HAL open science

Commande en juste-à-temps sous contraintes de systèmes (max,+)-linéaires

Laurent Houssin, Sébastien Lahaye, Jean-Louis Boimond

To cite this version:

Laurent Houssin, Sébastien Lahaye, Jean-Louis Boimond. Commande en juste-à-temps sous contraintes de systèmes (max,+)-linéaires. Modélisation des Systèmes Réactifs, Oct 2005, France. pp.335-350. hal-00367283

HAL Id: hal-00367283

https://hal.science/hal-00367283v1

Submitted on 10 Mar 2009

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.

Commande en juste-à-temps sous contraintes de systèmes $(max, +)$-linéaires

Laurent Houssin — Sébastien Lahaye — Jean-Louis Boimond

Laboratoire d'Ingénierie des Systèmes Automatisés
Université d'Angers - ISTIA
62 avenue Notre Dame du Lac
49000 Angers

{Houssin,Lahaye,Boimond}@istia.univ-angers.fr

RÉSUMÉ. *Ce papier présente une commande optimale vis-à-vis du critère juste-à-temps de systèmes à événements discrets mettant en jeu des phénomènes de synchronisation. Au regard des études existantes sur le sujet, on généralise le problème en considérant de nouvelles contraintes dans la spécification de l'objectif à atteindre. Pour résoudre le problème ainsi étendu, on le formule comme un problème de recherche d'un plus grand point fixe d'une application définie sur un dioïde. On présente de plus une application à la génération des tables d'horaires des réseaux de bus urbains.*

ABSTRACT. *In this paper, we present an optimal control regarding just-in-time criterion of discrete event systems involving synchronization phenomena. Compared to existing studies about this topic, we generalize the problem by considering additional constraints in the control objective. To solve this problem, we formulate it as a greatest fixed point research of an application defined over a dioid. We show the utility of this control to compute timetables in a urban bus network.*

MOTS-CLÉS : *algèbre des dioïdes, commande en juste-à-temps, systèmes de transport.*

KEYWORDS: *dioids algebra, just-in time control, transportation systems.*

1. Introduction

L'étude des *Systèmes à Evénements Discrets* (SED) fait l'objet de nombreuses recherches. Cette théorie est utilisée pour modéliser des phénomènes de synchronisation ou de concurrence, elle trouve ses applications dans des systèmes de conception humaine comme les systèmes de production, les réseaux informatiques, les réseaux de transport.

Le fonctionnement des SED qui mettent en jeu des phénomènes de synchronisation ne peut pas être décrit dans l'algèbre usuelle par des équations linéaires à cause de la non linéarité des opérateurs (le max par exemple). Il est cependant modélisable par des équations linéaires dans une algèbre particulière : l'algèbre des dioïdes [BAC 92]. La théorie des dioïdes présente de fortes analogies avec la théorie conventionnelle des systèmes linéaires.

On s'intéresse à la commande en juste-à-temps des SED pouvant être représentés comme des systèmes $(max, +)$-linéaires. Les premiers travaux sur ce sujet remontent à [COH 89] et ont été étendus notamment dans [MEN 00] et [LAH 99]. Dans toutes ces études, un seul type de contrainte a été considéré pour spécifier l'objectif à atteindre. On cherche exclusivement à poursuivre une trajectoire de sortie désirée, spécifiée par une consigne. Des résultats issus de la théorie de la résiduation fournissent une solution optimale à ce problème : on synthétise une trajectoire d'entrée donnant les dates d'occurrences les plus tardives pour les événements d'entrée telles que les événements en sortie aient lieu au plus tard avant les dates spécifiées par la consigne.

Dans ce papier, on généralise le problème en considérant des contraintes additionnelles pour la spécification de l'objectif à atteindre. Plus précisément, on peut considérer n'importe quelle contrainte s'exprimant sous la forme d'une inéquation implicite en l'état du système. Cela permet notamment de spécifier : qu'un nombre souhaité d'occurrences d'un événement aient lieu sur un intervalle de dates donné, qu'un écart temporel minimum et/ou maximum sépare deux occurrences d'un même événement, ou encore qu'une contrainte de temps critique existe pour une activité au sein du système. Le problème est formulé comme la recherche d'une solution extrêmale à une inéquation au point fixe. Une méthode itérative est proposée pour résoudre ce dernier problème. Enfin, on propose une application de ces résultats à l'étude des systèmes de transport. On met en avant que le problème de synthèse des tables d'horaires d'un réseau de bus urbains peut se ramener au problème de commande en juste-à-temps considéré. A partir de résultats de la théorie de la résiduation, une solution a déjà été proposée dans [HOU 04] pour des tables d'horaires seulement soumises à deux contraintes particulières. La méthode décrite dans ce papier est différente et permet de prendre en compte des contraintes additionnelles.

Ce papier est organisé de la façon suivante : la section 2 est consacrée à quelques rappels sur la théorie des dioïdes ainsi que sur la modélisation des SED. Dans le §3, nous présentons la mise en oeuvre de la commande optimale des SED. Un exemple sur les systèmes de transport est décrit dans le §4.

2. Préliminaires

2.1. *Dioïdes et structures ordonnées*

Un dioïde $(\mathcal{D}, \oplus, \otimes)$ est un semi-anneau pour lequel la loi additive est idempotente ($a \oplus a = a$). Les éléments neutres des lois additive et multiplicative sont notés respectivement e et ε. Un dioïde est dit complet s'il est fermé pour les sommes infinies et si la loi $\otimes$ distribue sur les sommes infinies. La borne supérieure d'un dioïde complet $\mathcal{D}$ existe et sera notée "$\top$".

Exemple 1 *$\mathbb{Z}_{max}$ est le dioïde ($\mathbb{Z} \cup \{-\infty\}, max, +$) muni du max (loi additive $\oplus$) et de l'addition usuelle + (loi multiplicative $\otimes$). Pour rendre ce dioïde complet, il suffit d'y ajouter $\top = +\infty$, il est alors noté $\overline{\mathbb{Z}}_{max}$. L'ensemble $\overline{\mathbb{Z}}_{min} = (\mathbb{Z} \cup \{-\infty\} \cup \{+\infty\}, min, +)$ est aussi un dioïde complet pour lequel $\varepsilon = +\infty$ et $\top = -\infty$.*

L'idempotence de la loi additive $\oplus$ permet de définir naturellement une relation d'ordre par $a \succeq b \Leftrightarrow a = a \oplus b$. La notation $a \prec b$ signifie $a \preceq b$ et $a \neq b$. Sur un dioïde complet $\mathcal{D}$, il est possible de définir la borne inférieure de toute paire d'éléments (a, b) par $a \wedge b = \bigoplus_{\{x | x \preceq a, x \preceq b\}} x$. L'opération $\wedge$ est associative, commutative, idempotente et possède également un élément neutre T ($T \wedge a = a, \forall a$).

