sortie future  $y$  dans l'horizon considéré, suive une trajectoire de référence  $z$  tout en minimisant un critère. Ce critère prend, en général, la forme d'une fonction d'erreur quadratique entre la sortie prédite et la sortie de référence en addition à l'effort de la commande.

## 5.5 Commande d'un GET sur un horizon glissant

Le contrôle optimal de la partie 5.3.1 peut maintenant être étendu à une approche en ligne, et en particulier à une commande effectuant une poursuite de trajectoire sur un horizon glissant. La procédure est la suivante. Après un contrôle optimal sur l'horizon  $[k_s + 1, k_f]$  à l'étape  $k_s$ , l'horizon est déplacé et devient  $[k_s + 2, k_f + 1]$  : le problème de la commande au plus tard est mis à jour avec une nouvelle information sur les mesures et une nouvelle optimisation à l'étape  $k_s + 1$  doit être effectuée. Ci-dessous, la technique générale de l'horizon glissant du modèle de la commande prédictive est détaillée.

### 5.5.1 Horizon glissant

Nous supposons l'hypothèse suivante : pour un numéro d'événements  $k$ , la date de chaque événement est disponible et ainsi, à l'étape  $k = k_s$ ,  $u_{k_s}$  et  $x_{k_s}$  sont connus. Une future commande  $u(k)$ , pour  $k \in [k_s + 1, k_f]$ , est déterminée de telle sorte que la commande est la solution optimale du problème 2. Le premier élément de la séquence optimale (ici  $u(k_s + 1)$ ) est alors appliqué au processus. Lors de la prochaine étape  $k_s + 1$ , l'horizon est déplacé : le problème est mis à jour avec une nouvelle information  $x_{k_s+1}$  et une nouvelle étape d'optimisation à  $k_s + 1$  est effectuée.

### 5.5.2 Deux situations importantes

Dans la partie (5.3.1), la commande de synthèse génère la plus grande trajectoire de l'état, notée  $x^+$ , et la commande notée,  $u^+$  à l'étape  $k_s$ . La plus grande sortie sortie  $y^+$  doit satisfaire  $y^+ \leq z$ . Après ce calcul, où on suppose que la condition est vérifiée, les trajectoires d'état au plus tôt  $x^-$  et de la sortie associée  $y^-$  peuvent être calculées connaissant  $x_{k_s}$  et la commande  $u$ . Cette technique est décrite dans la partie 4.5. La différence  $x^+(k) - x^-(k)$  exprime la marge qui est non-négative dans la partie 5.3.1. Deux cas importants se présentent alors :

- Si la condition  $x^- \leq x^+$  est satisfaite, la monotonie implique  $y^- \leq y^+$  et  $y^-$  satisfait alors  $y \leq z$ . Par conséquent, la commande de la partie 5.3.1 a une solution pour les données  $z$  et  $x_{k_s}$  : il existe une commande optimale de telle sorte que, à partir de l'état actuel  $x_{k_s}$ , le GET peut suivre le critère de juste-à-temps, défini par  $y \leq z$ . L'exemple (4.4.4) dans la partie 5.3.1 illustre cette situation.

- Si la condition  $x^- \leq x^+$  est non satisfaite, le GET ne peut pas (provisoirement) obéir au critère de juste-à-temps  $y \leq z$  et l'état  $x^+$  est le plus grand trajectoire possible correspondant à cette critère. Cette situation se produit lorsque le processus présente certains retards produits par une perturbation du processus d'activité par exemple. Une conséquence est une augmentation de la trajectoire de l'évolution passée qui produit, de manière générale, un retard global du trajectoire d'état après  $k_s$ . Le vecteur  $x_{k_s}$  exprime l'influence de cette évolution passée.

En réponse à ce problème, nous donnons ci-dessous une approche utilisant la relation d'ordre composante par composante.

### 5.5.3 Approche utilisant relation d'ordre composante par composante

Une approche peut être la synthèse des parties 4.5 et 5.3.1 à une commande effectuant une poursuite de trajectoire sur un horizon glissant. L'objectif est alors la considération des deux critères suivants dans le système de commande.

- Critère de juste-à-temps sur les entrées à **maximiser**.

Rappelons que la considération de ce critère permet de limiter autant qu'il est possible les niveaux des pièces dans le processus considéré si on considère un système de production. Le critère de juste-à-temps ayant déjà été présenté, nous considérons ci-dessous le critère sur les retards des sorties.

- Retards des sorties à **minimiser**.

Si une pénalité pour chaque retard d'une sortie par rapport à la sortie désirée  $z$  correspondante doit être payée, un critère est le retard de la sortie qui est égal à  $y_i - z_i$  si  $y_i \geq z_i$  ou 0 si  $y_i < z_i$ . Si le retard noté  $\delta_i$  dans l'inégalité  $y_i - z_i \leq \delta_i$  est minimisé, la sortie  $y_i$  est aussi minimisée. Le cas  $y_i < z_i$  est considéré si la contrainte  $\delta_i \geq 0$  est ajoutée et la minimisation de la sortie  $y_i$  est limitée. Ainsi, une résolution peut minimiser le retard du vecteur de sortie composante par composante, ceci sur l'horizon  $[k_s + 1, k_f]$ .

Une approche en deux étapes qui connecte les deux critères précédents est proposée ci-dessous. *A priori*, une approche peut être l'application d'une technique déjà appliquée dans la partie 5.3.1 qui est basée sur la maximisation. Le système approprié est alors une inégalité  $Ax \leq b$  qui est sup-monotone. La minimisation de  $\delta(k)$  est équivalente à la maximisation de  $-\delta(k)$ . Conformément à la maximisation de la trajectoire  $y(k)$ , la forme qui convient est la suivante :

$$\begin{pmatrix} I \\ 0 \end{pmatrix} y(k) + \begin{pmatrix} I \\ I \end{pmatrix} \cdot \alpha(k) \leq \begin{pmatrix} z(k) \\ 0 \end{pmatrix} \text{ avec } \alpha(k) = -\delta(k).$$

Cependant, le système qui définit le retard n'est pas sup-monotone car la matrice correspondant à  $A$  de  $Ax \leq b$ , présente des lignes avec deux coefficients positifs : la technique basée uniquement sur la relation d'ordre composante par composante ne peut pas être directement appliquée. La technique consiste alors à séparer la minimisation et la maximisation.

Par conséquent, le problème doit être modifié de telle sorte que la condition  $x^- \leq x^+$  soit satisfaite. Dans ce document, nous supposons que le modèle du GET ne peut pas être modifié et que la condition initiale  $x_{k_s}$  est le résultat d'une évolution passée. Par conséquent, la seule possibilité est de modifier le critère de juste-à-temps et de retarder la sortie désirée de telle sorte que le problème 2 présente une solution. La technique est ainsi de retarder la sortie désirée avec une sortie désirée qui peut être suivie par le graphe d'événements temporisé. L'objectif de la démarche suivante est la détermination de cette nouvelle sortie souhaitée telle que la commande du problème 2 présente une solution. Les deux étapes a) et b) de cette approche sont les suivantes :

**a- La prévision de la sortie au plutôt désirée  $z^-$**

Comme une sortie désirée minimum permet la limitation des retards, le problème est de trouver la sortie désirée au plus tôt  $z^-$  telle qu'il existe une commande dont l'application au GET génère une trajectoire d'état à partir de l'état courant  $x_{k_s}$ . Cette sortie désirée au plus tôt est une limite telle que le GET ne peut pas suivre une trajectoire inférieure satisfaisant les contraintes du problème. La trajectoire au plus tôt peut être calculée par l'application de l'approche décrite dans la partie 4.5. Les données connues sont la condition initiale  $x_{k_s}$  et la commande minimale  $u(k) = u_{k_s}$  pour  $k \geq k_s$ . Rappelons que cet problème est borné par dessous, puisque  $x_{k_s}$  est un minorant de  $x(k)$  pour  $k \geq k_s$ .

### b- La modification de la trajectoire de sortie désirée

L'étape (b) est le remplacement de la sortie désirée  $z$  par une nouvelle sortie désirée qui est le maximum de  $z^-$  et  $z$  :  $z_m(k) = z^-(k) \vee z(k)$  pour  $k \in [k_s + 1, k_f]$ . Le retard est minimisé puisque la trajectoire  $z^-$  est minimale. Enfin, la technique de commande décrite dans la partie 5.3.1 est appliquée connaissant la trajectoire de sortie désirée modifiée  $z_m$ . Le critère de juste-à-temps est satisfait et la commande calculée est optimale pour  $z_m$ .

#### Exemple 5.8 suite de l'exemple 4.7

Considérons le problème suivant qui ne peut être résolu que par une approche spécifique : le critère de juste-à-temps ( $y(k) \leq z(k)$ ) ne peut être satisfait pour l'état courant et la sortie désirée considérés. L'horizon est  $[k_s + 1, k_s + 10]$  et  $k_f = k_s + 10$ . On considère 10 pas de l'algorithme. La sortie désirée est  $z(k) = 42 + 24 \cdot \lfloor \frac{k-1}{3} \rfloor$ , pour  $k \geq 1$ , où  $\lfloor x \rfloor$  est le plus grand entier plus petit ou égal à  $x$ . Soit  $x(0) = \begin{pmatrix} 50 \\ 50 \end{pmatrix}$ .

