

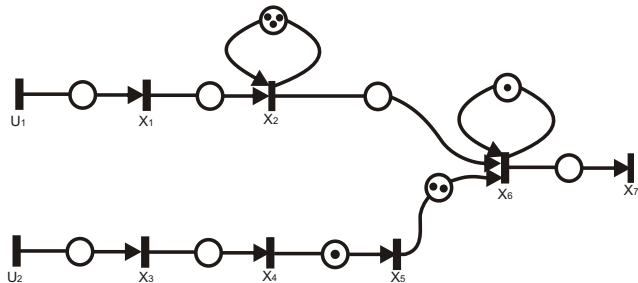
Commande de systèmes $(max, +)$ -linéaires

Laurent Houssin

LAAS

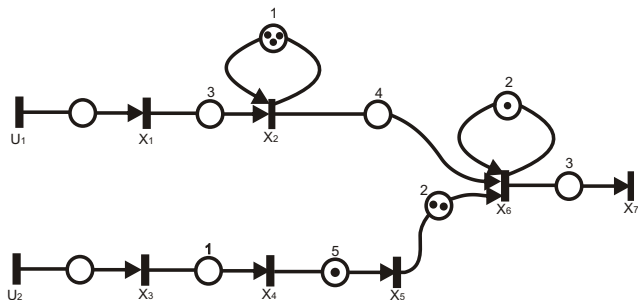
<http://www.laas.fr/laas>

10 janvier 2008



On considère une sous-classe de Réseaux de Petri : Les **Graphes d'Evènements**

Introduction



On considère une sous-classe de Réseaux de Petri : Les **Graphes d'Evènements Temporisés**.

Une structure algébrique est particulièrement adaptée pour décrire le fonctionnement de tels systèmes :

Une structure algébrique est particulièrement adaptée pour décrire le fonctionnement de tels systèmes : *les dioïdes*.

Plan

- 1 Quelques éléments de la théorie des diïdes

Plan

- 1 Quelques éléments de la théorie des diïodes
- 2 Les systèmes $(\max, +)$ -linéaires

Plan

- 1 Quelques éléments de la théorie des diïodes
- 2 Les systèmes $(max, +)$ -linéaires
- 3 Problèmes de commande dans les systèmes $(max, +)$ -linéaires

Plan

- 1 Quelques éléments de la théorie des diïodes
- 2 Les systèmes $(max, +)$ -linéaires
- 3 Problèmes de commande dans les systèmes $(max, +)$ -linéaires
- 4 Ordonnancement et algèbre $(max, +)$

Dioïde

Un dioïde est un semi-anneau idempotent dont les deux lois internes sont notées \oplus et \otimes .

Dioïde

Un dioïde est un semi-anneau idempotent dont les deux lois internes sont notées \oplus et \otimes .

Exemple

Algèbre $(\max, +)$: $\overline{\mathbb{Z}}_{\max} = (\mathbb{Z} \cup \{+\infty, -\infty\}, \max, +)$

Dioïde

Un dioïde est un semi-anneau idempotent dont les deux lois internes sont notées \oplus et \otimes .

Exemple

Algèbre $(max, +)$: $\overline{\mathbb{Z}}_{max} = (\mathbb{Z} \cup \{+\infty, -\infty\}, max, +)$

Structure ordonnée

La loi idempotente \oplus permet de définir une relation d'ordre sur les dioïdes. Un dioïde possède plus précisément une structure de treillis.

Dioïde

Un dioïde est un semi-anneau idempotent dont les deux lois internes sont notées \oplus et \otimes .

Exemple

Algèbre $(max, +)$: $\overline{\mathbb{Z}}_{max} = (\mathbb{Z} \cup \{+\infty, -\infty\}, max, +)$

Structure ordonnée

La loi idempotente \oplus permet de définir une relation d'ordre sur les dioïdes. Un dioïde possède plus précisément une structure de treillis.

Cette structure permet d'utiliser des outils issus de la théorie des treillis :

- *la résiduation*
- *l'étude des points fixes*

Définitions

pour $f : \mathcal{D} \rightarrow \mathcal{C}$

f isotone $a, b \in \mathcal{D}, a \preceq b \Rightarrow f(a) \preceq f(b)$

f antitone $a, b \in \mathcal{D}, a \preceq b \Rightarrow f(a) \succeq f(b)$

Résiduation¹

f isotone est résiduable \Leftrightarrow le plus grand élément de $\{x \in \mathcal{D} \mid f(x) \preceq b\}$ existe et est noté $f^\#(b)$

$f^\#$ est appelée *application résiduée* de f .

¹T. S. Blyth and M. F. Janowitz, *Residuation theory*, 1972, Pergamon press

Résiduation¹

f isotone est résiduable \Leftrightarrow le plus grand élément de $\{x \in \mathcal{D} \mid f(x) \preceq b\}$ existe et est noté $f^\#(b)$

$f^\#$ est appelée *application résiduée* de f .

Exemple

$L_a : x \mapsto a \otimes x$ est résiduable,
la plus grande solution de $L_a(x) \preceq b$ est $L_a^\#(b) = a \oslash b$.

¹T. S. Blyth and M. F. Janowitz, *Residuation theory*, 1972, Pergamon press

Notations

Pour $f : \mathcal{D} \rightarrow \mathcal{C}$

<i>points fixes</i>	$\mathcal{F}_f = \{x \mid f(x) = x\}$
<i>points post-fixes</i>	$\mathcal{P}_f = \{x \mid f(x) \succeq x\}$
<i>points pré-fixes</i>	$\mathcal{Q}_f = \{x \mid f(x) \preceq x\}$

Lorsque f est isotone²

Théorème

\mathcal{P}_f , \mathcal{F}_f et \mathcal{Q}_f sont des treillis complets.

²A. Tarski, *A lattice theoretical fixed point theorem and its applications*, 1955, Pacific Journal of Maths.

Lorsque f est isotone²

Théorème

\mathcal{P}_f , \mathcal{F}_f et \mathcal{Q}_f sont des treillis complets.

Théorème

$$\text{Sup } \mathcal{P}_f = \text{Sup } \mathcal{F}_f$$

²A. Tarski, *A lattice theoretical fixed point theorem and its applications*, 1955, Pacific Journal of Maths.

Lorsque f est isotone²

Théorème

\mathcal{P}_f , \mathcal{F}_f et \mathcal{Q}_f sont des treillis complets.

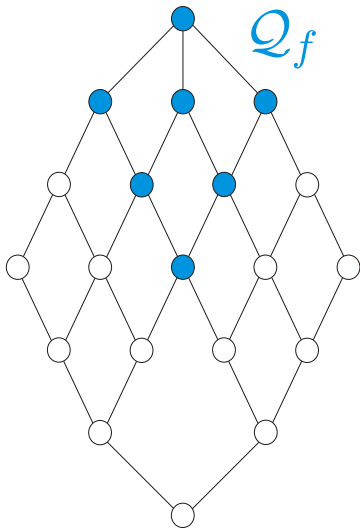
Théorème

$$\text{Sup } \mathcal{P}_f = \text{Sup } \mathcal{F}_f$$

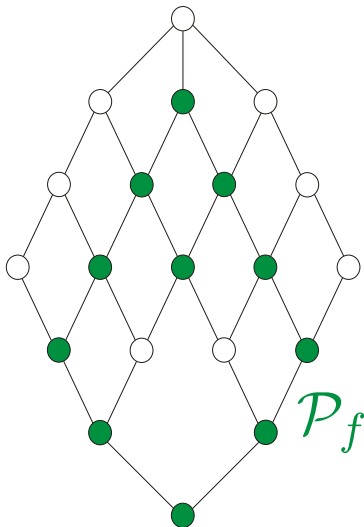
$$\text{Inf } \mathcal{Q}_f = \text{Inf } \mathcal{F}_f$$

²A. Tarski, *A lattice theoretical fixed point theorem and its applications*, 1955, Pacific Journal of Maths.