2.2. *Théorie de la résiduation*

Une application f définie d'un dioïde complet $\mathcal{D}$ vers un dioïde complet $\mathcal{C}$ est isotone si $\forall a, b \in \mathcal{D},\ a \preceq b \Rightarrow f(a) \preceq f(b)$. De plus, f est dite semi-continue inférieurement (s.c.i. en abrégé), si, pour tout sous ensemble $\mathcal{B} \subseteq \mathcal{D}$, on a $f(\bigoplus_{x \in \mathcal{B}}) = \bigoplus_{x \in \mathcal{B}} f(x)$.

La théorie de la résiduation [BLY 72] permet de définir des "pseudo-inverses" pour des applications isotones définies sur des ensembles ordonnés (comme les dioïdes par exemple). Plus précisément, la théorie de la résiduation permet d'établir, lorsqu'elle existe, la plus grande solution de l'inéquation $f(x) \preceq b$ avec $b \in \mathcal{C}$. Cet élément lorsqu'il existe est noté $f^\sharp(b)$ et $f^\sharp$ est appelée application résiduée de f.

Théorème 1 *[BAC 92, p174-175] Soit f une application isotone d'un dioïde complet $\mathcal{D}$ vers un dioïde complet $\mathcal{C}$. On a l'équivalence :*

$$f \text{ est résiduable} \Leftrightarrow f \text{ est s.c.i. et } f(\varepsilon_{\mathcal{D}}) = \varepsilon_{\mathcal{C}}.$$

Théorème 2 *[BAC 92, Th. 4.56] Si $f : \mathcal{D} \to \mathcal{C}$ et $g : \mathcal{B} \to \mathcal{D}$ sont deux applications résiduables, alors $f \circ g : \mathcal{B} \to \mathcal{C}$ est également résiduable et $(f \circ g)^\sharp = g^\sharp \circ f^\sharp$.*

2.3. *Plus grands points fixes d'applications définies sur des dioïdes complets*

Dans cette section, on s'intéresse aux bornes supérieures de l'ensemble des points fixes $\mathcal{F}_f = \{x|f(x) = x\}$ et de l'ensemble des points post-fixes $\mathcal{P}_f = \{x|f(x) \succeq x\}$ d'une application f isotone définie sur un dioïde complet $\mathcal{D}$. On rappelle que $\mathcal{P}_f$ a une structure de treillis complet ([BAC 92, Théorème 4.72]). Le théorème de Tarski [TAR 55] montre qu'une application isotone définie sur un treillis complet admet au moins un point fixe. De plus, il est possible de montrer que le plus grand point post-fixe de l'application est confondu avec le plus grand point fixe [DAV 90, théorème 4.11]. La spécification sur un dioïde complet de ce dernier théorème nous donne : $Sup\, \mathcal{P}_f = Sup\, \mathcal{F}_f$ et $Sup\, \mathcal{F}_f \in \mathcal{F}_f$.

Dans la proposition suivante, on spécifie aux dioïdes une méthode classique pour le calcul du plus grand point fixe d'une application isotone.

Proposition 1 *Si la suite y_{k+1} définie par*

$$\begin{array}{rcl} y_0 & = & \top \\ y_{k+1} & = & f(y_k) \end{array} \tag{1}$$

converge en un nombre fini k_e d'itérations, alors y_{k_e} est le plus grand point fixe de f.

2.4. *Modélisation des SED dans les dioïdes*

L'algèbre des dioïdes permet de modéliser des SED qui mettent en jeu (uniquement) des phénomènes de synchronisation et de délai. Le comportement de ces systèmes peut être représenté par des fonctions discrètes appelées *dateurs*. Plus précisément, à un événement noté x, on associe une variable discrète $x(\cdot)$ qui représente ses dates d'occurrence. Par convention, les occurrences d'événement sont numérotées à partir de 0 : $x(0)$ correspond à la date d'apparition du premier événement x. Ces variables sont prolongées vers les valeurs négatives par : $x(k) = -\infty = \varepsilon$ pour $k < 0$ de telle sorte qu'elles sont manipulées comme des applications de $\mathbb{Z}$ vers $\overline{\mathbb{Z}}_{max}$.

Il est possible de modéliser ces systèmes par une représentation d'état

$$\begin{array}{rcl} x(k) & = & Ax(k-1) \oplus Bu(k), \\ y(k) & = & Cx(k), \end{array} \tag{2}$$

où x, u et y sont les vecteurs d'état, d'entrée et de sortie.

L'état initial d'un système est défini par un vecteur $v(k)$ qui s'ajoute à l'équation d'état :

$$x(k) = Ax(k-1) \oplus Bu(k) \oplus v(k).$$

Plus précisément, $v_i(k)$ pour $0 \leq k < k_{d_i}$ renvoie les dates d'occurrence au plus tôt des événements initiaux. L'indice k_{d_i} est celui de la première occurrence de x_i qui est engendrée par les entrées du système (cette définition des conditions initiales est plus largement détaillée dans la référence [BAC 92, §5.4.4.1]). La notion de nombre caractéristique introduite dans [BOI 96] permet de déterminer cet indice.

Définition 1 (nombre caractéristique) *Soit $[A]_i$ la i-ième ligne de la matrice A, le nombre caractéristique associé à la variable d'état x_i du modèle décrit par (2), s'il existe, est le plus petit entier, noté k_{d_i}, tel que $[A^{k_{d_i}}]_i B \neq \varepsilon$.*

Pour pouvoir être manipulée comme un dateur, chaque variable v_i est prolongée comme suit : $v_i(k) = \varepsilon$ pour $k < 0$ et $v_i(k) = v_i(k_{d_i} - 1)$ pour $k \geq k_{d_i}$. On dit que les conditions initiales sont canoniques si $\forall k \in \mathbb{Z},\ v(k) = \varepsilon$ et on se ramène alors à l'équation d'état (2).