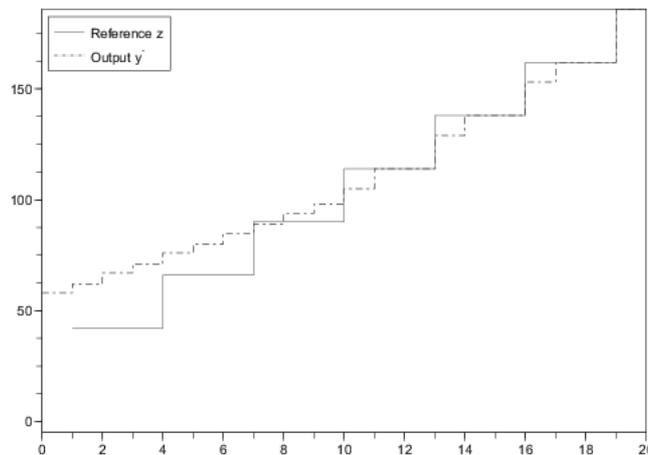


Fig. 5.3 : Commande sur un horizon glissant.

Du modèle, on déduit  $y(1) \geq 62 > z(1) = 42$  pour tous les contrôles et  $y(k) \not\leq z(k)$  pour

$k \geq 1$ . Dans ce cas, la Figure 5.3 montre l'efficacité de la technique proposée : la sortie  $y$  converge vers la sortie désirée  $z$  à  $k = 11$ .  $\diamond$

## 5.6 Conclusion

Dans ce chapitre, nous donnons les inégalités qui décrivent le comportement au plus tard. Cette trajectoire peut aussi être obtenue avec la programmation linéaire comme il est montré dans la section (5.3.3). La trajectoire au plus tard (section 5.3.1) permet d'obtenir une commande de GET sur un horizon glissant (section 5.5).

Dans un contexte plus formel, cette étude se base sur le concept de la relation d'ordre composante par composante, ainsi que sur la structure du polyèdre, inf-monotone et/ou sup-monotone. L'existence d'un système sup-monotone conduit à la détermination de la trajectoire au plus tard (section 5.3.1). L'application du théorème (1.49) montre que la solution maximale existe.



# Conclusion

Les travaux présentés dans ce mémoire portent sur l'étude des graphes d'événements temporisés et temporels. Ces travaux ont pour objectif d'obtenir certains résultats établis sur la modélisation, l'analyse de performances et la commande des graphes d'événements temporisés et temporels, dans l'algèbre conventionnelle où nous avons exploité des résultats importants de la programmation linéaire. Classiquement ces travaux sont étudiés dans le cadre de l'algèbre des dioïdes.

Ce mémoire s'articule autour de cinq chapitres. Dans le premier chapitre, figurent les rappels fondamentaux concernant les propriétés mathématiques des treillis, de dioïdes, ainsi qu'un ensemble de définition sur la programmation linéaire et quelques résultats généraux qui éclairent cette thèse. Ces résultats généraux sont ceux, assez anciens, de Farkas (1894) et de Veinott (dont on retrouvera une synthèse dans son article de 1989). Le premier prouve l'existence d'une solution d'un système d'inégalités  $Ax \leq b$ , et le deuxième établit la dualité entre les familles de polyèdres  $P = \{x | Ax \leq b\}$  ayant une structure de demi-treillis et les systèmes d'inégalités dites monotones. Une propriété analyse l'existence du plus petit élément (du plus grand élément). Ce résultat simplifie considérablement la résolution de ce type de problèmes (il les fait passer, en effet, de la classe des problèmes combinatoires difficiles à la classe des problèmes bien résolus). Nous avons finalement emprunté le terme d'inégalités *bi-monotones* de Hochbaum & Naor (1994) qui l'utilisent notamment dans l'étude d'algorithmes spécialisés pour des systèmes à deux variables par inégalité.

Dans le second chapitre, intitulé réseaux de Pétri temporisés et temporels, et leurs modèles algébriques, nous avons utilisé les dateurs et les compteurs pour présenter une modélisation décrite dans l'algèbre standard afin de décrire un modèle linéaire sous la forme  $A.x \leq b$ . La matrice  $A$  contient les matrices d'incidences du graphe d'événements considéré.

Le problème d'évaluation de performances des graphes d'événements P-temporels fait l'objet du troisième chapitre. Habituellement dans l'algèbre  $(\max,+)$ , on applique certains résultats de base de la théorie spectrale, pour évaluer les performances des graphes d'événements temporisés. Les résultats de ce chapitre ouvrent la voie à d'autres développements liés à l'analyse des performances des graphes d'événements P-temporels. Nous proposons une approche liée à la pro-

grammation linéaire pour prouver l'existence du temps de cycle (Lemme de Farkas), son calcul utilisant une fonction objectif.

Nous avons ensuite proposé une standardisation du modèle algébrique des graphes d'événements temporisés définis dans l'algèbre standard. Le résultat de la technique proposée est un modèle auto-régressif similaire aux équations d'état de l'automatique classique. Basé sur les matrices d'incidences standard, ce nouveau modèle est un ensemble d'inégalités d'entrées/sorties, qui permet un calcul efficace de la trajectoire d'état connaissant l'état initial et la commande  $u$ . Le point de départ de cette approche est le modèle algébrique déduit du graphe d'événements temporisé initial. Utilisant la programmation linéaire, nous avons proposé deux techniques duales qui permettent la construction du modèle standard et le graphe d'événements temporisé correspondant. Cette étude fait le lien entre la programmation linéaire et la théorie des graphes.

Nous avons calculé la trajectoire au plus tôt en utilisant deux techniques. Pour la première technique, nous avons prouvé que la trajectoire d'état minimale du modèle d'entrées/sorties existe. L'unicité de cette solution minimale est assurée par le concept de relation d'ordre composante par composante et l'existence d'un système inf-monotone. Une deuxième technique, basée sur la programmation linéaire, est définie avec un problème primal  $P_{fw}$  pour calculer cette trajectoire. Nous avons utilisé l'environnement **LPSolve IDE** pour résoudre le système d'inégalités. La complexité de notre résolution est polynomiale dans le pire des cas. Un problème dual  $D_{fw}$  est calculé.

La dernière partie traite de la synthèse de la commande selon le critère de juste-à-temps. Cette méthode se base sur la structure du polyèdre  $A.x \leq b$  et aboutit à deux techniques de résolution comme le cas du calcul de la trajectoire au plus tôt. Une perspective à court terme est de trouver la signification physique des solutions des problèmes duaux  $D_{fw}$  et  $D_{bw}$ .

Enfin, une commande effectuant une poursuite de trajectoire sur un horizon glissant est calculée. Après un contrôle optimal sur l'horizon  $[k_s + 1; k_f]$  à l'étape  $k_s$ , l'horizon est déplacé et devient  $[k_s + 2; k_f + 1]$  : la synthèse de la commande selon le critère juste-à-temps est mise à jour avec une nouvelle information sur les mesures et une nouvelle optimisation à l'étape  $k_s + 1$  doit être effectuée. Alors, nous avons établi une synthèse des deux techniques pour calculer la trajectoires au plus tôt et la trajectoire au plus tard.

Une perspective est clairement d'introduire des contraintes additionnelles. On peut ainsi désirer commander un graphe d'événements temporisé tout en respectant des spécifications comme un ou plusieurs graphes d'événements temporel du type T-temporel, P-temporel,... Le problème est alors de disposer d'algorithmes efficaces pouvant calculer en ligne la commande sous peine de ne pas pouvoir appliquer la démarche. Dans ce mémoire, la complexité de notre algorithme

de résolution pour le calcul de la trajectoire au plus tôt et la synthèse de commande, mais sans la considération de spécifications, est en  $O(n^2)$ . Une possibilité est d'utiliser des algorithmes génériques comme le Simplex ou plus récents (Khashiyan, Karmarkar,...). Cependant, la complexité de ces algorithmes est au mieux pseudo-polynomiale (exponentielle pour le Simplex), ce qui peut être une limitation lorsque l'échelle de grandeur des paramètres est importante. Il est donc plus intéressant d'utiliser des algorithmes comme celui de Hochbaum [Hochbaum 1993], où la complexité est en  $O(mn^2 \log n)$ . La contre-partie est cependant une limitation de la classe des systèmes considérés. Il s'agira alors de définir les problèmes pratiques où ces algorithmes peuvent être appliqués.

Une autre perspective de recherche peut porter sur l'optimisation des ressources des graphes d'événements temporisés et P-temporels. Le principe consisterait à minimiser le nombre de ressources (machines, palettes, capacité de stocks...) d'un système. Pour une structure du modèle (GET ou GE P-temporels) donné (mais sans connaissance *a priori* du marquage initial), avec un comportement 1-périodique, le taux de production doit être compris entre deux bornes données, minimale et maximale. L'objectif serait de calculer le marquage optimal selon un coût fonction du marquage.

Enfin, l'élargissement des modèles considérés est aussi une perspective. Ce travail a déjà commencé sur les graphes d'événements T-temporels dans le cadre de ce mémoire. Mais, on peut aussi déterminer le modèle algébrique des graphes d'événements valués ce qui exigera l'emploi non des dateurs mais des compteurs, ainsi que la prise en compte du caractère entier des compteurs. Les perspectives sont ainsi larges.