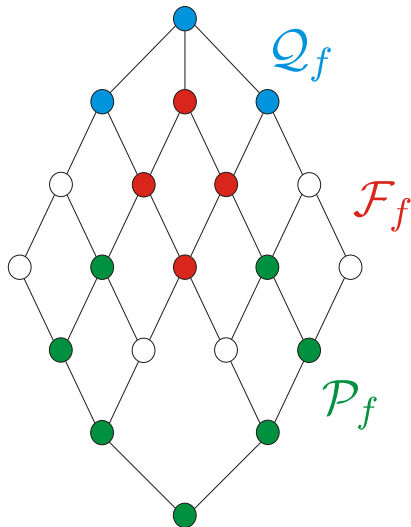
$$\begin{aligned} \mathcal{F}_f &= \{x \mid f(x) = x\} \\ \mathcal{P}_f &= \{x \mid f(x) \succ x\} \\ \mathcal{Q}_f &= \{x \mid f(x) \prec x\} \end{aligned}$$



$$\begin{aligned} \mathcal{F}_f &= \{x \mid f(x) = x\} \\ \mathcal{P}_f &= \{x \mid f(x) \succ x\} \\ \mathcal{Q}_f &= \{x \mid f(x) \prec x\} \end{aligned}$$



$$\begin{aligned} \mathcal{F}_f &= \{x \mid f(x) = x\} \\ \mathcal{P}_f &= \{x \mid f(x) \succ x\} \\ \mathcal{Q}_f &= \{x \mid f(x) \prec x\} \end{aligned}$$



Proposition : plus grand point fixe de f

Si la suite définie par

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

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

Proposition : plus grand point fixe de f

Si la suite définie par

$$\begin{aligned}y_0 &= T \\ y_{k+1} &= f(y_k)\end{aligned}$$

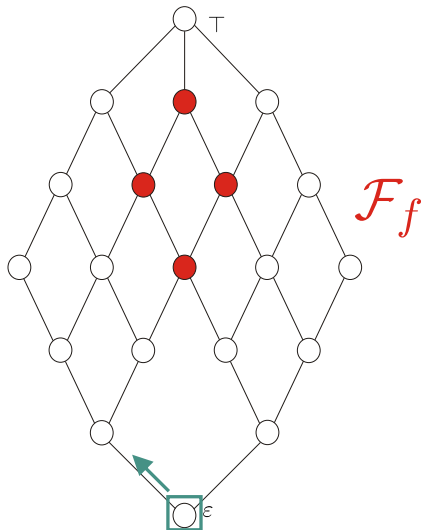
converge en un nombre fini k_e d'itérations, alors y_{k_e} est le plus grand point fixe de f .

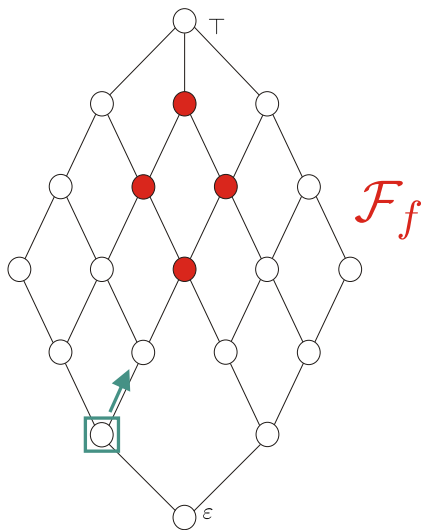
Proposition : plus petit point fixe de f

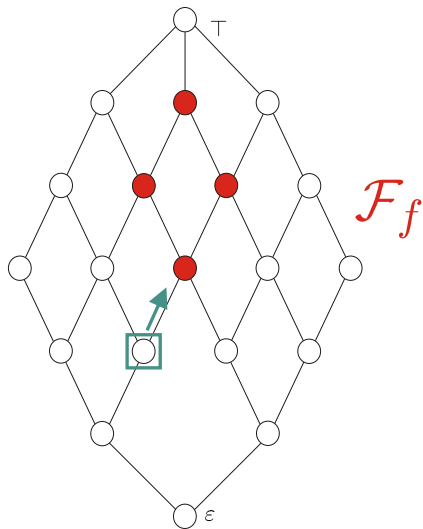
Si la suite définie par

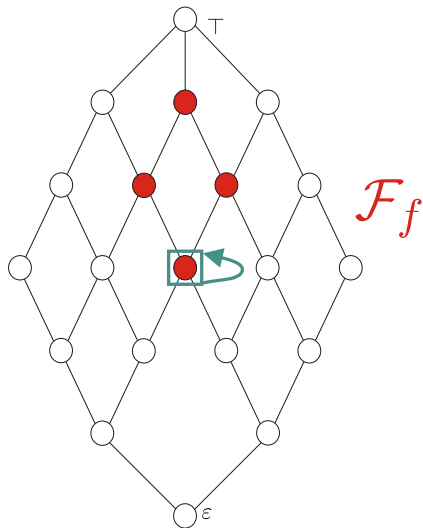
$$\begin{aligned}y_0 &= \varepsilon \\ y_{k+1} &= f(y_k)\end{aligned}$$

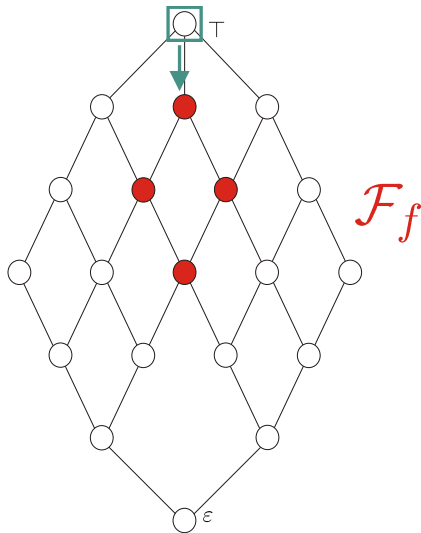
converge en un nombre fini k'_e d'itérations, alors $y_{k'_e}$ est le plus petit point fixe de f .