Le nombre caractéristique correspond au décalage événementiel entre les entrées du systèmes et un état x_i. Il est également possible de calculer le décalage événementiel entre un état x_i et une sortie y_j. On le définit, s'il existe, comme le plus petit entier $k_{f_{ji}}$ tel que $C_j[A^{k_{f_{ji}}}]^i \neq \varepsilon$ (la notation $[A]^i$ désigne la $i-$ème colonne de A).

Tout comme il existe la transformée en z pour les systèmes échantillonnés, les SED qui mettent en jeu des phénomènes de synchronisation disposent de la transformée en γ et en δ. Cette transformation permet de manipuler des séries formelles, en deux variables commutatives γ et δ, représentant les trajectoires des dateurs. L'ensemble de ces séries formelles est un dioïde noté $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ (ce dioïde est plus largement décrit dans [BAC 92] et [COH 89]). Plus formellement, le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ correspond au dioïde des séries formelles en γ et δ (à coefficients booléens et exposants dans $\mathbb{Z}$) quotienté par la relation d'équivalence suivante : $x \equiv y \ \Leftrightarrow\ \gamma^*(\delta^{-1})^* x = \gamma^*(\delta^{-1})^* y$ avec $\gamma^* = \bigoplus_{i \in \mathbb{N}} \gamma^i$ (ce quotientage permet de prendre en compte la monotonie des dateurs). On notera x l'élément correspondant au dateur $\{x(k)\}_{k \in \mathbb{Z}}$ dans $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. Le support d'une série x est définie par $Supp(x) = \{k \in \mathbb{Z} | x(k) \neq \varepsilon\}$. Vis-à-vis du comportement dynamique du système, on peut interpréter γ comme un opérateur de retard événementiel et δ comme un opérateur de décalage temporel.

La représentation d'état (2) devient :

$$\begin{array}{rcl} x & = & Ax \oplus Bu, \\ y & = & Cx, \end{array} \tag{3}$$

où les coefficients des matrices A, B et C sont des éléments de $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. En considérant le fonctionnement au plus tôt du système, on choisit la plus petite solution de la première équation de (3) qui est donnée par $x = A^* Bu$ avec $A^* = \bigoplus_{i \in \mathbb{N}} A^i$ ([BAC 92, Th 4.75]). La matrice de transfert H du système apparaît alors naturellement : $H = CA^*B$.

2.5. *Étude de quelques applications définies dans les dioïdes*

L'application val

La fonction de valuation des séries formelles du dioïde $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ ([BAC 92, définition 5.19]) est définie comme la borne inférieure de leur support.

$$\begin{array}{rcccl} val & : & \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!] & \longrightarrow & \overline{\mathbb{Z}}_{min} \\ & & x & \longmapsto & val(x) = Min(Supp(x)) \end{array}$$

On a par exemple : $val(\gamma^3\delta^1 \oplus \gamma^5\delta^2) = 3$.

L'application val satisfait ([BAC 92, lemme 4.93]) :

$\forall \mathcal{X} \subseteq \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!],\ val(\bigoplus_{x\in\mathcal{X}}) = \bigoplus_{x\in\mathcal{X}} val(x)$ et $val(\varepsilon) = \varepsilon$.

On en déduit directement que l'application val est résiduable (*cf.* théorème 1). Expliciter la résiduée de val revient à trouver la plus grande série x qui respecte l'inéquation $val(x) \preceq b$. Cette plus grande série est $val^\sharp(b) = \gamma^b\delta^*$.

Applications Pr_a et Pr_+

Soit Pr_a l'application de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ dans lui même définie par

$Pr_a : x \longmapsto Pr_a(x) = Pr_a(\bigoplus_{n,t\in\mathbb{Z}^2} x(n,t)\gamma^n\delta^t) = \bigoplus_{n,t\in\mathbb{Z}^2} x_a(n,t)\gamma^n\delta^t$,

où $x_a(n,t) = \begin{cases} x(n,t) & \text{si } t \geq a, \\ \varepsilon & \text{sinon.} \end{cases}$

Vis-à-vis d'une série x de $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, l'opération $Pr_a(x)$ consiste à ne conserver que les mônomes du support de x à exposants de δ supérieurs ou égaux à a.

On a par exemple : $Pr_3(\gamma^1\delta^2 \oplus \gamma^3\delta^3 \oplus \gamma^4\delta^5) = \gamma^3\delta^3 \oplus \gamma^4\delta^5$.

On peut vérifier facilement que Pr_a est s.c.i. et satisfait $Pr_a(\varepsilon) = \varepsilon$, elle est donc résiduable (th. 1). On explicite maintenant la résiduée de $Pr_a(x)$.

Définition 2 *On décompose toute série $x \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ en la somme $x = x_{a<} \oplus x_{a\geq}$ où*

- *$x_{a<}$ est la série qui contient les monômes $\gamma^n\delta^t$ de x tels que $t < a$.*
- *$x_{a\geq}$ est la série qui contient les monômes $\gamma^n\delta^t$ de x tels que $t \geq a$.*

Proposition 2 *Soient $x, y \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, la plus grande solution de $Pr_a(x) \preceq y$ est $Pr_a^\sharp(y) = y_{a<} \oplus (\gamma^{-1})^*\delta^{a-1}$.*

Preuve :

$$\begin{array}{rrcccl} & Pr_a(x) & = & Pr_a(x_{a<} \oplus x_{a\geq}) & \preceq & y \\ \Leftrightarrow & & & Pr_a(x_{a<}) \oplus Pr_a(x_{a\geq}) & \preceq & y & \text{car } Pr_a \text{ est s.c.i.} \\ \Leftrightarrow & & & x_{a\geq} & \preceq & y & \text{car } Pr_a(x_{a<}) = \varepsilon \\ & & & & & & \text{et } Pr_a(x_{a\geq}) = x_{a\geq} \end{array}$$

On se convainc alors facilement que la plus grande solution x est donnée par $x_{a\geq} = y_{a\geq}$ et $x_{a<} = (\gamma^{-1})^* \delta^{a-1}$ (plus grande série de $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ dont les exposants en δ sont strictement inférieurs à a). □

Par la suite, on aura recours à une autre projection dite causale, notée Pr_+ et définie par :

$$Pr_+(x) = Pr_+(\bigoplus_{x \in I} x(n_i, t_i)\gamma^{n_i}\delta^{t_i}) = \bigoplus_{x \in I} x_+(n_i, t_i)\gamma^{n_i}\delta^{t_i},$$

où $x_+(n_i, t_i) = \begin{cases} x(n_i, t_i), & \text{si } (n_i, t_i) \geq (0, 0)\,; \\ \varepsilon, & \text{sinon.} \end{cases}$

3. Commande optimale

Cette partie est consacrée à la détermination d'une commande optimale pour les SED représentés comme des systèmes linéaires dans les dioïdes. On s'inspire ici du formalisme adopté dans [BOR 90] pour la commande des systèmes linéaires conventionnels. Le principe de cette commande se résume en trois points :

- Le processus vérifie des conditions initiales et finales données.
- Les variables d'état sont soumises à certaines contraintes.
- La commande est dite "optimale" dans le sens où elle optimise un critère.