# Annexes



# Annexe A

## La résiduation

### A.1 Éléments de la théorie de la résiduation

La résolution de problèmes qui se résument à l'inversion d'applications dans l'algèbre usuelle, est souvent réalisée à l'ordre de la propriété de bijection de ces applications. En effet, si  $f$  est une application bijective définie sur  $\mathbb{R}, \mathbb{Z}, \dots$ , on peut définir une application inverse notée  $f^{-1}$ . En revanche, si on considère une application  $\Pi$  définie sur un dioïde, et sachant que les opérateurs  $\oplus$  et  $\otimes$  ne sont pas inversibles, alors  $\Pi^{-1}$  ne peut pas être toujours définie directement. C'est dans ce contexte que la théorie de la résiduation intervient comme une alternative à la notion d'inversibilité. Cette théorie concerne uniquement les applications monotones définies sur des ensembles ordonnés.

Après introduction des définitions de base sur la monotonie, les homomorphismes, et la semi-continuité, nous exposerons un ensemble de résultats importants issus de cette théorie. Une exploitation directe de ces résultats va permettre d'étudier (existence et calcul) la plus petite et la plus grande solution respectivement des inéquations suivantes :

$$x \leq \Pi(x), \tag{A.1}$$

$$\Pi(x) \leq x. \tag{A.2}$$

Le lecteur pourra trouver des études détaillées sur les notions de la résiduation dans [Blyth 1972], [Baccelli 1992], [Cohen 1998], [Cottenceau 1999] et [Menguy 1997].

#### A.1.1 Homomorphismes de dioïdes, isotonie et continuité

**Définition A.1. (Isotonie, antitonie)**

Soit  $\Pi$  une application définie sur des ensembles ordonnés  $\Pi : E \longrightarrow F$ .

$\Pi$  est dite isotone si :

$$\forall x, y \in E, \quad x \leq y \implies \Pi(x) \leq \Pi(y).$$

$\Pi$  est dite antitone si :

$$\forall x, y \in E, \quad x \leq y \implies \Pi(x) \geq \Pi(y).$$

**Remarque :** Une application est dite monotone si elle est isotone ou antitone. ◇

**Définition A.2. (Homomorphisme)**

Une application  $\Pi$ , définie d'un dioïde  $\mathcal{D}$  vers un dioïde  $\mathcal{C}$ , est un homomorphisme si  $\forall x, y \in \mathcal{D}$

$$\Pi(x \oplus y) = \Pi(x) \oplus \Pi(y) \quad \text{et} \quad \Pi(\varepsilon) = \varepsilon, \tag{A.3}$$

$$\Pi(x \otimes y) = \Pi(x) \otimes \Pi(y) \quad \text{et} \quad \Pi(e) = e. \tag{A.4}$$

Notons que chaque application qui vérifie la propriété A.3 sera appelée  $\oplus$ -morphisme. Une application qui ne vérifie que la propriété A.4 sera appelée  $\otimes$ -morphisme. Un homomorphisme est donc un  $\oplus$ -morphisme et un  $\otimes$ -morphisme.

**Remarque A.3**

Si l'application  $\Pi$  est un  $\oplus$ -morphisme, alors  $\Pi$  est forcément isotone. En effet, soit  $x \leq y$ , d'où  $x \oplus y = y$  et  $\Pi(x \oplus y) = \Pi(y) = \Pi(x) \oplus \Pi(y)$ . D'où le résultat  $\Pi(x) \leq \Pi(y)$ .

Dualement, soit  $\Pi$  est un  $\wedge$ -morphisme. Si  $x \leq y$ , alors  $\Pi(x \wedge y) = \Pi(x) = \Pi(x) \wedge \Pi(y)$ . On en déduit que  $\Pi(x) \leq \Pi(y)$ , d'où  $\Pi$  est isotone. ◇

La réciproque est cependant fautive comme le montre le lemme suivant. La démonstration permet de mieux saisir ces équivalences.

**Lemme A.4. [Baccelli 1992]**

Soit  $\Pi$  une application d'un dioïde  $\mathcal{D}$  dans un autre dioïde  $\mathcal{C}$ . Alors les propositions suivantes sont équivalentes :

1. L'application  $\Pi$  est isotone.
2. L'application  $\Pi$  est  $\oplus$ -supermorphisme, c'est-à-dire :

$$\forall x, y \in \mathcal{D}, \quad \Pi(x \oplus y) \geq \Pi(x) \oplus \Pi(y)$$

3. Si la borne inférieure existe dans  $\mathcal{D}$  et  $\mathcal{C}$ ,  $\Pi$  est un  $\wedge$ -sousmorphisme, c'est-à-dire :

$$\forall x, y \in \mathcal{D}, \quad \Pi(x \wedge y) \leq \Pi(x) \wedge \Pi(y).$$

**Preuve**

(1  $\Leftrightarrow$  2) Supposons que  $\Pi$  est isotone. Puisque  $x$  et  $y$  sont inférieurs à  $x \oplus y$ , alors,  $\Pi(x)$  et  $\Pi(y)$ , ainsi que leur borne supérieure  $\Pi(x) \oplus \Pi(y)$ , sont inférieurs à  $\Pi(x \oplus y)$ .

Inversement, soit  $y \leq x$ , ou de façon équivalente,  $x = x \oplus y$ . Alors,  $\Pi(x) = \Pi(x \oplus y) \geq \Pi(x) \oplus \Pi(y)$ ,

ce qui montre que  $\Pi(y) \leq \Pi(x)$ , d'où  $\Pi$  est isotone. Alors, les deux points 1 et 2 sont équivalents. (1  $\Leftrightarrow$  3) Soit  $\Pi$  isotone. Puisque  $x$  et  $y$  sont supérieurs à  $x \wedge y$ , alors,  $\Pi(x)$  et  $\Pi(y)$ , ainsi que leur borne inférieure (supposée existante)  $\Pi(x) \wedge \Pi(y)$ , sont supérieurs à  $\Pi(x \wedge y)$ .

Inversement, soit  $y \geq x$ , ou de façon équivalente,  $x = x \wedge y$ . Alors,  $\Pi(x) = \Pi(x \wedge y) \leq \Pi(x) \wedge \Pi(y)$ , ce qui montre que  $\Pi(y) \geq \Pi(x)$ , d'où  $\Pi$  est isotone. D'où l'équivalence entre les deux points 1 et 3.

### Remarque A.5

Si  $\mathcal{D}$  est une chaîne,  $x \oplus y$  est égal à  $x$  ou  $y$ . Par conséquent,  $\Pi(x \oplus y)$  est égal à  $\Pi(x)$  ou  $\Pi(y)$ . Si  $\Pi$  est isotone, on peut appliquer le lemme A.4 et donc  $\Pi(x \oplus y) = \Pi(x) \oplus \Pi(y)$ , et donc  $\Pi$  est  $\oplus$ -morphisme dans ce cas.

Dualement,  $\Pi$  est  $\wedge$ -morphisme. ◇

### Définition A.6. (Semi-continuité inférieurement)

Une application  $\Pi$  d'un dioïde complet  $\mathcal{D}$  dans un dioïde complet  $\mathcal{C}$  est semi-continue inférieurement, en abrégé S.C.I. si, pour tout sous ensemble (fini ou infini)  $X$  de  $\mathcal{D}$ ,

$$\Pi\left(\bigoplus_{x \in X} x\right) = \bigoplus_{x \in X} \Pi(x). \quad (\text{A.5})$$

### Définition A.7. (Semi-continuité supérieurement)

Une application  $\Pi$  d'un dioïde complet  $\mathcal{D}$  dans un dioïde complet  $\mathcal{C}$  est semi-continue supérieurement, en abrégé S.C.S. si, pour tout sous ensemble (fini ou infini)  $X$  de  $\mathcal{D}$ ,

$$\Pi\left(\bigwedge_{x \in X} x\right) = \bigwedge_{x \in X} \Pi(x). \quad (\text{A.6})$$

Nous remarquons que toute application S.C.I. (respectivement, S.C.S.) est un  $\oplus$ -morphisme (respectivement, est un  $\wedge$ -morphisme) et donc un  $\oplus$ -supermorphisme (respectivement un  $\wedge$ -sousmorphisme). D'après le lemme A.4,  $\Pi$  est isotone.

## A.1.2 Applications résiduelles et dualement résiduelles

Nous considérons d'abord des ensembles ordonnés.

### Définition A.8. Application résiduable

Soit  $\Pi$  une application isotone définie d'un ensemble ordonné  $(\mathcal{D}, \leq)$  dans un ensemble ordonné  $(\mathcal{C}, \leq)$ . L'application  $\Pi$  est dite résiduable, si pour tout  $y \in \mathcal{C}$  l'équation  $\Pi(x) \leq y$  admet une plus grande solution dans  $\mathcal{D}$ . L'application résiduée, noté  $\Pi^\sharp$ , est définie par

$$\Pi^\sharp(y) = \bigoplus \{x \in \mathcal{D} \mid \Pi(x) \leq y\}$$

.