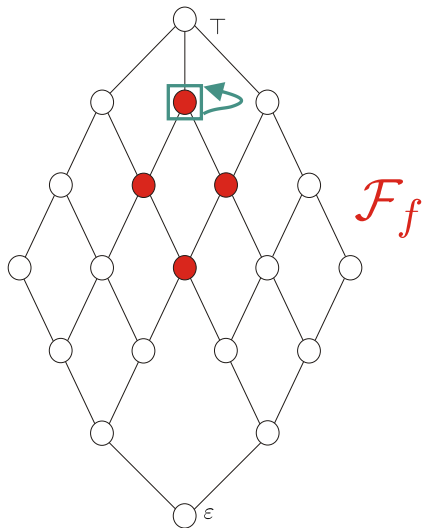












Lorsque f est antitone

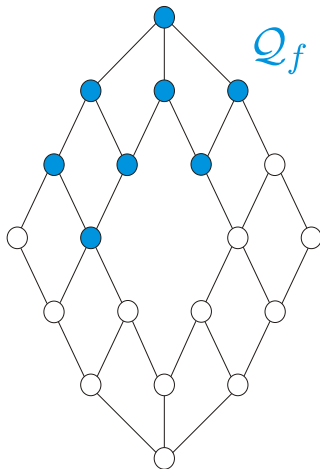
Proposition

\mathcal{P}_f est un demi-treillis inférieur.

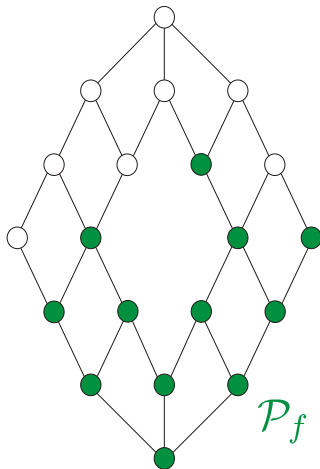
\mathcal{Q}_f est un demi-treillis supérieur.

Si \mathcal{F}_f n'est pas vide, il forme une anti-chaîne.

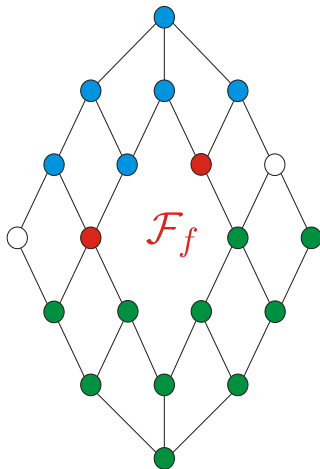
$$\begin{aligned} \mathcal{F}_f &= \{x \mid f(x) = x\} \\ \mathcal{P}_f &= \{x \mid f(x) \succ x\} \\ \mathcal{Q}_f &= \{x \mid f(x) \prec x\} \end{aligned}$$



$$\begin{aligned} \mathcal{F}_f &= \{x \mid f(x) = x\} \\ \mathcal{P}_f &= \{x \mid f(x) \neq x\} \\ \mathcal{Q}_f &= \{x \mid f(x) \perp x\} \end{aligned}$$



$$\begin{aligned} \mathcal{F}_f &= \{x \mid f(x) = x\} \\ \mathcal{P}_f &= \{x \mid f(x) \succ x\} \\ \mathcal{Q}_f &= \{x \mid f(x) \prec x\} \end{aligned}$$



lorsque f est antitone

Propriété

f antitone $\implies f \circ f = f^2$ isotone.

lorsque f est antitone

Propriété

f antitone $\implies f \circ f = f^2$ isotone.

Proposition

En notant $u = \text{Inf } \mathcal{F}_{f^2}$ et $v = \text{Sup } \mathcal{F}_{f^2}$, on a $u \in \mathcal{P}_f$ et $v \in \mathcal{Q}_f$.

lorsque f est antitone

Propriété

f antitone $\implies f \circ f = f^2$ isotone.

Proposition

En notant $u = \text{Inf } \mathcal{F}_{f^2}$ et $v = \text{Sup } \mathcal{F}_{f^2}$, on a $u \in \mathcal{P}_f$ et $v \in \mathcal{Q}_f$.

Propriétés

- $\forall x \succeq v \implies x \in \mathcal{Q}_f$
- $\forall x' \in \mathcal{F}_f \implies u \preceq x' \preceq v$

lorsque f est antitone

Propriété

f antitone $\implies f \circ f = f^2$ isotone.

Proposition

En notant $u = \text{Inf } \mathcal{F}_{f^2}$ et $v = \text{Sup } \mathcal{F}_{f^2}$, on a $u \in \mathcal{P}_f$ et $v \in \mathcal{Q}_f$.

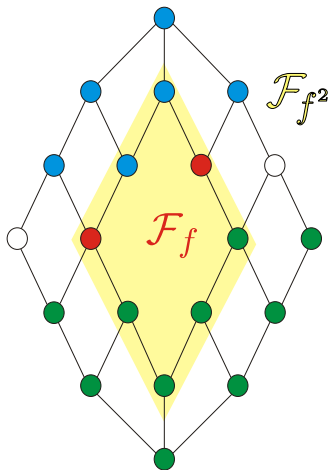
Propriétés

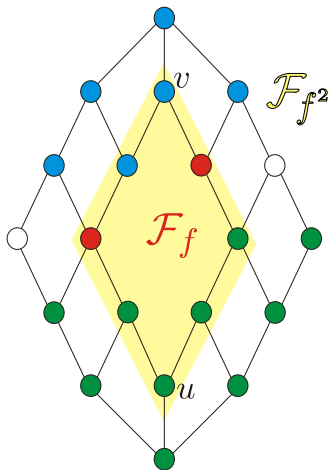
- $\forall x \succeq v \Rightarrow x \in \mathcal{Q}_f$
- $\forall x' \in \mathcal{F}_f \Rightarrow u \preceq x' \preceq v$

Corolaire

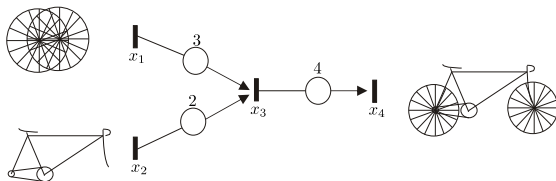
Si $v = u$, v est un élément minimal de \mathcal{Q}_f .

Sinon v sera utilisé comme une approximation d'un élément minimal de \mathcal{Q}_f .



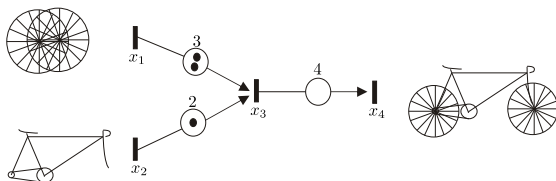


Systèmes ($\max, +$)-linéaires



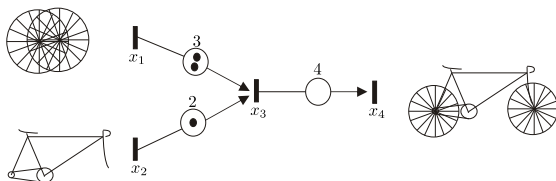
$$x_4(k) = 4 + \max(3 + x_1(k), 2 + x_2(k))$$

Systèmes $(max, +)$ -linéaires



$$x_4(k) = 4 + \max(3 + x_1(k - 2), 2 + x_2(k - 1))$$

Systèmes $(max, +)$ -linéaires

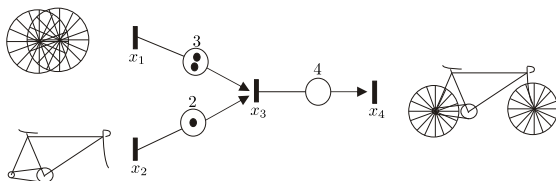


$$x_4(k) = 4 + \max(3 + x_1(k-2), 2 + x_2(k-1))$$

Dans l'algèbre $(max, +)$, cette équation devient :

$$x_4(k) = 4 \otimes (3 \otimes x_1(k-2) \oplus 2 \otimes x_2(k-1))$$

Systèmes $(max, +)$ -linéaires



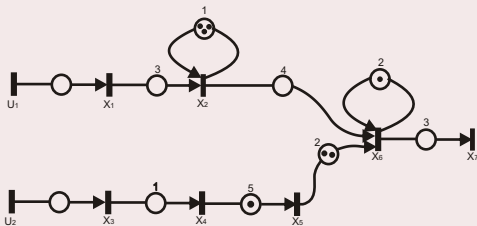
$$x_4(k) = 4 + \max(3 + x_1(k - 2), 2 + x_2(k - 1))$$