3.1. *Conditions terminales*

Les conditions terminales caractérisent à la fois l'état initial, c'est-à-dire l'état à l'instant où l'on commence à agir sur le processus (*via* les entrées), et l'état final, c'est-à-dire l'état à l'issue de l'application d'une commande en entrée.

Comme on a pu le détailler au §2.4, l'état initial d'un SED modélisé dans $\overline{\mathbb{Z}}_{max}$ est donné par les dates d'ocurrence au plus tôt des événements initiaux. On considère ici des conditions initiales canoniques. Pour chaque variable d'état x_i, on a défini le nombre caractéristique (définition 1), c'est-à-dire l'indice k_{d_i} de la première occurrence engendrée par les entrées du système.

Dans le cadre de notre commande, l'état final, que l'on se donne, correspond aux indices des dernières occurrences des événements de sortie y que l'on désire commander. De cet objectif, on peut déduire quelles sont les dernières occurrences des variables d'état qu'il est nécessaire de piloter. On note $k_{f_{y_j}}$ (resp. k_{f_i}) la dernière occurrence de l'événement y_j (resp. x_i) que l'on désire commander. Le calcul de l'indice k_{f_i} fait intervenir le décalage événementiel $k_{f_{ji}}$ entre l'état x_i et chaque sortie y_j du système (voir §2.4). Le dernier événement de x_i à commander pour générer le dernier événement de la sortie y_j est donnée par $k'_{f_{ji}} = k_{f_{y_j}} - k_{f_{ji}}$. On en déduit ensuite que

le dernier événement de x_i à commander est celui qui permet d'assurer les derniers événements de toutes les sorties, il en résulte $k_{f_i} = max(k'_{f_{1i}}, k'_{f_{2i}}, \ldots, k'_{f_{pi}})$ pour un système comportant p sorties.

3.2. *Contraintes*

Le système que l'on désire commander peut être soumis à des contraintes. Celles-ci peuvent être de nature différente, elles permettent de traduire des éléments du comportement dynamique du système et/ou l'objectif à atteindre. On considère ici des contraintes qui peuvent s'exprimer sous la forme d'une inéquation implicite sur l'état. Ces contraintes s'appliquent sur un intervalle d'occurrences pour chacune des variables d'état concernées. En effet, pour une variable d'état x_i elles ne peuvent être appliquées avant l'occurrence k_{d_i}, car avant celle-ci les occurrences de x_i ne sont pas contrôlables *via* les entrées du système. La dernière occurrence de x_i qui nous intéresse est k_{f_i}, au delà, il est inutile de contraindre x_i. Les contraintes ne s'appliquent donc que pour les indices d'occurrences appartenant à l'intervalle $[k_{d_i}, k_{f_i}]$ pour chaque variable d'état x_i. Afin de les coder dans le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$, dans lequel on manipule l'ensemble de la trajectoire d'un dateur $\{x_i(k)\}_{k\in\mathbb{Z}}$ sous la forme de la série x_i, on a recours à des vecteurs ω et ν comme ci-dessous :

$$\begin{array}{lcl} x & \preceq & (g_1(x) \wedge \omega) \oplus \nu \\ x & \preceq & (g_2(x) \wedge \omega) \oplus \nu \\ \vdots & \vdots & \vdots \\ x & \preceq & (g_q(x) \wedge \omega) \oplus \nu, \end{array} \tag{4}$$

avec ω (respectivement ν) un vecteur tel que $\omega_i = \gamma^{k_{d_i}}\delta^*$ (respectivement $\nu_i = \gamma^{k_{f_i}+1}\delta^*$) et chaque g_l, $l = 1, 2, \ldots, q$, est une application de $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^n$ dans $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^n$ (n est la dimension du vecteur d'état) modélisant une contrainte. Les vecteurs ω et ν permettent de relaxer les contraintes pour les indices d'occurrences des événements x_i, $i = 1, 2, \ldots, n$, en dehors de l'intervalle $[k_{d_i}, k_{f_i}]$.

3.3. *Critère*

Un objectif consacré pour la commande des SED est de chercher à retarder au maximum les occurrences des événements d'entrée (c'est-à-dire chercher le plus grand vecteur de commande u) tout en respectant les performances imposées par une spécification (ici les conditions terminales et les contraintes)[1]. Le critère à maximiser pour assurer l'optimalité est donc $J = u$.

1. Ceci correspond à la problématique de commande en juste-à-temps qui de façon générale vise à fournir la "quantité juste" (le besoin) au "temps voulu" (date du besoin).

3.4. *Synthèse*

Dans cette section, on montre que la synthèse de la commande optimale se résume à la recherche d'un plus grand point fixe d'une application. En considérant le fonctionnement au plus tôt du système, on a $x = A^*Bu$ et en utilisant la notation [2] $g'_l(u) = A^*B \backslash ((g_l(A^*Bu) \wedge \omega) \oplus \nu)$, la recherche de la plus grande commande u revient à determiner la plus grande solution du système d'inéquations :

$$\begin{cases} u & \preceq & g'_1(u) \\ u & \preceq & g'_2(u) \\ \vdots & \vdots & \vdots \\ u & \preceq & g'_q(u), \end{cases}$$

ce qui est équivalent à trouver le plus grand u respectant l'inéquation

$$u \preceq g'_1(u) \wedge g'_2(u) \wedge \ldots \wedge g'_q(u) = f(u). \tag{5}$$

L'objectif est de trouver la plus grande solution de (5) (la commande qui maximise le critère).