La définition duale est donnée comme suit :

**Définition A.9. Application dualement résiduable**

Soit  $\Pi$  une application isotone définie d'un ensemble ordonné  $(\mathcal{D}, \leq)$  dans un ensemble ordonné  $(\mathcal{C}, \leq)$ . L'application  $\Pi$  est dite *dualement résiduable*, si pour tout  $y \in \mathcal{C}$  l'équation  $\Pi(x) \geq y$  admet une plus petite solution dans  $\mathcal{D}$ . L'application résiduée duale  $\Pi^\flat$  est définie par  $\Pi^\flat(y) = \bigwedge \{x \in \mathcal{D} \mid \Pi(x) \geq y\}$ .

Les démonstrations de l'ensemble des théorèmes A.10, A.11, A.12 et A.13 sont exposées en détail dans : [Blyth 1972], [Baccelli 1992], [Cohen 1998] et [Menguy 1997].

**Théorème A.10.** Soit  $\Pi$  une application isotone de  $(\mathcal{D}, \leq)$  dans  $(\mathcal{C}, \leq)$ .

Alors sont équivalents :

1. L'application  $\Pi$  est résiduable.
2. Il existe une application  $\Pi^\sharp$  de  $\mathcal{C}$  dans  $\mathcal{D}$  isotone telle que

$$\Pi \circ \Pi^\sharp \leq I_{\mathcal{C}} \quad \text{avec } I_{\mathcal{C}} \text{ identité de } \mathcal{C}, \quad (\text{A.7})$$

$$\Pi^\sharp \circ \Pi \geq I_{\mathcal{D}} \quad \text{avec } I_{\mathcal{D}} \text{ identité de } \mathcal{D}. \quad (\text{A.8})$$

Le théorème A.11 dual relatif à l'application dualement résiduable est le suivant.

**Théorème A.11.** Soit  $\Pi$  une application isotone de  $(\mathcal{D}, \leq)$  dans  $(\mathcal{C}, \leq)$ . Sont équivalents :

1. L'application  $\Pi$  est dualement résiduable.
2. Il existe une application  $\Pi^\flat$  de  $\mathcal{C}$  dans  $\mathcal{D}$  isotone telle que

$$\Pi \circ \Pi^\flat \geq I_{\mathcal{C}} \quad \text{avec } I_{\mathcal{C}} \text{ identité de } \mathcal{C}, \quad (\text{A.9})$$

$$\Pi^\flat \circ \Pi \leq I_{\mathcal{D}} \quad \text{avec } I_{\mathcal{D}} \text{ identité de } \mathcal{D}. \quad (\text{A.10})$$

**Unicité de  $\Pi^\sharp$  et  $\Pi^\flat$** 

Du fait que l'application  $\Pi$  est résiduable, cela implique que l'application  $\Pi^\sharp$  est unique. Ceci peut être montré par l'absurde en considérant qu'il existe une autre application  $\Psi$  qui satisfait le point 2 du théorème A.10. Nous constatons alors que :

$$\Pi^\sharp = I_{\mathcal{D}} \circ \Pi^\sharp \leq (\Psi \circ \Pi) \circ \Pi^\sharp = \Psi \circ (\Pi \circ \Pi^\sharp) \leq \Psi \circ I_{\mathcal{C}} = \Psi,$$

$$\Psi = I_{\mathcal{D}} \circ \Psi \leq (\Pi^\sharp \circ \Pi) \circ \Psi = \Pi^\sharp \circ (\Pi \circ \Psi) \leq \Pi^\sharp \circ I_{\mathcal{C}} = \Pi^\sharp.$$

$\implies \Psi = \Pi$ .

De façon similaire, on peut montrer que lorsque  $\Pi$  est dualement résiduable, l'application  $\Pi^\flat$  est unique.

Les théorèmes A.12 et A.13 suivants considèrent des dioïdes.

**Théorème A.12.** Soit  $\Pi$  une application isotone d'un dioïde complet  $\mathcal{D}$  dans un dioïde complet  $\mathcal{C}$ . Sont équivalents :

1. L'application  $\Pi$  est résiduable.
2.  $\Pi$  est S.C.I. et  $\Pi(\varepsilon) = \varepsilon$ .

Le théorème dual de A.12 est le suivant.

**Théorème A.13.** *Soit  $\Pi$  une application isotone d'un dioïde complet  $\mathcal{D}$  dans un dioïde complet  $\mathcal{C}$ . Sont équivalents :*

1.  $\Pi$  est une application dualement résiduable.
2.  $\Pi$  est S.C.S. et  $\Pi(T) = T$ .

Le théorème suivant donne quelques propriétés existant entre des applications résiduables et leurs résiduées ainsi que pour les applications dualement résiduables et leurs résiduées duales.

**Théorème A.14.** [*Baccelli 1992*]

1. Si les deux applications  $\Pi : \mathcal{D} \rightarrow \mathcal{D}$  et  $\Phi : \mathcal{D} \rightarrow \mathcal{D}$  sont résiduables dans un dioïde complet  $\mathcal{D}$ , alors  $\Pi \oplus \Phi$  est résiduable

$$(\Pi \oplus \Phi)^\sharp = \Pi^\sharp \wedge \Phi^\sharp. \quad (\text{A.11})$$

2. Si les deux applications  $\Pi : \mathcal{D} \rightarrow \mathcal{D}$  et  $\Phi : \mathcal{D} \rightarrow \mathcal{D}$  sont dualement résiduables dans un dioïde complet  $\mathcal{D}$ , alors  $\Pi \wedge \Phi$  est aussi dualement résiduable

$$(\Pi \wedge \Phi)^\flat = \Pi^\flat \oplus \Phi^\flat. \quad (\text{A.12})$$

### A.1.2.1 Résiduation de l'addition et de la multiplication

Les applications  $T_a$  (translation),  $L_a$  (multiplication à gauche) et  $R_a$  (multiplication à droite) sur un dioïde  $\mathcal{D}$  complet sont définies comme suit :

$$T_a : x \mapsto a \oplus x, \quad (\text{A.13})$$

$$L_a : x \mapsto a \otimes x, \quad (\text{A.14})$$

$$R_a : x \mapsto x \otimes a. \quad (\text{A.15})$$

Les références [Blyth 1972], [Baccelli 1992] et [Gaubert 1992] proposent plus de détails sur les propriétés relatives aux applications :  $L_a^\sharp$ ,  $R_a^\sharp$  et  $T_a^\flat$ .

Le fait que le dioïde  $\mathcal{D}$  soit complet implique que les applications  $L_a$  et  $R_a$  sont semi-continues inférieurement (distributivité de  $\otimes$  sur  $\oplus$  pour des ensembles finis ou infinis). En outre, comme  $\varepsilon$  est absorbant pour  $\otimes$ , alors  $L_a(\varepsilon) = a \otimes \varepsilon = \varepsilon$  et  $R_a(\varepsilon) = \varepsilon \otimes a = \varepsilon$ . D'après le théorème A.12, les applications  $L_a$  et  $R_a$  sont résiduables. On peut donc considérer  $L_a^\sharp$  et  $R_a^\sharp$  les applications résiduées respectivement de  $L_a$  et  $R_a$  qui sont définies par :

$$L_a^\sharp(b) = \bigoplus \{x \mid L_a(x) = a \otimes x \leq b\} = \frac{b}{a} = a \wp b,$$

$$R_a^\sharp(b) = \bigoplus \{x \mid R_a(x) = x \otimes a \leq b\} = \frac{b}{a} = a \not\phi b.$$

A l'opposé,  $T_a$  n'est pas en général résiduable car la condition  $T_a(\varepsilon)$  n'est pas vérifiée (sauf pour  $a = \varepsilon$ ). Cependant, si le dioïde  $\mathcal{D}$  est distributif, alors l'application  $T_a$  est semi-continue supérieurement. De plus,  $T_a(T) = a \oplus T = T$ , d'où  $T_a$  est dualement résiduée. La résiduée duale de  $T_a$  est donnée par :

$$T_a^\flat(b) = \bigwedge \{x \mid T_a(x) = a \oplus x \geq b\}$$

le théorème qui suit regroupe un ensemble de propriétés concernant les applications  $L_a$  et  $R_a$ .