Dans l'algèbre $(max, +)$, cette équation devient :

$$x_4(k) = 4 \otimes (3 \otimes x_1(k - 2) \oplus 2 \otimes x_2(k - 1))$$

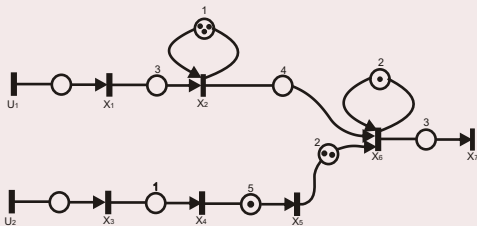
$$x_4(k) = 7x_1(k - 2) \oplus 6x_2(k - 1)$$

Représentation de GET dans un dioïde



- *dateur* $x(k)$: date de la $k + 1$ ème occurrence

Représentation de GET dans un dioïde



► *dateur* $x(k)$: date de la $k + 1$ ème occurrence

► représentation d'état :
$$\begin{cases} x(k) &= Ax(k-1) \oplus Bu(k) \\ y(k) &= Cx(k) \end{cases}$$

Représentation dans un dioïde

- ▶ transformée en γ, δ d'un dateur $x(k)$: $x = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{x(n)}$

$\mathcal{M}_{in}^{ax}[\gamma, \delta]$: dioïde des séries formelles en γ, δ

Représentation dans un dioïde

- ▶ transformée en γ, δ d'un dateur $x(k)$: $x = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{x(n)}$

$\mathcal{M}_{in}^{ax}[\gamma, \delta]$: dioïde des séries formelles en γ, δ

- ▶ représentation d'état dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:
$$\begin{cases} x & = & Ax \oplus Bu \\ y & = & Cx \end{cases}$$

Représentation dans un dioïde

- ▶ transformée en γ, δ d'un dateur $x(k)$: $x = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{x(n)}$

$\mathcal{M}_{in}^{ax}[\gamma, \delta]$: dioïde des séries formelles en γ, δ

- ▶ représentation d'état dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:
$$\begin{cases} x &= Ax \oplus Bu \\ y &= Cx \end{cases}$$
- ▶ représentation entrées/sorties : $y = CA^*B \otimes u = H \otimes u$
 H : matrice de transfert

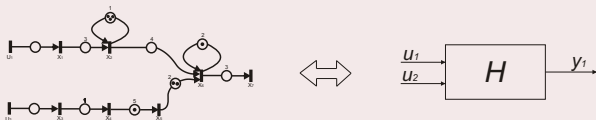
Représentation dans un dioïde

- ▶ transformée en γ, δ d'un dateur $x(k) : x = \bigoplus_{n \in \mathbb{Z}} \gamma^n \delta^{x(n)}$

$\mathcal{M}_{in}^{ax}[\gamma, \delta]$: dioïde des séries formelles en γ, δ

- ▶ représentation d'état dans $\mathcal{M}_{in}^{ax}[\gamma, \delta] : \begin{cases} x &= Ax \oplus Bu \\ y &= Cx \end{cases}$

- ▶ représentation entrées/sorties : $y = CA^*B \otimes u = H \otimes u$
 H : matrice de transfert



Le critère du juste-à-temps

- effectuer le nombre minimum d'occurrences des évènements d'entrées

Le critère du juste-à-temps

- effectuer le nombre minimum d'occurrences des évènements d'entrées
- retarder au maximum ces occurrences

tout en satisfaisant des objectifs de commande.

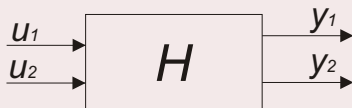
Commande en juste-à-temps

Le critère du juste-à-temps

- effectuer le nombre minimum d'occurrences des évènements d'entrées
- retarder au maximum ces occurrences

tout en satisfaisant des objectifs de commande.

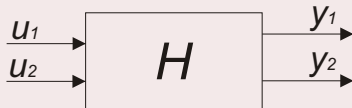
Traduction dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$



$$y = Hu$$

rechercher la plus grande commande u

La poursuite de trajectoire en juste-à-temps³

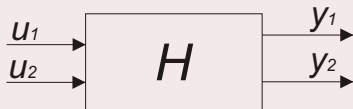


Trouver le plus grand u tel que $y = Hu \preceq z$.

- structure de commande : boucle ouverte
- objectif de commande : poursuite de trajectoire
- critère : juste-à-temps

³G. Cohen, P. Moller, J. P. Quadrat et M. Viot, Algebraic Tools for the Performance Evaluation of Discrete Event Systems, 1989, *IEEE Proceedings : Special issue on Discrete Event Systems*.

La poursuite de trajectoire en juste-à-temps³



Trouver le plus grand u tel que $y = Hu \preceq z$.

- structure de commande : boucle ouverte
- objectif de commande : poursuite de trajectoire
- critère : juste-à-temps

→ solution (donnée par la théorie de la résiduation) :

$$u_{opt} = H \setminus z$$

³G. Cohen, P. Moller, J. P. Quadrat et M. Viot, Algebraic Tools for the Performance Evaluation of Discrete Event Systems, 1989, *IEEE Proceedings : Special issue on Discrete Event Systems*.

Généralisation de la commande en juste-à-temps

- structure de commande : boucle ouverte

Généralisation de la commande en juste-à-temps

- structure de commande : boucle ouverte
- objectif de commande : contraintes sur l'état du système

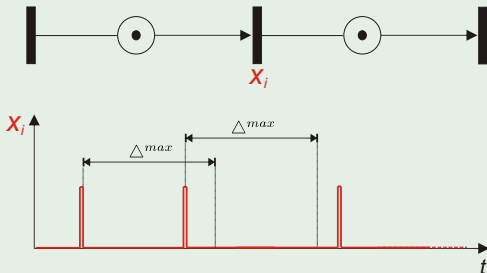
Généralisation de la commande en juste-à-temps

- structure de commande : boucle ouverte
- objectif de commande : contraintes sur l'état du système
- critère : juste-à-temps