Proposition 3 *Si l'heuristique suivante converge en un nombre fini k_e d'itérations*

$$\begin{aligned} u_0 &= \top \\ u_{k+1} &= f(u_k), \end{aligned}$$

alors u_{k_e} est la commande qui respecte les conditions terminales ainsi que les contraintes traduites par f (équation (5)) pour laquelle le critère J est optimal.

Preuve :

On remarque que l'ensemble des commandes qui respectent les contraintes est en fait l'ensemble $\mathcal{P}_f$ des points post-fixes de f. La suite définie dans la proposition 1, si elle converge, nous fournit le plus grand élément $Sup\, \mathcal{P}_f$ de cet ensemble.

□

4. Application aux réseaux de transport urbain

On s'intéresse dans cette partie aux systèmes de transport et plus particulièrement aux réseaux de bus urbains. Un modèle de ces systèmes est proposé, celui-ci se présente sous la forme d'une représentation d'état dans les dioïdes.

---

2. $a \backslash b$ désigne la plus grande solution de $ax \preceq b$ (résiduée du produit).

Le comportement de ces systèmes est régi en partie par une table d'horaires. Elle définit, pour chaque arrêt, les dates de départ théoriques des bus. Ces tables sont générées lors d'une optimisation hors ligne du processus réalisée dans des conditions moyennes d'exploitation. Cette optimisation, appelée tableau de marche, se décompose en trois étapes : une étude géographique et sociale, une définition des performances du réseau et la synthèse des table d'horaires ([HOU 04]). Partant du modèle de réseau de bus proposé, on s'attache ensuite à décomposer le problème de génération de tables d'horaires sous la forme de plusieurs contraintes s'appliquant à son état. On résout par la suite ce problème en appliquant la démarche exposée au §3.

4.1. *Modélisation des réseaux de transport urbain*

Un réseau de transport peut être modélisé par une équation d'état dans $\overline{\mathbb{Z}}_{max}$ par :

$$\begin{array}{rcl} x(k) & = & Ax(k-1) \oplus Bu(k), \\ y(k) & = & Cx(k), \end{array} \tag{6}$$

où $x(k)$ est un vecteur tel que $x_i(k)$ correspond à la date du $k+1$-ième départ de bus à l'arrêt i du réseau. La matrice A est définie de la façon suivante : $A_{ij} = a_{ij}$ où a_{ij} désigne le temps de parcours de l'arrêt j à l'arrêt i si j précède i, $A_{ij} = \varepsilon$ sinon. Le vecteur $y(k)$ correspond au vecteur des dateurs associées à des arrêts considérés comme "stratégiques" au sein du réseau (la qualité de service doit plus particulièrement y être respectée). La table d'horaires du système est modélisée par le vecteur d'entrée $u(k)$, ainsi la variable $u_i(k)$ désigne la date de départ prévue du $k+1$-ème bus à l'arrêt i. En pratique, les synchronisations des bus avec ces tables ne se font qu'à certains arrêts particuliers du réseau tels que les terminus et les départs de ligne. Pour les autres arrêts, la table d'horaires n'a qu'une valeur indicative. Il en découle la structure de la matrice B : $B_{ii} = e$ si la table d'horaires est respectée à l'arrêt x_i, $B_{ij} = \varepsilon$ sinon.

4.2. *Présentation du problème de génération de tables d'horaires*

On présente ici le problème de génération de tables d'horaires en le décomposant sous la forme de contraintes sur l'état du modèle proposé précédemment pour les réseaux de bus urbains.

• Dans un premier temps, on spécifie une qualité de service désirée aux arrêts stratégiques du réseau. Cette qualité de service s'exprime sous la forme de dates au plus tard (par exemple le départ du k-ième bus à l'arrêt i doit avoir lieu avant la date t). Si on note $z(k)$ le vecteur représentant ces dates au plus tard pour les arrêts stratégiques, il en découle la contrainte suivante :

$$C \otimes x(k) \preceq z(k) \qquad \text{pour } k_{d_i} \leq k \leq k_{f_i}, \tag{7}$$

où k_{d_i} et k_{f_i} sont les bornes de l'intervalle d'indices de l'événement x_i que l'on souhaite commander (voir §3.1). On considere que $z_j(k) \in \mathbb{N} \cup \{\varepsilon\}$ pour $k \leq k_{f_{y_j}}$.

• Le respect de la contrainte d'écart-temps maximum entre deux bus permet d'assurer une fréquence minimum du passage des bus dans le réseau. Pour un arrêt i, cette contrainte s'exprime par :

$$\begin{cases} x_i(k) \preceq \triangle_i^{max} \otimes x_i(k-1) & \text{pour } k_{d_i} < k \leq k_{f_i}, \\ x_i(k) = x_i(k_{d_i}) & \text{pour } k = k_{d_i}. \end{cases}$$

$$\Longleftrightarrow x_i(k) \preceq \triangle_i^{max} \otimes x_i(k-1) \oplus x_i(k_{d_i}) \qquad \text{pour } k_{d_i} \leq k \leq k_{f_i}. \tag{8}$$

On rappelle que les conditions initiales du système considéré sont canoniques et donc pour $k < k_{d_i}$, on a $x_i(k) = \varepsilon$ (ce qui assure l'équivalence (8)). L'écart maximum est généralement différent d'une ligne à une autre[3].

• D'autre part, le respect d'une contrainte d'écart-temps minimum permet d'éviter les phénomènes d'accumulation. Un tel scénario a tendance à se produire dès qu'un bus prend du retard. En effet, le bus en retard a logiquement plus de passagers à enlever (son retard augmente alors) et le bus qui suit moins de passagers, ce dernier aura donc tendance à rattraper le bus qui le précède. La contrainte d'écart-temps minimum s'écrit pour un arrêt i :

$$x_i(k) \succeq \triangle_i^{min} \otimes x_i(k-1) \qquad \text{pour } k_{d_i} \leq k \leq k_{f_i}. \tag{9}$$

Tout comme la contrainte d'écart-temps maximum, la contrainte d'écart temps minimum est différente suivant les lignes.