### Théorème A.15.

$$\begin{array}{lll}
a \not\phi (x \wedge y) = a \not\phi x \wedge a \not\phi y & f.1 & (x \wedge y) \not\phi a = x \not\phi a \wedge y \not\phi a \\
a \not\phi (x \oplus y) \geq a \not\phi x \oplus a \not\phi y & f.2 & (x \oplus y) \not\phi a \geq x \not\phi a \oplus y \not\phi a \\
(a \oplus b) \not\phi x = a \not\phi x \wedge b \not\phi x & f.3 & x \not\phi (a \oplus b) = x \not\phi a \wedge x \not\phi b \\
(a \wedge b) \not\phi x \geq a \not\phi x \oplus b \not\phi x & f.4 & x \not\phi (a \wedge b) \geq x \not\phi a \oplus x \not\phi b \\
a(a \not\phi x) \leq x & f.5 & (x \not\phi a)a \leq x \\
a \not\phi (ax) \geq x & f.6 & (xa) \not\phi a \geq x \\
a(a \not\phi (ax)) = ax & f.7 & ((xa) \not\phi a)a = xa \\
a \not\phi (a(a \not\phi x)) = a \not\phi x & f.8 & ((x \not\phi a)a) \not\phi a = x \not\phi a \\
(ab) \not\phi x = b \not\phi (a \not\phi x) & f.9 & x \not\phi (ba) = (x \not\phi a) \not\phi b \\
(a \not\phi x) \not\phi b = a \not\phi (x \not\phi b) & f.10 & b \not\phi (x \not\phi a) = (b \not\phi x) \not\phi a \\
b(a \not\phi x) \leq (a \not\phi b) \not\phi x & f.11 & (x \not\phi a)b \leq x \not\phi (b \not\phi a) \\
(a \not\phi x)b \leq a \not\phi (xb) & f.12 & b(x \not\phi a) \leq (bx) \not\phi a \\
(a \not\phi x) \oplus b \leq a \not\phi (x \oplus ab) & f.13 & (x \not\phi a) \oplus b \leq (x \oplus ba) \not\phi a
\end{array}$$

Dans le dioïde  $\overline{\mathbb{R}}_{max}$ , on peut noter que dans le cas où deux éléments d'un dioïde  $a$  et  $b$  sont scalaires et finis, alors une résiduation à gauche  $a \not\phi b$  ou à droite  $b \not\phi a$  se résume à une soustraction dans l'algèbre usuelle, soit  $b - a$ . Néanmoins, ceci ne couvre pas les cas avec  $a$  ou  $b$  infini. Les règles de manipulation sont données comme suit :

$$\begin{aligned}
a \backslash b &= b - a, & \text{si } a \text{ et } b \text{ sont finis,} \\
a \backslash T &= T, & \text{pour tout } a, \\
a \backslash \varepsilon &= \varepsilon, & \text{pour tout } a \text{ fini,} \\
\varepsilon \backslash a &= T, & \text{pour tout } a, \\
T \backslash a &= \varepsilon, & \text{pour } a \neq T.
\end{aligned}$$

Remarquons aussi que  $T \otimes \varepsilon = \varepsilon = -\infty$  alors que  $\varepsilon \backslash \varepsilon = T$ .

On note que ces règles sont valables aussi dans le cas où l'on a une résiduation à droite  $\phi$ . Il en résulte les mêmes règles d'opérations dans le dioïde  $\overline{\mathbb{R}}_{min}$  avec  $\varepsilon = +\infty$  et  $T = -\varepsilon$ .

On peut étendre la relation  $a \backslash b = b - a$  aux valeurs infinies si l'on prend la convention  $+\infty - \infty = +\infty$

### A.1.2.2 Résiduation dans le cas matriciel

Tous les résultats et propriétés présentés précédemment restent valables dans le cas matriciel à condition que les dimensions des matrices soient compatibles. On considère l'application  $L_A$  donnée par

$$\begin{aligned}
L_A : \mathcal{D}^{p \times q} &\rightarrow \mathcal{D}^{n \times q}, \\
X &\mapsto A \otimes X \quad (A \in \mathcal{D}^{n \times p}).
\end{aligned}$$

**Théorème A.16.** Soit  $B \in \mathcal{D}^{n \times q}$ . La plus grande solution de  $Ax \leq B$  est la matrice  $L_A^\sharp(B) \in \mathcal{D}^{p \times q}$ , que l'on peut noter  $\frac{B}{A} = A \backslash B$ .

$$(A \backslash B)_{ij} = \bigwedge_{k=1}^n A_{ki} \backslash B_{kj}.$$

**Preuve :** Les propositions suivantes sont équivalentes :

$$\begin{aligned}
A \otimes X &\leq B, \\
(A \otimes X)_{kj} &= \bigoplus_{i=1}^p (A_{ki} X_{ij}) \leq B_{kj}, & \forall k \in [1, n], \forall j \in [1, q], \\
A_{ki} X_{ij} &\leq B_{kj}, & \forall k \in [1, n], \forall j \in [1, q], \forall i \in [1, p], \\
X_{ij} &\leq A_{ki} \backslash B_{kj}, \\
X_{ij} &\leq \bigwedge_{k=1}^n A_{ki} \backslash B_{kj}, & \forall j \in [1, q], \forall i \in [1, p].
\end{aligned}$$

□

De même pour l'application

$$\begin{aligned}
R_A : \mathcal{D}^{q \times n} &\rightarrow \mathcal{D}^{q \times p}, \\
X &\mapsto X \otimes A \quad (A \in \mathcal{D}^{n \times p}).
\end{aligned}$$

Soit  $B \in \mathcal{D}^{q \times p}$ , alors :

$$(B \phi A)_{ij} = \bigwedge_{k=1}^p B_{ik} \phi A_{jk}.$$

**Remarque A.17**

Nous remarquons d'une part que  $(A \setminus B)_{ij} = \bigwedge_{k=1}^n A_{ki} \setminus B_{kj} = \bigwedge_{k=1}^n (B_{kj} - A_{ki}) = \bigwedge_{k=1}^n (B_{kj} + (-A_{ik}^t))$  avec  $A^t$  la transposée de  $A$  et la convention ci-dessus. D'autre part,  $(A \odot B)_{ij} = \bigwedge_{k=1}^n A_{ik} \odot B_{kj} = \bigwedge_{k=1}^n A_{ik} + B_{kj}$ , d'où le résultat suivant :

$$A \setminus B = (-A^t) \odot B.$$

De la même manière, on peut vérifier que :

$$A \setminus' B = (-A^t) \otimes B.$$

◇

## Annexe B

# Théorie des graphes et algorithme du Simplexe

### B.1 Rappel de la théorie des graphes

**Définition B.1.** Un *graphe orienté*  $G$  est un graphe défini par deux ensembles : un ensemble de *nœuds* (ou *sommets*), noté  $N$ , et un ensemble d'*arcs orientés*, noté  $A$ .

**Définition B.2.** Un *chemin*, noté  $\omega$ , est une suite de *nœuds* que l'on peut parcourir successivement en empruntant les arcs de  $A$ . On note  $i \rightarrow j$  un chemin allant d'un *nœud d'origine*, noté  $i$ , vers un *nœud d'extrémité*, noté  $j$ .

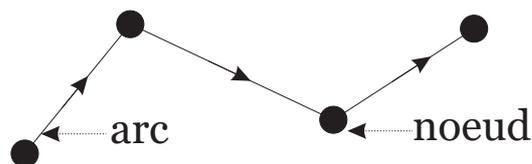


Fig. B.1 : Un chemin.

**Définition B.3.** On appelle *circuit* une suite d'*arcs consécutifs* (chemin) dont les deux sommets (place ou transition) extrémités sont identiques.

Si le chemin est **élémentaire**, c'est-à-dire, ne passe pas deux fois par un même sommet, on parle de **circuit élémentaire**. On note le circuit élémentaire  $\zeta$ . Un circuit élémentaire ne contient pas d'autre circuit.

**Définition B.4.** (Composante fortement connexe). Une *composante fortement connexe* est une classe d'équivalence induite par une relation d'équivalence, notée  $\Delta$ , définie sur  $N$  comme suit :

$$i\Delta j \Leftrightarrow i \rightarrow j \text{ et } j \rightarrow i :$$

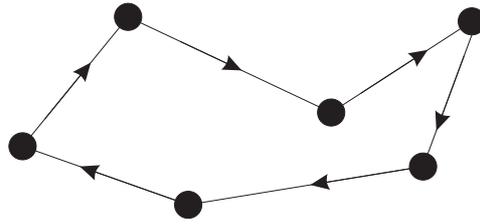


Fig. B.2 : Un circuit élémentaire.

Un graphe est **fortement connexe** s'il est composé d'une seule composante fortement connexe

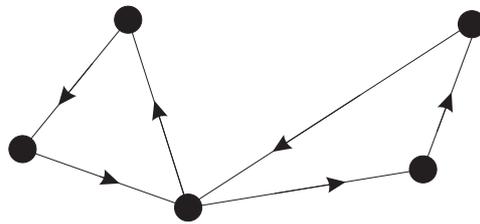


Fig. B.3 : Composante fortement connexe.

## B.2 Algorithme du Simplexe

### B.2.1 Définition - Notations

Le problème linéaire est le suivant :

Trouver les  $x_1, x_2, \dots, x_n$  qui optimisent la fonction :

$$c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

et vérifient les contraintes :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1,$$

...

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \leq b_k,$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m,$$

$$x_1 \geq 0; x_2 \geq 0; \dots; x_n \geq 0,$$

avec  $c_1 \dots c_n, a_{11} \dots a_{mn}, b_1 \dots b_m$  connus.

En notation condensée :

$$\min z = C^T X \text{ sous les contraintes } A.X \leq b \text{ et } X \geq 0$$

avec  $C(n \times 1), A(m \times n)$  et  $b(m \times 1)$  connus et  $X(n \times 1)$ .

### B.2.1.1 Forme standard

La forme standard est un problème d'optimisation sous la forme :

$$\begin{aligned} \min z &= C^t X \text{ sous } A.X = b \text{ et } X \geq 0 \\ &\text{avec } C(n \times 1) \text{ et } A(m \times n) \text{ connus.} \end{aligned}$$

### B.2.1.2 Mise sous forme standard

On peut toujours remplacer les inégalités :

$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n \leq b_k$$

par des égalités, en introduisant des variables d'écart.  $a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n + e_k = b_k$  avec  $e_k$  positif ou nul.