Exemples de contrainte

- ▶ temps maximum de séparation entre deux occurrences d'un événement

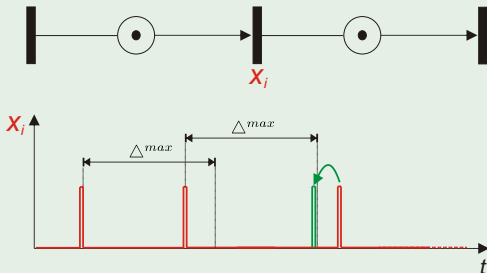
$$x_i(k+1) \leq \Delta^{max} x_i(k)$$



Exemples de contrainte

- ▶ temps maximum de séparation entre deux occurrences d'un événement

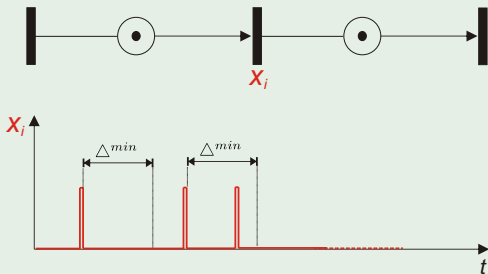
$$x_i(k+1) \leq \Delta^{max} x_i(k)$$



Exemples de contrainte

- ▶ temps minimum de séparation entre deux occurrences d'un événement

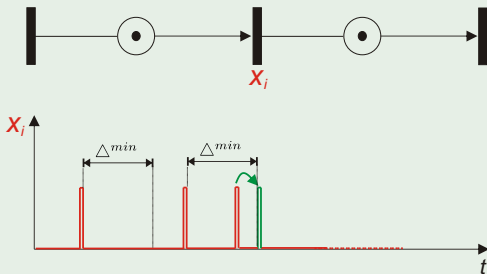
$$x_i(k+1) \succeq \Delta^{\min} x_i(k)$$



Exemples de contrainte

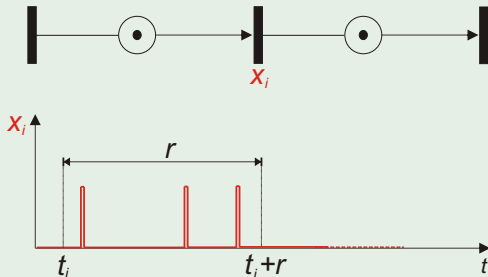
- ▶ temps minimum de séparation entre deux occurrences d'un événement

$$x_i(k+1) \succeq \Delta^{min} x_i(k)$$



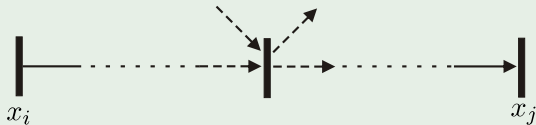
Exemples de contrainte

- ▶ nombre désiré d'occurrences d'un événement pendant un intervalle de temps



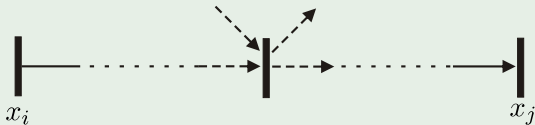
Exemples de contrainte

- ▶ borner la durée de certaines activités



Exemples de contrainte

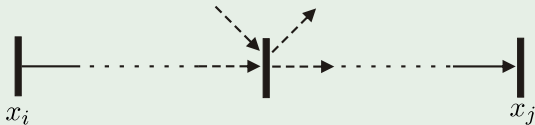
- ▶ borner la durée de certaines activités



- ▶ borner le nombre de jetons dans certains chemins d'un GET

Exemples de contrainte

- ▶ borner la durée de certaines activités



- ▶ borner le nombre de jetons dans certains chemins d'un GET
- ▶ poursuite d'une trajectoire de référence

Synthèse

Toutes ces contraintes se reformulent sur le vecteur de commande dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$u \preceq g'(u)$$

Synthèse

Toutes ces contraintes se reformulent sur le vecteur de commande dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$u \preceq g'(u)$$

avec q contraintes

$$\left\{ \begin{array}{l} u \preceq g_1'(u) \\ u \preceq g_2'(u) \\ \vdots \\ u \preceq g_q'(u), \end{array} \right.$$

Synthèse

Toutes ces contraintes se reformulent sur le vecteur de commande dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$u \preceq g'(u)$$

avec q contraintes

$$\left\{ \begin{array}{l} u \preceq g_1'(u) \\ u \preceq g_2'(u) \\ \vdots \\ u \preceq g_q'(u), \end{array} \right.$$

$$\Leftrightarrow u \preceq g_1'(u) \wedge g_2'(u) \wedge \dots \wedge g_q'(u) \triangleq f(u).$$

Proposition

Si la procédure itérative suivante

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

converge en un nombre fini k_e d'itérations, alors u_{k_e} est la commande recherchée.

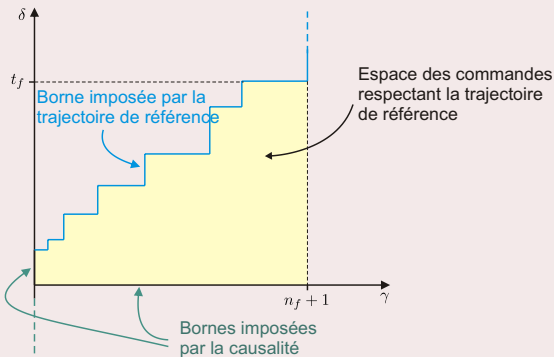
Proposition

Si parmi les contraintes, une poursuite de trajectoire est considérée alors la suite converge.

Proposition

Si parmi les contraintes, une poursuite de trajectoire est considérée alors la suite converge.

preuve



Réseaux de transport

La génération de la table d'horaire fait partie d'une optimisation qui consiste en :

⁴ A. Ceder and B. Golany and O. Tal, 2003, Creating bus timetables with maximal synchronization.

⁵ A. Naït-Sidi-Moh, 2001, Contribution à la modélisation, à l'analyse et à la commande des systèmes de transport public par les réseaux de Petri et l'algèbre (Max,plus).

Réseaux de transport

La génération de la table d'horaire fait partie d'une optimisation qui consiste en :

- 1 la construction du réseau (localisation des arrêts, allocation des bus,...)

⁴ A. Ceder and B. Golany and O. Tal, 2003, Creating bus timetables with maximal synchronization.

⁵ A. Naït-Sidi-Moh, 2001, Contribution à la modélisation, à l'analyse et à la commande des systèmes de transport public par les réseaux de Petri et l'algèbre (Max,plus).

Réseaux de transport

La génération de la table d'horaire fait partie d'une optimisation qui consiste en :

- 1 la construction du réseau (localisation des arrêts, allocation des bus,...)
- 2 l'établissement fréquences de passage des bus pour chaque ligne (écart-temps minimum et écart-temps maximum)

⁴ A. Ceder and B. Golany and O. Tal, 2003, Creating bus timetables with maximal synchronization.

⁵ A. Naït-Sidi-Moh, 2001, Contribution à la modélisation, à l'analyse et à la commande des systèmes de transport public par les réseaux de Petri et l'algèbre (Max,plus).

Réseaux de transport

La génération de la table d'horaire fait partie d'une optimisation qui consiste en :

- 1 la construction du réseau (localisation des arrêts, allocation des bus,...)
- 2 l'établissement fréquences de passage des bus pour chaque ligne (écart-temps minimum et écart-temps maximum)
- 3 le calcul des tables d'horaires^{4 5}

⁴ A. Ceder and B. Golany and O. Tal, 2003, Creating bus timetables with maximal synchronization.

⁵ A. Naït-Sidi-Moh, 2001, Contribution à la modélisation, à l'analyse et à la commande des systèmes de transport public par les réseaux de Petri et l'algèbre (Max,plus).