• A certaines heures (connues) de la journée des pics de charge apparaissent. Ceux-ci peuvent être dus à l'intermodalité (par exemple l'arrivée d'un train pour des arrêts situés à proximité d'une gare) ou aux activités urbaines (sorties d'école, débauche d'usine...). Dans ce cas, on souhaite qu'un ou plusieurs bus passe(nt) à l'arrêt concerné dans un intervalle de dates données afin d'absorber un tel pic de charge. Pour l'arrêt x_i et pour un intervalle de dates $[t_j, t_j + r]$, on modélise cette contrainte par :

$$\exists k \in [k_{d_i}, k_{f_i}] \text{ tel que } x_i(k) \succeq t_j \text{ et } x_i(k+m) \prec t_j + r, \tag{10}$$

où m est le nombre de bus souhaités à l'arrêt i pendant l'intervalle $[t_j, t_j + r]$ pour absorber le pic.

• On peut souhaiter limiter le temps d'attente des bus à certains arrêts. Il peut s'agir d'un objectif de qualité de service (une attente supérieure à un "seuil" donné peut être perçue négativement par les usagers) ou d'une contrainte matérielle (cas d'un arrêt situé sur une voie commune avec les voitures). On note ϕ_{ji}^{max} la durée du trajet de l'arrêt i à l'arrêt j cumulée avec le temps d'attente maximum souhaité à l'arrêt j. Cette contrainte peut s'écrire :

3. Ces écart-temps sont définis dans la seconde étape du tableau de marche, c'est-à-dire lors du choix des performances du réseau.

$$\begin{cases} x_j(k) \preceq \phi_{ji}^{max} \otimes x_i(k-1) & \text{pour } k_{d_j} < k \leq k_{f_j}, \\ x_j(k) = x_j(k_{d_i}) & \text{pour } k = k_{d_j}, \end{cases}$$
$$\Longleftrightarrow x_j(k) \preceq \phi_{ji}^{max} \otimes x_i(k-1) \oplus x_j(k_{d_j}) \qquad \text{pour } k_{d_j} \leq k \leq k_{f_j}, \tag{11}$$

puisque pour $k < k_{d_i}$, on a $x_i(k) = \varepsilon$.

Dans un SED modélisé par un GET, cette contrainte revient à considérer que le temps de séjour des jetons ne doit pas excéder une durée maximale dans certaines places. On parle alors de "contrainte de temps critique". Dans [LAH 04], ce type de contrainte a été considéré dans le cadre de la synthèse d'une précompensation par modèle de référence.

4.3. *Résolution*

Afin d'appliquer les résultats du §3 à ce système, les contraintes doivent être exprimées sous la forme (4) dans le dioïde de séries formelles $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$.

• La contrainte (7) s'exprime naturellement dans le dioïde $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$: $x \preceq (C \wp z \wedge \omega) \oplus \nu$.

• La transformée en $\gamma\delta$ de la contrainte (8) conduit à $x \preceq ((\gamma \triangle^{max} x \oplus x_d) \wedge \omega) \oplus \nu$, où $\triangle^{max} = \begin{pmatrix} \delta^{\triangle_1^{max}} & \varepsilon & \varepsilon \\ \varepsilon & \delta^{\triangle_2^{max}} & \varepsilon \\ \varepsilon & \varepsilon & \ddots \end{pmatrix}$ et x_d est défini par $x_{d_i} = \gamma^{k_{d_i}} \delta^{x_i(k_{d_i})}$.

• De la même façon, on modélise la contrainte (9) par l'inéquation $x \preceq (\gamma \triangle^{min} \wp x \wedge \omega) \oplus \nu$, où $\triangle^{min}$ a la même structure que $\triangle^{max}$.

• Pour exprimer formellement le fait qu'au moins m événements aient lieu dans un intervalle de date $[t_j, t_j + r]$ (contrainte (10) dans $\overline{\mathbb{Z}}_{max}$), on utilise les applications Pr_a et val définies au §2.5.

On considère que cette contrainte s'applique à la variable x_i. Le numéro du premier événement de la série x_i ayant lieu après $t_j + r$ (c'est-à-dire pour $t \geq t_j + r + 1$) doit être supérieur au numéro du premier événement qui a lieu après $t_j - 1$ (c'est-à-dire pour $t \geq t_j$), ce qui traduit la présence d'au moins un événement entre les dates t_j et $t_j + r$. La mise en équation dans $\overline{\mathbb{Z}}_{min}$ de cette contrainte donne :

$$\begin{array}{rcl} val(Pr_{t_j+r+1}(x_i)) & \prec & val(Pr_{t_j}(x_i)) \\ \Longleftrightarrow \quad val(Pr_{t_j+r+1}(x_i)) & \preceq & 1 \otimes val(Pr_{t_j}(x_i)). \end{array}$$

Pour modéliser la présence de m événements, on utilise l'inéquation

$$val(Pr_{t_j+r+1}(x_i)) \preceq m \otimes val(Pr_{t_j}(x_i)).$$

Les applications val et Pr_a étant résiduables, en utilisant le théorème 2, l'inéquation précédente peut être reformulée comme suit :

$$x_i \preceq Pr_{t_j+r+1}^{\sharp}(val^{\sharp}(m \otimes val(Pr_{t_j}(x_i)))).$$

• La contrainte (11) s'exprime naturellement dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ par : $x \preceq ((\phi^{max}\ x \oplus x_d) \wedge \omega) \oplus \nu$, où la matrice ϕ est définie de la façon suivante :

$$\begin{cases} \phi_{ll}^{max} = e & \text{pour tout } l \neq j \\ \phi_{ji}^{max} = \gamma\delta^{\phi_{ji}^{max}} & \\ \phi_{\alpha\beta}^{max} = \varepsilon & \text{sinon.} \end{cases}$$

• Afin qu'elles puissent être utilisées comme des tables d'horaires, les séries calculées ne doivent pas comporter de dates négatives. Pour spécifier cette contrainte, on utilise la projection causale définie au §2.5. L'inéquation $u \preceq Pr^+(u)$ permet de ne conserver que la partie causale de u. Pour contraindre u sur l'intervalle d'occurrences désirés, on utilise : $u \preceq (Pr^+(u) \wedge (A^*B \backslash\!\!\!\!\backslash \omega)) \oplus (A^*B \backslash\!\!\!\!\backslash \nu)$. La forme particulière de cette contrainte provient du fait qu'elle s'applique directement sur le vecteur d'entrée.