Le problème devient :

$$\min \hat{z} = \begin{bmatrix} C^t & 0^t \end{bmatrix} \times \begin{bmatrix} X \\ e \end{bmatrix} \text{ sous } \begin{bmatrix} A & 1 \end{bmatrix} \times \begin{bmatrix} X \\ e \end{bmatrix} = b \text{ et } \begin{bmatrix} X \\ e \end{bmatrix} \geq 0 \text{ avec } C, A \text{ et } b \text{ connus.}$$

## B.2.2 Définitions

**Définition B.5.** On appelle solution tout vecteur  $X$  tel que  $A \times X = b$ .

**Définition B.6.** Une solution  $X$  qui vérifie  $\begin{cases} A \times X = b \\ X \geq 0 \end{cases}$  est une solution admissible.

**Définition B.7.**  $A$  étant une matrice  $(m \times n)$  de rang  $m$ . Une base est une matrice  $B(m \times m)$  extraite de  $A$  dont le déterminant est non nul.

On remarque que le système à résoudre possède  $m$  égalités et  $n + m$  inconnues. La valeur de  $n$  variables peut être fixées arbitrairement, par exemple à zéro. On dit qu'elles sont mises **hors base**.

**Définition B.8.** On appelle variables hors base les  $n$  variables de  $\mathbb{R}^{n+m}$  fixées à zéros. Les  $m$  variables restantes sont appelées variables de base.

**Définition B.9.** On appelle **solution de base** une solution où en ayant choisi  $n$  variables hors base, on obtient une solution unique en résolvant les  $m$  contraintes d'égalité obtenues en ajoutant les variables d'écart.

Notons  $x_B$  les variables de base correspondant aux  $m$  variables positives, ou nulles, et  $x_H$  les variables hors base, correspondant aux  $n$  variables nulles.

**Définition B.10.** Une solution de base telle que  $x_B \geq 0$  est dite admissible.

**Définition B.11.** Si  $x_B \geq 0$ , alors la solution est dite solution de base admissible dégénérée.

### B.2.3 Résultats fondamentaux

#### B.2.3.1 Définitions

**Définition B.12.** Soient  $X$  et  $Y$  deux points d'un domaine  $K$ , le segment qui joint  $X$  et  $Y$  est défini par l'ensemble des  $U$  tels que  $U = \lambda X + (1 - \lambda)Y$  avec  $0 \leq \lambda \leq 1$ .

**Définition B.13.** Soient  $X$  et  $Y$  deux points intérieurs de  $K$ , le domaine  $K$  sera convexe si le segment joignant  $X$  et  $Y$  appartient à  $K$ .

**Définition B.14.** Une fonction  $f$  est convexe dans un domaine convexe  $K$  si  $f(\lambda X + (1 - \lambda)Y) \leq \lambda f(X) + (1 - \lambda)f(Y)$ .

**Définition B.15.**  $U$  est un sommet du domaine  $K$  ssi  $\begin{cases} U = \lambda X + (1 - \lambda)Y \\ 0 < \lambda < 1 \end{cases} \Rightarrow U = X = Y$ .

Autrement dit, un point d'un polyèdre est un sommet s'il ne peut pas être vu comme la combinaison linéaire convexe de deux autres points du polyèdre. Tout point d'un polyèdre peut être vu comme une combinaison linéaire convexe des sommets du polyèdre.

#### B.2.3.2 Théorèmes

**Théorème B.16.** S'il existe une solution admissible au problème  $A \cdot X = b, X \geq 0$  alors il existe une solution de base admissible. S'il existe une solution optimale admissible, il existe une solution de base admissible optimale.

**Théorème B.17.** L'ensemble des solutions admissibles forme un ensemble convexe  $K$  on a :

**Preuve** Soient  $X$  et  $Y$  deux solutions dans  $K$ .

$$\begin{cases} AX = bX & \geq 0 & X \in K \\ AY = bY & \geq 0 & Y \in K \end{cases} .$$

Alors :

$$\begin{cases} \lambda AX = \lambda b \\ (1 - \lambda)AY = (1 - \lambda)b \end{cases} \Rightarrow A[\lambda X + (1 - \lambda)Y] = b.$$

Soit  $U = \lambda X + (1 - \lambda)Y$  alors  $AU = b$  donc  $U \in K$  et  $U \geq 0$ .

**Théorème B.18.** Si  $K$  est un ensemble convexe, un point de  $K$  solution admissible de  $A \cdot X = b, X \geq 0$  est un sommet de  $K$  ssi  $X$  est une solution de base admissible.

**Théorème B.19.** Toute solution de  $A \cdot X = b, X \geq 0$  est une combinaison linéaire convexe des sommets. La solution optimale est donc une combinaison linéaire convexe des sommets.

**Théorème B.20.** L'optimum de  $z$ , s'il existe, est atteint en au moins un sommet de  $K$ .

**Théorème B.21.** Si  $z$  atteint son optimum pour plusieurs sommets, il l'atteint aussi pour toute combinaison linéaire convexe de ces sommets.

### B.2.4 Principe

L'algorithme procède de la façon suivante :

1. on recherche un sommet de départ ;
2. on teste si ce sommet est l'optimum ou si la fonction objectif n'est pas bornée inférieurement ; dans ce cas le problème n'a pas de solution finie ;
3. si le sommet que l'on vient d'examiner n'est pas optimal, on se déplace sur un sommet voisin pour lequel la fonction diminue et on repasse à l'étape précédente. Le nombre de sommets étant fini et tout minimum local étant absolu, le sommet optimal est atteint lorsqu'aucun des sommet voisin ne permet plus la diminution du critère.

### B.2.5 Recherche d'un sommet de départ

#### Forme simpliciale

Le problème étant sous la forme standard B.2.1.1, il peut se décomposer sous la forme

$$\begin{bmatrix} B & H \end{bmatrix} \cdot \begin{bmatrix} x_B \\ x_H \end{bmatrix} = b,$$

avec  $x_B$  les variables de base et  $x_H$  les variables hors base.

Supposons que  $B$  est inversible, alors  $\begin{bmatrix} I & B^{-1}H \end{bmatrix} \cdot \begin{bmatrix} x_B \\ x_H \end{bmatrix} = B^{-1}b$ .

Si  $B^{-1}b \geq 0$  alors la solution

$$x_B = B^{-1}b \text{ et } x_H = 0,$$

est une solution de base admissible.

Cette approche nécessite la recherche d'une matrice  $B$  telle que  $B$  soit inversible et  $B^{-1}b \geq 0$ .

La forme simpliciale décrite ci-dessous permet d'éliminer ces inconvénients. En effet, la mise en évidence d'une solution est particulièrement simple si  $B = I$  et  $b \geq 0$ .

On appellera forme simpliciale une expression des contraintes égalités sous la forme :

$$\begin{bmatrix} I & H \end{bmatrix} \cdot \begin{bmatrix} x_B \\ x_H \end{bmatrix} = b.$$

On a alors une solution directe :  $\begin{cases} x_B = b \\ x_H = 0 \end{cases}$ .

### B.2.6 Passage d'un sommet à un autre

Dès que l'on quitte un sommet, une des variables hors base correspondantes au moins devient non nulle. Comme on veut se diriger vers un sommet voisin, il ne faut rendre non nulle qu'une

variable, qui va ainsi devenir variable de base pour le prochain sommet. Elle *entre* donc dans la base.

### B.2.6.1 Choix de la variable entrante

Le critère de sélection de la variable entrante dans le critère  $z$  est de prendre la variable hors base qui fournit la plus forte baisse de la fonction objectif. Autrement dit, on choisit celle dont le coût marginal est le plus négatif.

### B.2.6.2 Choix de la variable sortante

En considérant le problème sous forme simpliciale, les contraintes lorsqu'une variable hors base  $x_r$  entre dans la base sont :

$$\begin{aligned} x_1 + a_{1r}x_r &= b_1, \\ &\vdots \\ x_s + a_{sr}x_r &= b_s, \\ &\vdots \\ x_m + a_{mr}x_r &= b_m. \end{aligned}$$

La solution doit être admissible, donc les  $x_i$  sont positifs ou nuls. On va donc *pousser* la variable entrante jusqu'à ce qu'une des variables de base s'annule. C'est cette variable qui est choisie comme variable sortante. Pour les  $i$  tels que  $a_{ir} > 0$ , on choisit le plus petit des  $\frac{b_i}{a_{ir}}$ , noté  $\frac{b_s}{a_{sr}}$ , et on donne à  $x_r$  la valeur  $\frac{b_s}{a_{sr}}$ , alors  $x_s = 0$ . Notons que si tous les  $a_{ir}$  sont inférieurs à zéro, alors la solution du problème est non bornée.

### B.2.6.3 Calcul du nouveau sommet

Le calcul du nouveau sommet consiste à remplacer

- les  $a_{ij}$  par  $a_{ij} - a_{jr} \frac{a_{si}}{a_{sr}}$ ,
- les  $b_j$  par  $b_j - a_{jr} \frac{b_s}{a_{sr}}$ .

Ce calcul est appelé *opération de pivot*.