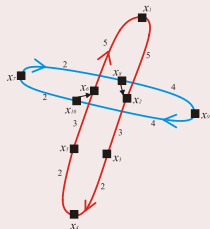
Notre approche

- modélisation du réseau dans l'algèbre ($max, +$)

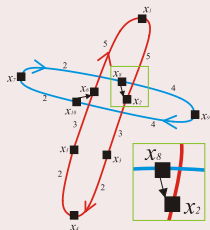
Notre approche

- modélisation du réseau dans l'algèbre ($max, +$)
- synthèse des tables d'horaires
 - ▶ formuler
 - ▶ résoudre

le problème en un problème de commande en juste-à-temps de systèmes ($max, +$)-linéaires contraints



$x_i(k)$: date de départ du $k + 1$ -ème bus à l'arrêt i .



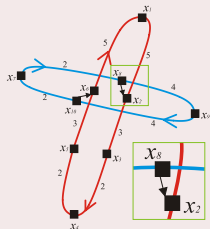
$x_i(k)$: date de départ du $k + 1$ -ème bus à l'arrêt i .

$$x(k) \geq Ax(k - 1)$$

A_{ij} : temps de parcours de j à i

$$x_2(k) = 5 \otimes x_1(k) \oplus x_8(k)$$

dynamique du système



$x_i(k)$: date de départ du $k + 1$ -ème bus à l'arrêt i .

$$x(k) \geq Ax(k - 1)$$

A_{ij} : temps de parcours de j à i

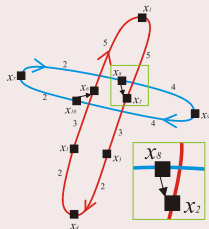
$$x_2(k) = 5 \otimes x_1(k) \oplus x_8(k)$$

influence externe

$$x(k) \geq Bu(k)$$

$B_{ij} = e$ si les bus respectent la table d'horaires à l'arrêt i , $B_{ij} = \varepsilon$ sinon

dynamique du système



$x_i(k)$: date de départ du $k + 1$ -ème bus à l'arrêt i .

$$x(k) \geq Ax(k - 1)$$

A_{ij} : temps de parcours de j à i

$$x_2(k) = 5 \otimes x_1(k) \oplus x_8(k)$$

influence externe

$$x(k) \geq Bu(k)$$

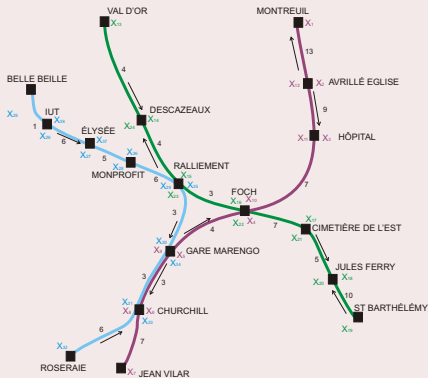
$B_{ii} = e$ si les bus respectent la table d'horaires à l'arrêt i , $B_{ij} = \varepsilon$ sinon

représentation d'état

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

Contraintes

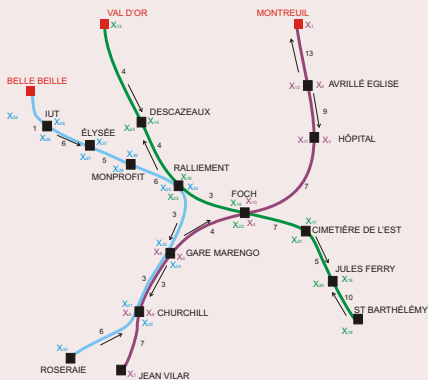
- ▷ échéances à certains arrêts stratégiques du réseaux :



Contraintes

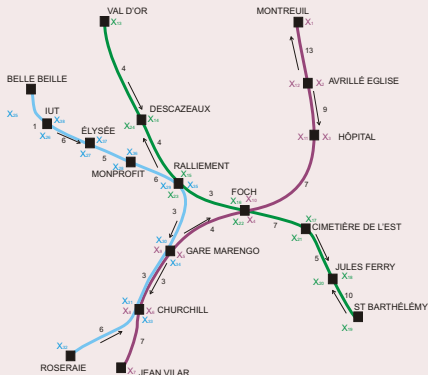
▷ échéances à certains arrêts stratégiques du réseaux :

	6 :30	6 :50	7 :10	7 :30	7 :50	8 :10	8 :30
x_1 (Montreuil)	1	4	3	3	1	2	
x_{13} (Val d'or)	1	2	3	3	4	4	
x_{25} (Belle Beille)	1	2	4	4	2	2	



Contraintes

- ▷ écart-temps maximum :
temps d'écart maximum entre deux départs de bus à l'arrêt X_i .

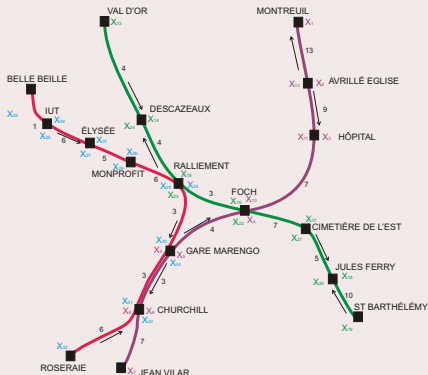


Contraintes

- ▷ écart-temps maximum :
temps d'écart maximum entre deux départs de bus à l'arrêt

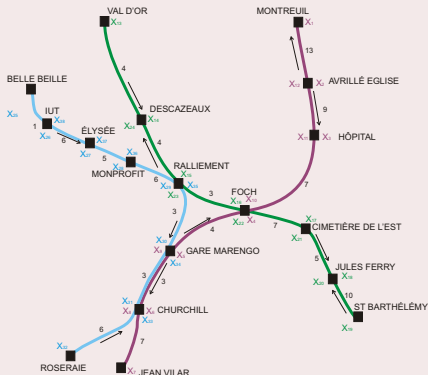
X_j .

$$\triangle^{max} = 16$$



Contraintes

- ▷ écart-temps minimum :
temps d'écart minimum entre deux départs de bus à l'arrêt X_j .



Contraintes

- ▷ écart-temps minimum :
temps d'écart minimum entre deux départs de bus à l'arrêt

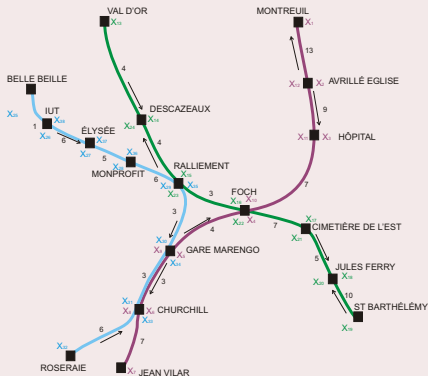
X_j .

$$\Delta^{min} = 5$$



Contraintes

- ▶ absorption de la charge
nombre de départs de bus dans un intervalle donné à un arrêt.



Contraintes

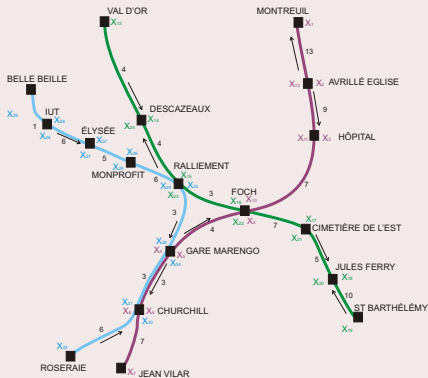
- ▶ absorption de la charge
nombre de départs de bus dans un intervalle donné à un arrêt.