Les six contraintes du système ont été modélisées sous la forme (4). En utilisant les notations du §3.4, ce système d'inéquations est équivalent à l'inéquation suivante :

$$u \preceq g_1'(u) \wedge g_2'(u) \wedge \ldots \wedge g_6'(u) = f(u). \tag{12}$$

Proposition 4 *Appliqué à l'inéquation (12), l'algorithme de la proposition 3 converge en un nombre fini k_e d'itérations. Le vecteur u_{k_e} est alors la table d'horaire optimale pour les arrêts stratégiques.*

Preuve : La preuve de la proposition 3 permet d'affirmer que u_{k_e} est la commande optimale. On s'efforce maintenant de montrer que l'algorithme converge en un nombre fini d'itérations.

On remarque tout d'abord qu'en considérant seulement la première contrainte, on a :

$$f(u_k) \preceq A^*B \backslash\!\!\!\!\backslash ((C \backslash\!\!\!\!\backslash z \wedge \omega) \oplus \nu) \qquad \text{pour tout } k \geq 0 \tag{13}$$

On observe d'autre part, que l'on a $g_l'(u_k) = A^*B \backslash\!\!\!\!\backslash ((g_l(A^*Bu_k) \wedge \omega) \oplus \nu) \succeq A^*B \backslash\!\!\!\!\backslash ((\varepsilon \wedge \omega) \oplus \nu) = A^*B \backslash\!\!\!\!\backslash \nu$ pour $l \in \{1, \ldots, 5\}$ et que pour $l = 6$, on a $g_6'(u_k) = (Pr^+(u_k) \wedge (A^*B \backslash\!\!\!\!\backslash \omega)) \oplus (A^*B \backslash\!\!\!\!\backslash \nu) \succeq A^*B \backslash\!\!\!\!\backslash \nu$. On en déduit alors :

$$f(u_k) \succeq A^*B \backslash\!\!\!\!\backslash \nu \qquad \text{pour tout } k \geq 0 \tag{14}$$

On s'interesse maintenant à la structure des vecteurs qui encadrent $f(u_k)$ dans (13) et (14). L'élément i du vecteur $A^*B \backslash\!\!\!\!\backslash ((C \backslash\!\!\!\!\backslash z \wedge \omega) \oplus \nu)$ est de la forme suivante : $(A^*B \backslash\!\!\!\!\backslash ((C \backslash\!\!\!\!\backslash z \wedge \omega) \wedge \nu))_i = \gamma^{n_d}\delta^{t_d} \oplus \ldots \oplus \gamma^{n_f}\delta^{t_f} \oplus \gamma^{n_f+1}\delta^*$. Le premier exposant n_d en γ correspond au numéro du premier événement de x_i controllable par les entrées (donné par ω_i), soit k_{d_i}, moins le décalage événementiel entre la variable d'état x_i et les entrèes du système, c'est-à-dire k_{d_i}. On a donc $n_d = k_{d_i} - k_{d_i} = 0$. De la même façon, n_f correspond à la dernière occurrence de x_i à commander, soit k_{f_i}, moins le décalage événementiel entre la variable d'état x_i et les entrèes du système, c'est-à-dire k_{d_i}. Il en découle $n_f = k_{f_i} - k_{d_i}$. Les exposants $\{t_d, \ldots, t_f\}$ appartienent à $\mathbb{N} \cup \{\varepsilon\}$, puisque $z_j(k) \in \mathbb{N} \cup \{\varepsilon\}$ pour $k \leq k_{f_{y_j}}$.

Les éléments du vecteur $A^*B \backslash\!\!\!\!\backslash \nu$ ont eux la forme $(A^*B \backslash\!\!\!\!\backslash \nu)_i = \gamma^{n_f+1}\delta^*$.

Compte tenu de la forme des vecteurs $A^*B \backslash\!\!\!\!\backslash ((C \backslash\!\!\!\!\backslash z \wedge \omega) \oplus \nu)$ et $A^*B \backslash\!\!\!\!\backslash \nu$, on déduit que tout élément $(u_{k+1})_i$ a la forme suivante $(u_{k+1})_i = \gamma^{n_d}\delta^{t'_d} \oplus \ldots \oplus \gamma^{n_f}\delta^{t'_f} \oplus \gamma^{n_f+1}\delta^*$ où les exposants $\{t'_d, \ldots, t'_f\}$ appartiennent à $\mathbb{N} \cup \{\varepsilon\}$. Etant donnée l'intégrité des exposants, il existe un nombre fini de trajectoires s telle que $\gamma^{n_f+1}\delta^* \preceq s \preceq (u_{k+1})_i$. On observe finalement que la suite est décroissante. En effet, on a $u_1 = f(u_0) \preceq u_0 = \top$. De plus, si on considère $u_n = f(u_{n-1}) \preceq u_{n-1}$ alors on a aussi $u_{n+1} = f(u_n) \preceq f(u_{n-1}) = u_n$ puisque f est isotone. On peut donc conclure que la suite $u_{k+1} = f(u_k)$ stationnarise en un nombre fini d'itérations.

□

Remarque 1 *Aux arrêts pour lesquels $B_{ii} = \varepsilon$ (table d'horaires non respectée), les horaires de passage des bus sont obtenues par simulation du modèle (6) avec u_{k_e}.*

4.4. *Exemple*

On applique la méthode du paragraphe précédent pour calculer la table d'horaires d'un réseau de transport composé de deux lignes ($cf.$ figure 1).

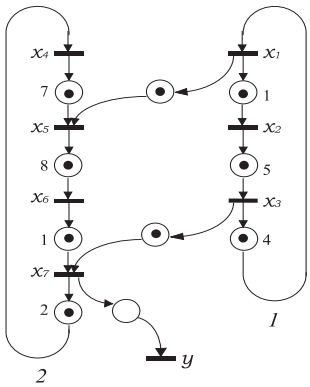

Figure 1. *Représentation en GET d'un réseau de bus comportant deux lignes*

Le système est décrit par le modèle (6) avec

$$A = \begin{pmatrix} \varepsilon & \varepsilon & \gamma^1\delta^4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \gamma^1\delta^1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \gamma^1\delta^5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma^1\delta^2 \\ \gamma\delta^0 & \varepsilon & \varepsilon & \gamma^1\delta^7 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma^1\delta^8 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \gamma\delta^0 & \varepsilon & \varepsilon & \gamma^1\delta^1 & \varepsilon \end{pmatrix},$$

$B = Id$ et $C = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{pmatrix}$.