### B.2.6.4 Test d'optimalité

Le critère d'arrêt est le suivant : La solution de base courante est optimale si, lorsque la fonction est exprimée avec les variables hors base, aucune des variable ne conduit à l'amélioration du critère. Autrement dit, si tous les coûts marginaux sont positifs ou nuls.

# Bibliographie

- [Allgöwer 1999] F. Allgöwer, T. A. Badgwell, S. J. Qin et J. Rawlings and S. Wright. Nonlinear predictive control and moving horizon estimation - and introductory overview. 1999.
- [Amari 2005] S. Amari, I. Demongodin et J.J Loiseau. *Méthode formelle de commande sous contraintes de temps dans les dioïdes*. In Modélisation des Systèmes Réactifs, MSR05, 2005.
- [Aspvall 1980] B. Aspvall et Y. Shiloach. *A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality*,. Siam J. Comput., vol. 9,(N°4), 1980.
- [Baccelli 1992] F. Baccelli, G. Cohen, G.J. Olsder et J.P. Quadrat. Synchronization and linearity. An algebra for Discrete Event Systems. Wiley, New York, 1992.
- [Berthomieu 1991] B. Berthomieu et M. Diaz. *Modeling and verification of time dependent systems using time petri nets*. IEEE Trans. on Software Eng, vol. 17(N°3), pages 259–273, 1991.
- [Blyth 1972] T.-S. Blyth et M.-F. Janowitz. Residuation theory. Pergamon press, 1972.
- [Boimond 1996] J.-L. Boimond et J.-L. Ferrier. *Internal Model Control and Max-Algebra*. IEEE Trans. on Automatic Control, vol. 41(3), pages 457–461, 1996.
- [Boyer 2001] M. Boyer. *Contribution à la modélisation des systèmes à temps contraint et application au multimédia*. Thèse, Université Paul Sabatier, Juillet 2001.
- [Chrétienne 1983] P. Chrétienne. Les réseaux de Petri temporisés. Thèse Université Pierre et Marie Curie (Paris VI), Paris, 1983.
- [Cohen 1989] G. Cohen, P. Moller, S. Gaubert et J. P. Quadrat. *Algebraic Tools for the Performance Evaluation of discret Event System*. IEEE Proceedings : Special issue on Discrete Event Systems, 1989.
- [Cohen 1993] E. Cohen et N. Megiddo. *Maximizing Concave Functions in Fixed Dimension*. In in : Complexity in Numeric Computation, pages 74–87. World Scientific Press, 1993.
- [Cohen 1998] G. Cohen, S. Gaubert et J.-P. Quadrat. *Max-plus algebra and system theory : Where we are and where to go now?* In IFAC Conference on System Structure and Control, Nantes, France, 1998.

- [Cook 1971] S. A. Cook. *The complexity of theorem-proving procedures*. In Proceeding 3rd Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, pages 151–158, 1971.
- [Cormen 2001] T. H. Cormen, C. E. Leiserson, R. L. Rivest et C. Stein. *Introduction to algorithms*. McGraw-Hill Book Company, Cambridge, London, 1993 édition, 2001.
- [Cottenceau 1999] B. Cottenceau. *Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes*. Thèse, LISA - Université d'Angers, 1999.
- [Cottenceau 2001] B. Cottenceau, L. Hardouin, J.-L. Boimond et J.-L. Ferrier. *Model Reference Control for Timed Event Graphs in Dioid*. *Automatica*, vol. 37, pages 1451–1458, August 2001.
- [Cottle 1972] R.W. Cottle et A.F. Veinott. *Polyhedral sets having a least element*. *Mathematical Programming*, vol. 3, pages 238–249, 1972.
- [Cuninghame-Green 1979] R.-A. Cuninghame-Green. *Minimax algebra*. Numeéro 166 de Lecture notes in Economics and Mathematical Systems. Springer, 1979.
- [D.Andersson 2006] D.Andersson et S. Vorobyov. *Fast Algorithms for Monotonic Discounted Linear Programs with Two Variables per Inequality*. In Preprint NI06019-LAA, Isaac Newton Institute for Mathematical Sciences, 2006.
- [Davey 2002] B.A. Davey et H.A. Priestley. *Introduction to lattices and order*. Cambridge University Press, Cambridge, UK, 2002.
- [David 2004] R. David et H. Alla. *Discrete, continuous, and hybrid petri nets*. Springer-Verlag, Berlin, 2004.
- [De Schutter 2000] Bart De Schutter et Ton van den Boom. *Model predictive control for max-plus-linear systems*. In In Proceedings of American Control Conference, Chicago, Illinois, 2000.
- [Declerck 2007] P. Declerck, A. Guezzi et J.-L. Boimond. *Cycle Time of P-time Event Graphs*. In ICINCO 2007, Angers, France, Mai 2007.
- [Declerck 2009] P. Declerck et A. Guezzi. *Trajectory Tracking Control of a Timed Event Graph with Specifications Defined by a P-time Event Graph*. In POSTA 09, Valencia, Spain, Septembre 2009.
- [Declerck 2010] P. Declerck et M.K. Didi-Alaoui. *Optimal control synthesis of timed event graphs with interval model specifications*. *IEEE Transactions on Automatic Control*, pages 518–523, 2010.
- [Didi-Alaoui 2005] M.K. Didi-Alaoui. *Étude et supervision des graphes d'événements temporisés et temporels : vivacité, estimation et commande*. Thèse, ISTIA, Université d'Angers, Décembre 2005.

- [Edmonds 1962] J. Edmonds. *Covers and packings in a family of sets*. Bull. American Mathematical Society, vol. 68, pages 494–499, 1962.
- [Farkas 1894] J. Farkas. *On the applications of the mechanical principle of Fourier*. Matematikai és természettudományi Értesítő, vol. 12, pages 457–472, 1894.
- [Farkas 1898] J. Farkas. *Paraméteres módszer Fourier mechanikai elvéhez*. Matematikai és Fizikai Lapok, vol. 7, pages 63–71, 1898.
- [Fossen 2003] T. I. Fossen, M. Breivik et R. Skjetne. *Line-of-sight path following of underactuated marine craft*. In IFAC, Girona, Spain, 2003.
- [Fourier 1827] J.B.J. Fourier. *Analyse des travaux de l'Académie Royale des Sciences pendant l'année 1824*. Histoire de l'Académie Royale des Sciences de l'Institut de France, vol. 7, 1827.
- [Frezza 1999] R. Frezza. *Path following for air vehicles in coordinated flight*. In IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, Atlanta, GA, USA, 1999.
- [Gabbay 1976] H. Gabbay. *A note on polyhedral sets having a least element*. Mathematical Programming, vol. 11, pages 94–96, 1976.
- [Garey 1979] M.R. Garey et D.S. Johnson. *Computers and Intractability : A guide to the theory of NP-Completeness*. 1979.
- [Gaubert 1992] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes*. Thèse, École des Mines de Paris, July 1992.
- [Gaubert 1995] S. Gaubert. *Resource Optimization and (min,+) Spectral Theory*. IEEE Trans. on Automatic Control, vol. 40, pages 1931–1934, 1995.
- [Giua 2000] A. Giua, A. Piccaluga et C. Seatzu. *Optimal Token Allocation in Timed Cyclic Event-Graphs*. In Proceedings of 5th international work on discrete event systems, pages 209–218, Ghent, Belgium, 2000.
- [Gondran 1979] M. Gondran et M. Minoux. *Graphes et algorithmes*. Eyrolles, Paris, France, 1979. Engl. transl. *Graphs and Algorithms*, Wiley, 1984.
- [Gondran 2001] M. Gondran et M. Minoux. *Graphes, dioïdes et semi-anneaux*. Tec & Doc, Paris, France, 2001.
- [Gros 2000] C. Gros. *Rapport DEA en Biotechnologies et Industries Alimentaires de l'École Nationale Supérieure d'Agronomie et des Industries Alimentaires*. PhD thesis, ENSAIA-INPL, 2000.
- [Guezzi 2009] A. Guezzi, P. Declerck et J.-L. Boimond. *Commande de graphes d'évènements temporisés sur un horizon glissant*. In MSR 2009, Nantes, France, Novembre 2009.
- [Hochbaum 1993] Dorit S. Hochbaum, Nimrod Megiddo, Joseph (Seffi) Naor et Arie Tamir. *Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality*. Mathematical Programming, vol. 62, pages 69–83, 1993.