3 départs de bus entre 7 :30 et 7 :50 à l'arrêt x_{26} (IUT)



Contraintes

- ▷ temps d'attente maximum des bus à un arrêt ou temps de parcours maximum sur un chemin



Contraintes

- ▷ temps d'attente maximum des bus à un arrêt ou temps de parcours maximum sur un chemin

Les bus ne stationnent pas plus de 2 min à l'arrêt Foch



Résolution

On peut exprimer toutes ces contraintes sous une forme générale

$$u \preceq f(u)$$

où u représente la table d'horaires et f une application isotone formulant les contraintes considérées.

Résolution

On peut exprimer toutes ces contraintes sous une forme générale

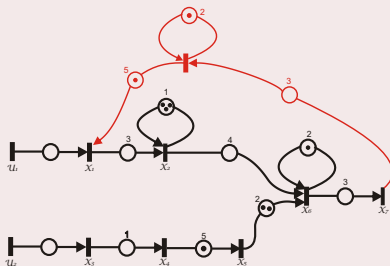
$$u \preceq f(u)$$

où u représente la table d'horaires et f une application isotone formulant les contraintes considérées.

La [suite itérative](#) présentée précédemment converge et u_{k_e} est la table d'horaires optimale.

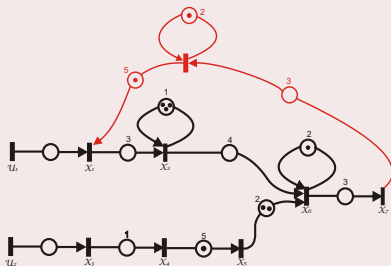
Commande de systèmes contraints en minimisant les retards

Correcteurs ($\max, +$) en boucle fermée



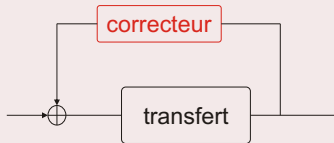
Commande de systèmes contraints en minimisant les retards

Correcteurs ($\max, +$) en boucle fermée



ralentissement du transfert du système

Présentation du problème : critère

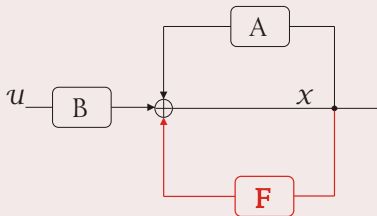


trouver un correcteur le plus petit possible.

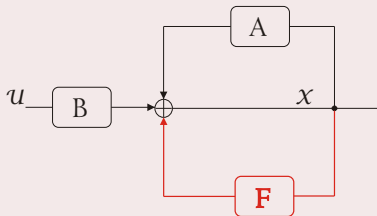
Présentation du problème : objectif de commande

Le système corrigé respecte des spécifications : $x \vDash \phi x$

Présentation du problème : structure de commande

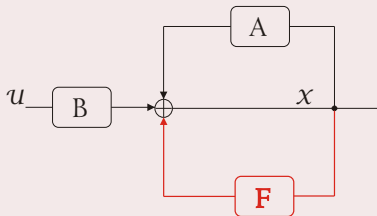


Présentation du problème : structure de commande



$$x = Ax \oplus Fx \oplus Bu$$

Présentation du problème : structure de commande



$$x = Ax \oplus Fx \oplus Bu$$

fonctionnement au plus tôt : $x = (A \oplus F)^* Bu$

$$x \succeq \phi x \quad \forall u$$

on cherche le plus petit état corrigé x qui satisfait

$$x \succeq \phi x \oplus Ax \oplus Bu \quad \forall u$$

$$x \succeq \phi x \quad \forall u$$

on cherche le plus petit état corrigé x qui satisfait

$$x \succeq \phi x \oplus Ax \oplus Bu \quad \forall u$$

ce qui revient à trouver le plus petit F tel que

$$(A \oplus F)^* B \succeq (A \oplus \phi)^* B \quad (1)$$

- $(A \oplus \phi)^* B$ causale \Leftrightarrow il existe un correcteur F causal.

- $(A \oplus \phi)^* B$ causale \Leftrightarrow il existe un correcteur F causal.
- Si $(A \oplus \phi)^* B$ n'est pas causale, on a mis à jour une CNS d'existence d'une surapproximation causale permettant le calcul d'un correcteur F causal.

Proposition

Les solutions de (1) correspondent aux éléments de \mathcal{Q}_f avec $f : F \mapsto (A \oplus \phi)^* \ominus (A \oplus F)^*$.

Proposition

Les solutions de (1) correspondent aux éléments de \mathcal{Q}_f avec $f : F \mapsto (A \oplus \phi)^* \ominus (A \oplus F)^*$.

Calcul de F

f antitone $\Rightarrow \mathcal{Q}_f$ est un **demi-treillis supérieur**.

Proposition

Les solutions de (1) correspondent aux éléments de \mathcal{Q}_f avec $f : F \mapsto (A \oplus \phi)^* \ominus (A \oplus F)^*$.

Calcul de F

f antitone $\Rightarrow \mathcal{Q}_f$ est un **demi-treillis supérieur**.

\mathcal{Q}_f n'admet pas nécessairement de plus petit élément cependant on peut calculer $v = \text{Sup } \mathcal{F}_f$.

Proposition

Les solutions de (1) correspondent aux éléments de Q_f avec $f : F \mapsto (A \oplus \phi)^* \ominus (A \oplus F)^*$.

Calcul de F

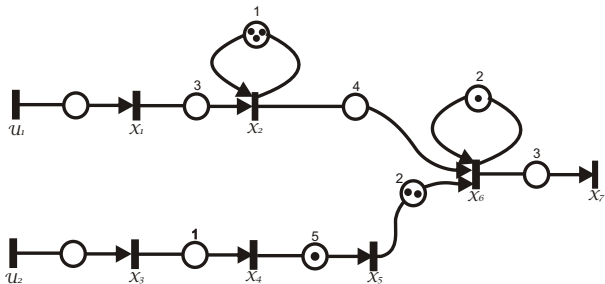
f antitone $\Rightarrow Q_f$ est un **demi-treillis supérieur**.

Q_f n'admet pas nécessairement de plus petit élément cependant on peut calculer $v = \text{Sup } \mathcal{F}_f$.

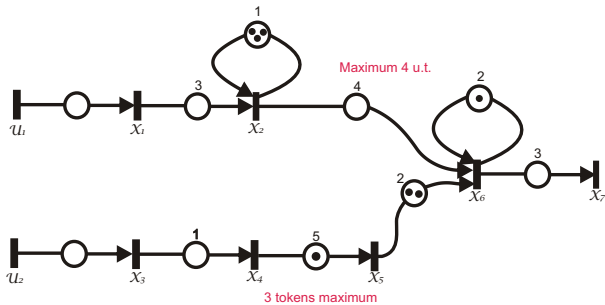
Si $v = u$, v est un élément minimal de Q_f .

Sinon v peut être utilisé comme une approximation d'un feedback minimal.