L'état initial du système nous conduit au vecteur ω suivant : $\forall i,\ \omega_i = \gamma^0\delta^\top$. On désire effectuer 15 départs pour l'arrêt $y = x_7$, on obtient donc le vecteur ν suivant :

$$\nu = \begin{pmatrix} \gamma^{12}\delta^\top & \gamma^{13}\delta^\top & \gamma^{14}\delta^\top & \gamma^{12}\delta^\top & \gamma^{13}\delta^\top & \gamma^{14}\delta^\top & \gamma^{15}\delta^\top \end{pmatrix}^t$$

Le système est soumis aux contraintes suivantes :

▷ le dixième bus à l'arrêt x_7 doit partir avant la date 150 et le quinzième avant la date 200, on a donc $z = \gamma^0\delta^{150} \oplus \gamma^{10}\delta^{200}$,

▷ la contrainte d'écart-temps minimum correspond à $\triangle_{ii}^{min} = \gamma^0\delta^6\ \ \forall i$,

▷ les départs de bus doivent être séparés d'au plus 10 unités de temps, on a donc $\triangle_{ii}^{max} = \gamma^0\delta^{10}\ \ \forall i$,

▷ il doit y avoir un départ de bus entre les dates 130 et 132 à l'arrêt x_4.

▷ Les bus ne peuvent pas stationner plus de 2 unités de temps à l'arrêt x_2. En tenant compte du temps de trajet entre les arrêt x_1 et x_2, la matrice ϕ est définie par :

$$\begin{cases} \phi_{ll} = e & \text{pour tout } l \neq 2 \\ \phi_{21} = \gamma^1\delta^3 & \\ \phi_{\alpha\beta} = \varepsilon & \text{sinon.} \end{cases}$$

Appliquée à ce problème, la méthode itérative de la proposition 3 converge en 9 itérations, les tables d'horaires générées sont présentés ci-dessous :

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
u_1	105	111	117	123	129	135	141	151	161	171	181	191	–	–	-
u_2	102	108	114	120	126	132	138	144	154	164	174	184	194	–	-
u_3	102	108	114	120	126	132	138	144	150	160	170	180	190	200	-
u_4	96	102	108	114	120	126	132	142	152	162	172	182	–	–	-
u_5	99	105	111	117	123	129	135	141	151	161	171	181	191	–	-
u_6	101	107	113	119	125	131	137	143	149	159	169	179	189	199	-
u_7	96	102	108	114	120	126	132	138	144	150	160	170	180	190	200

Cette table d'horaires est la plus grande table qui respecte l'ensemble des contraintes. Afin de se rendre compte de l'importance des contraintes (10) et (11) pour notre exemple, on recommence le calcul de la table d'horaires sans tenir compte de ces dernières. La table ainsi générée se trouve ci-dessous.

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
u_1	105	111	117	123	129	135	141	151	161	171	181	191	–	–	-
u_2	103	109	115	121	127	133	139	145	155	165	175	185	195	–	-
u_3	102	108	114	120	126	132	138	144	150	160	170	180	190	200	-
u_4	98	104	110	116	122	128	134	144	154	164	174	184	–	–	-
u_5	99	105	111	117	123	129	135	141	151	161	171	181	191	–	-
u_6	101	107	113	119	125	131	137	143	149	159	169	179	189	199	-
u_7	96	102	108	114	120	126	132	138	144	150	160	170	180	190	200

On remarque bien l'absence de passage de bus à l'arrêt x_4 entre les dates 130 et 132. On remarque également que les bus à l'arrêt x_2 s'arrêtent pendant 3 unités de temps (il faut tenir compte de l'unité de temps qui correspond au temps de trajet entre l'arrêt x_1 et l'arrêt x_2).

5. Conclusion

Dans ce papier, nous présentons une méthode pour la mise en oeuvre de la commande optimale en juste-à-temps des SED. L'originalité de cette commande réside dans la prise en compte des contraintes sur l'état, la seule condition étant que celles-ci doivent se formuler en une inéquation sur l'état. Elle fournit une solution optimale exacte si le calcul converge. C'est le cas lorsqu'on l'applique à un problème de génération de tables d'horaires de systèmes de transport. Pour cette application, il est envisagé de comparer l'efficacité de la méthode aux techniques existantes. Il serait également intéréssant de mettre à jour des conditions générales de convergence du calcul de la commande.

6. Bibliographie

[BAC 92] BACCELLI F., COHEN G., OLSDER G. J., QUADRAT J. P., *Synchronization and Linearity*, Wiley, 1992.

[BLY 72] BLYTH T. S., JANOWITZ M. F., *Residuation Theory*, Pergamon press, 1972.

[BOI 96] BOIMOND J. L., FERRIER J. L., « Internal Model Control and Max-Algebra : Controller Design », *IEEE Trans. on Automatic Control*, vol. 41, 1996, p. 457-461.

[BOR 90] BORNE P., DAUPHIN-TANGUY G., RICHARD J., ROTELLA F., ZAMBETTAKIS I., *Commande et optimisation des processus*, Technip, 1990.

[COH 89] COHEN G., MOLLER P., QUADRAT J. P., VIOT M., « Algebraic Tools for the Performance Evaluation of Discrete Event Systems », *IEEE Proceedings : Special issue on Discrete Event Systems*, vol. 77, n° 1, 1989.

[DAV 90] DAVEY B., PRIESTLEY H., *Introduction to lattices and order*, Cambrige University press, 1990.

[HOU 04] HOUSSIN L., LAHAYE S., BOIMOND J. L., « Modelling and control of urban bus networks in dioids algebra », *Proceedings of WODES'2004*, Reims, France, 2004.

[LAH 99] LAHAYE S., BOIMOND J. L., HARDOUIN L., « Optimal control of (min,+) linear time-varying systems », *Proceedings of PNPM'99*, Zaragoza, Sept. 1999.

[LAH 04] LAHAYE S., COTTENCEAU B., CORREÏA A., « Commande de GET avec contrainte de temps critique », *Proceedings of CIFA'2004*, Douz, Tunisie, 2004.

[MEN 00] MENGUY E., BOIMOND J. L., HARDOUIN L., FERRIER J. L., « Just in time control of linear systems in dioid : cases of an update of reference input and uncontrollable input », *IEEE Trans. on Automatic Control*, vol. 45, n° 11, 2000, p. 2155-2159.

[TAR 55] TARSKI A., « A lattice theorical fixed point theorem and its applications », *Pacific Journal of Maths*, , 1955, p. 285-309.