- [Houssin 2007] L. Houssin, S. Lahaye et J.-L. Boimond. *Just in Time Control of constrained (max,+)-Linear Systems*. Discrete Event Dynamic Systems, vol. vol. 17, n° 2, pages 159–178, 2007.
- [J. Campos 1992] J.M. Colom J. Campos G. Chiola et M. Silva. *Properties and Performance Bounds for Timed Marked Graphs*. IEEE Trans. on Circuits and Systems, vol. 39, May 1992.
- [Karmarkar 1984] N. Karmarkar. *A new polynomial-time algorithm for linear programming*. Combinatorica, vol. 4, no. 4, pages 373–395, December 1984.
- [Karp 1978] Richard M. Karp. *A characterization of the minimum cycle mean in a digraph*. Discrete Mathematics, no. 23, pages 309–311, 1978.
- [Kerivin 2000] Hervé Kerivin. *Réseaux faibles et polyèdres*. Thèse, Université Blaise Pascal - Clermont-Ferrand II, Novembre 2000.
- [Khansa 1996] W. Khansa, J. P. Denat et S. Collart-Dutilleul. *P-time petri nets for manufacturing systems*. In proceedings of WODES'96, pages 94–102, Edimburgh, UK, 1996.
- [Khansa 1997] W. Khansa. *Réseaux de Petri p-temporels : contribution à l'étude des systèmes à événements discrets*. Thèse, Université de Savoie, Annecy, Mars 1997.
- [Kuhn 1956] H. W. Kuhn. *Solvability and Consistency for Linear Equations and Inequalities*. The American Mathematical Monthly, no. 63, pages 217–232, 1956.
- [Kumar 1995] R. Kumar et V.K. Garg. Modeling and control of logical discrete event systems. Kluwer Academic Publishers, USA, 1995.
- [Lahaye 2000] S. Lahaye. *Contribution à l'étude des systèmes linéaires non stationnaires dans l'algèbre des dioïdes*. Thèse, LISA - Université d'Angers, 2000.
- [Lahaye 2004] S. Lahaye, B. Cottenceau et A. Correia. *Commande de Graphes d'Événements Temporisés avec Contrainte de Temps Critique*. In CIFA'2004, Douz, Tunisie, Novembre 2004. Proceedings sur CD-ROM.
- [Lhommeau 2003] M. Lhommeau. *Étude 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*. Thèse, LISA - Université d'Angers, 2003.
- [Magott 1984] J. Magott. *Performance Evaluation of Concurrent Systems using Petri Nets*. In Information Processing Letters 18, pages 7–13, North-Holland, 1984.
- [Megiddo 1983] N. Megiddo. *Towards a genuinely polynomial algorithm for linear programming*. SIAM J. Comput., vol. 12, pages 347–353, 1983.
- [Menguy 000a] E. Menguy, J.-L. Boimond, L. Hardouin et J.-L. Ferrier. *A First Step Towards Adaptive Control for linear in Max Algebra*. Journal of Discrete Event Dynamic Systems, vol. 10(4), pages 347–367, 2000a.

- [Menguy 000b] E. Menguy, J.-L. Boimond, L. Hardouin et J.-L. Ferrier. *Just in Time Control of Timed Event Graphs : Update of Reference Input, Presence of Uncontrollable Input*. IEEE Trans. on Automatic Control, vol. 45, no. 11, pages 2155–2158, November 2000b.
- [Menguy 1997] E. Menguy. *Contribution à la commande des systèmes linéaires dans les dioïdes*. Thèse, LISA - Université d'Angers, 1997.
- [Merlin 1974] P. Merlin. *A study of the Recoverability of Computer System*. Phd thesis, Dep. Comput. Sci., Univ. California, Irvine, 1974.
- [Minkowski 1896] H. Minkowski. *Geometrie des Zahlen (Erste Lieferung)*. Teubner, Leipzig, 1896.
- [Motzkin 1936] T. S. Motzkin. *Beiträge zur Theorie des linearen Ungleichungen*. 1936.
- [Murata 1989] T. Murata. *Petri Nets : Properties, Analysis and Applications*. IEEE Proceedings : Special issue on Discrete Event Systems, vol. 77, no. 4, pages 541–581, 1989.
- [Olms 1970] G.H. Olms. *Analyse des travaux de l'Académie Royale des sciences, pendant l'année 1823, Partie mathématique*. Gauthiervillars, Paris, 1890. pages 339–347, 1970.
- [Petri 1962] C. A. Petri. *Kommunikation mit Automaten*. Thèse, Institut für instrumentelle Mathematik, Bonn, Germany, 1962.
- [Pratt 1977] V. Pratt. Two easy theories whose combination is hard. Technical report, Institute of Technology, Massachusetts, 1977.
- [Propoi 1963] A. Propoi. *Using of linear programming methods for synthetizing sampled data automatic systems*. Automatic Remote Control, vol. 24, pages 837–844, 1963.
- [Proth 1995] J. M. Proth et X. Xie. Les réseaux de petri pour la conception et la gestion des systèmes de production. Masson, Paris, 1995.
- [R. Jeantet 2006] P. Schuk et G. Brulé R. Jeantet T. Croguennec. *Sciences des aliments.T2-Technologie des produits alimentaires*. pages 137–190, 2006.
- [Rafal 1968] M. Rafal et W. Stevens. *Discrete dynamic optimization applied to on-line optimal control*. AIChE Journal, vol. 14, pages 85–91, 1968.
- [Ramamoorthy 1980] C.V. Ramamoorthy et S. Gary. *Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets*. IEEE Trans. on Software Engineering, vol. 6, sep 1980.
- [Richalet J. 1978] Testud J.L. et Papon J. Richalet J. Rault A. *Model predictive heuristic control : application to industrial processes*. Automatica, vol. 14, pages 413–428, 1978.
- [Schrijver 1987] A. Schrijver. *Theory of linear and integer programming*. John Wiley and Sons, 1987.
- [Shostak 1981] A. Shostak. *Deciding linear inequalities by computing loop residues*. J. Assoc. Comput, pages 769–779, 1981.

- [Sifakis 1978] J. Sifakis. *Structural properties of petri nets*. In Mathematical foundation of computer science, volume 64. J. Windowski (ed), Springer Verlag, 1978.
- [Tj. C. Koopmans 1951] ed. Tj. C. Koopmans. *Maximization of a linear function of variables subject to linear inequalities, in : Activity Analysis of Production and allocation*. pages 339–347, 1951.
- [Veinott 1968] A. Veinott. *Extreme Points of Leontief Substitution Systems*. Linear Algebra and its Applications, vol. 1, pages 181–194, 1968.
- [Verbruggen 1984] P. Bruijn H. Verbruggen. *Model algorithmic control using impulse response models*. A, vol. 25, 1984.
- [Wang 1972] S. Wang et C. Desoer. *The exact model matching of linear multivariable systems*. IEEE Trans. on Automatic Control, vol. 17(3), pages 347–349, 1972.
- [Weyl 1935] H. Weyl. *Elementare theorie der konvexen polyeder*. Commentarii Mathematici Helvetici, vol. 7, pages 290–306, 1935.
- [Yamada 1994] T. Yamada et S. Kataoka. *On some LP problems for performance evaluation of timed marked graphs*. IEEE Transactions on Automatic Control, pages 696–698, 1994.
- [Zhou 1999] M. Zhou et K. Venkatesh. Modeling, simulation, and control of flexible manufacturing systems : A Petri net approach (series in intelligent control and intelligent automation, vol. 6). World Scientific Publishing Co. Pte. Ltd., London, 1999.

## Modélisation, analyse de performance et commande des systèmes à événements discrets

**Résumé** : Ce mémoire porte sur la modélisation et l'analyse de réseaux de Petri de type Graphes d'Événements (GE) temporisés et temporels, au moyen d'outils algébriques utilisée dans l'algèbre conventionnelle. La modélisation mathématique de ces systèmes dynamiques à événements discrets (SDED) conduit à une écriture polyédrale de la forme  $A.x \leq b$ , où  $x$  est un vecteur de dates. Nous donnons une technique algébrique permettant d'exprimer les trajectoires au plus tôt et réalisons une synthèse de la commande sous le critère classique de juste-à-temps d'un GE temporisé. On utilise les concepts d'ordre composante par composante, de demi-treillis et d'inégalités monotones. Nous analysons la performance d'un graphe d'événements p-temporels, cette analyse se réduit à un problème de la programmation linéaire dont l'objectif est de calculer la valeur maximale et minimale du temps de cycle d'un graphe d'événements P-temporels. Dans une autre partie, nous constituons un modèle entrées/sorties dont le fonctionnement est proche de celui de l'équation d'état de l'automatique classique. Ensuite, en appliquant une formulation de la programmation linéaire, on calcule la trajectoire au plus tôt et au plus tard en utilisant une fonction objectif. Enfin, nous considérons le problème de la poursuite de trajectoire sur un horizon glissant.

**Mots-clés** : Systèmes dynamiques à événements discrets, graphes d'événements temporisés et temporels, inégalités monotones, dioïdes, temps de cycle, programmation linéaire, commande.

## Modeling, performance analysis and control of discret event systems

**Abstract** : This report concerns the modeling and the analysis of timed and time Event Graphs (EG), by means of algebraic tools used in conventional algebra. The mathematical modeling of these Discrete Event Dynamic Systems (DEDS) is to transform the dater inequalities to a complete system of linear inequalities. It leads to a polyhedral writing of the form  $Ax \leq b$ . An algebraic technique for expressing the earliest trajectories and synthesis of control under the criterion of just-in-time to a timed event graphs uses the concepts of order components, lattice and monotone inequality. We analyze the performance of p-timed event graphs, the general form obtained becomes a problem of linear programming which aims to calculate the maximum and minimum cycle time of a P-timed event graphs. In another part, we set up model inputs/outputs is close to the equation of state of classic automatic. Then, applying a formulation of linear programming, the earliest and latest trajectories are calculated using an objective function. Finally, we consider the problem of tracking trajectory on receding horizon.

**Key-words** : Discrete event dynamic systems, time and timed event graphs, monotone inequality, dioids, cycle time, linear programming, control.