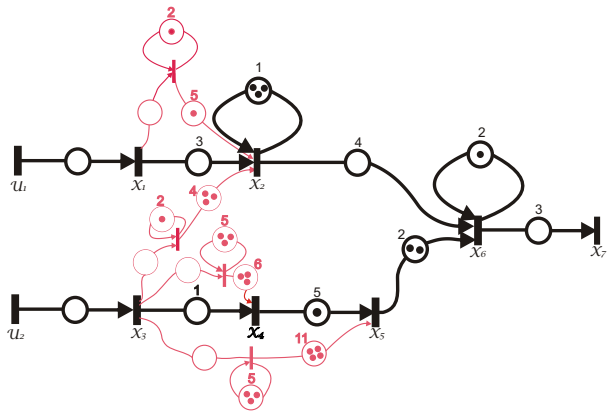
Exemple



Exemple



Exemple



Conclusion et perspectives

- La commande minimisant l'effet de la correction ne fournit pas nécessairement un correcteur minimal.

Conclusion et perspectives

- La commande minimisant l'effet de la correction ne fournit pas nécessairement un correcteur minimal.
- On peut envisager la mise à jour de conditions moins restrictives pour l'utilisation d'autres structures.

La théorie des tas⁶

Mise en oeuvre d'algorithmes de calcul du makespan pour un ordonnancement périodique.

⁶S. Gaubert, J. Mairesse, *Modeling and Analysis of Timed Petri Nets Using Heaps of Pieces*, 1999, IEEE TAC

La théorie des tas⁶

Mise en oeuvre d'algorithmes de calcul du makespan pour un ordonnancement périodique.

→ 2 approches.

⁶S. Gaubert, J. Mairesse, *Modeling and Analysis of Timed Petri Nets Using Heaps of Pieces*, 1999, IEEE TAC

Approche classique

Pour un séquençement donné :

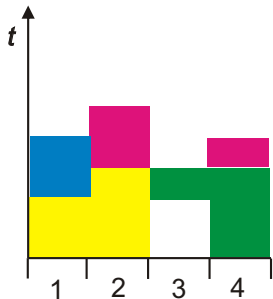
- construction du GET associé à la séquence
- construction du modèle ($max, +$)

Approche classique

Pour un séquençement donné :

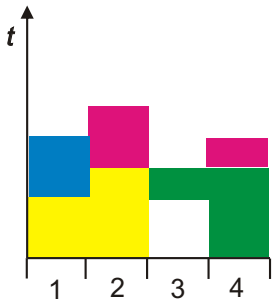
- construction du GET associé à la séquence
- construction du modèle ($max, +$)

Le calcul de la valeur propre de la matrice d'évolution fournit le temps de cycle de la séquence.



Théorie des tas

$x(a)$ correspond à un vecteur contenant les dates de libération des ressources après la tâche élémentaire a .



Théorie des tas

$x(a)$ correspond à un vecteur contenant les dates de libération des ressources après la tâche élémentaire a .

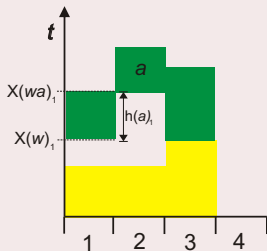
$x(w)$ vecteur contenant les dates de libération des ressources après le mot $w = a_1 a_2 \dots a_k$

Théorie des tas

La dynamique du système se traduit par :

$$\forall r \in \mathcal{R}(a), \quad x(wa)_r = \max_{s \in \mathcal{R}(a)} (h(a)_r + x(w)_s)$$

$$\forall r \in \mathcal{R}(a), \quad x(wa)_r = x(w)_r$$



Théorie des tas

A chaque tâche élémentaire a , on fait correspondre une matrice $M(a)$.

Théorie des tas

A chaque tâche élémentaire a , on fait correspondre une matrice $M(a)$. La séquence est alors modélisée par un système $(\max, +)$ -linéaire. Par exemple,

$$x(k) = M(a) \otimes M(b) \otimes M(c) \otimes M(d) \otimes x(k - 1)$$

correspond à la séquence de tâches $w = abcd$.

Théorie des tas

A chaque tâche élémentaire a , on fait correspondre une matrice $M(a)$. La séquence est alors modélisée par un système $(max, +)$ -linéaire. Par exemple,

$$x(k) = M(a) \otimes M(b) \otimes M(c) \otimes M(d) \otimes x(k - 1)$$

correspond à la séquence de tâches $w = abcd$.

La valeur propre de la matrice $M(w) = M(a)M(b)M(c)M(d)$ correspond au temps de cycle de la séquence w .

Théorie des tas

- ▶ Complexité identique.

Théorie des tas

- ▶ Complexité identique.
- ▶ Pour chaque séquencement, dans la méthode classique, on doit remodeliser.

Théorie des tas

- ▶ Complexité identique.
- ▶ Pour chaque séquençement, dans la méthode classique, on doit remodeliser.
- ▶ Avec la théorie des tas, la modélisation reste la même. Il suffit de recalculer la matrice correspondant au nouveau mot. Par exemple :
$$M(w') = M(a)M(b)M(d)M(c).$$

Théorie des tas

- ▶ Complexité identique.

Théorie des tas

- ▶ Complexité identique.
- ▶ Pour chaque séquencement, dans la méthode classique, on doit remodeliser.

Théorie des tas

- ▶ Complexité identique.
- ▶ Pour chaque séquençement, dans la méthode classique, on doit remodeliser.
- ▶ Avec la théorie des tas, la modélisation reste la même. Il suffit de recalculer la matrice correspondant au nouveau mot. Par exemple :
$$M(w') = M(a)M(b)M(d)M(c).$$

Algèbre $(\max, +)$ et flowshop⁷

Résolution de problème de flowshop à 2 machines.

$$C(i) = C(i-1) \otimes \begin{pmatrix} p^{(i)1} & p^{(i)1}p^{(i)2} \\ \varepsilon & p^{(i)2} \end{pmatrix} = C(i-1)M(i).$$

⁷J.L. Bouquard, C. Lenté, J.C. Billaut *Application of an optimization problem in Max-Plus algebra to scheduling problem*, 2006, Discrete Applied Mathematics

Algèbre $(max, +)$ et flowshop⁷

Résolution de problème de flowshop à 2 machines.

$$C(i) = C(i-1) \otimes \begin{pmatrix} P^{(i)1} & P^{(i)1}P^{(i)2} \\ \varepsilon & P^{(i)2} \end{pmatrix} = C(i-1)M(i).$$

$$C(i) = C(0)M(1) \otimes M(2) \otimes \dots \otimes M(i)$$

En utilisant une propriété des matrices triangulaires, les auteurs parviennent à trouver le séquençement optimale.
(revient à l'algorithme de Johnson).

⁷J.L. Bouquard, C. Lenté, J.C. Billaut *Application of an optimization problem in Max-Plus algebra to scheduling problem*, 2006, Discrete Applied Mathematics

Algèbre ($max, +$) et flowshop

Résolution de problème de flowshop à 3 machines.

Algèbre ($\max, +$) et flowshop

Résolution de problème de flowshop à 3 machines.

Algèbre ($max, +$) et flowshop

Résolution de problème de flowshop à 3 machines.

Le problème devient NP-complet.

proposition d'un algorithme de branch-and-bound.

Conclusion

- mise en place de méthodes heuristiques pour le calcul d'un séquençement pour le problème de jobshop.
- idem pour le problème de flowshop.
- généralisation du problème de flowshop à plusieurs machines.