Calcul du temps de réponse de systèmes temps réel dans le semi-anneau (Max,+)

Anne Bouillard

31 octobre 2001

1 Introduction

J'ai effectué mon stage au sein de l'équipe TRIO (Temps Réel et InterOpérabilité) du LORIA à Nancy. Cette équipe travaille sur le thème de la conception d'applications temps réel. Ses principaux axes de recherche portent sur les mécanismes exécutifs et protocoles, permettant par exemple le respect de contraintes temporelles, les techniques de vérification de propriétés temporelles, et les procédés de modélisation.

On représente des processus dont les comportements sont synchronisés par des graphes d'événements. De tels systèmes apparaissent dans la modélisation d'un ensemble de tâches, effectuées par un processus embarqué dans un robot par exemple. Nous étudions ici des graphes d'événements représentant des tâches qui interagissent à travers un schéma de préemption. Certaines ont une forte priorité, et doivent par conséquent être exécutées avant celles qui ont de faibles priorités. Mais toutes ces tâches, qu'elles aient une forte ou une faible priorité, doivent obéir à des contraintes temps réel.

Le stage avait pour but de trouver des algorithmes permettant de savoir si ces contraintes sont respectées, à l'aide d'un outil : l'algèbre (max, +).

Nous allons tout d'abord introduire quelques éléments théoriques utiles à la résolution du problème, puis voir comment ils ont été utilisés dans un programme permettant de tester la cohérence des tâches.

2 Définitions et préliminaires

2.1 Synchronisation et préemption

Dans un système, différentes tâches doivent être effectuées. Celles-ci sont périodiques et elles dépendent les unes des autres.Elles ne sont exécutées que lorsque d'autres l'ont été. Il y a donc des périodes d'inactivité. Il est donc important de chercher à diminuer le temps de latence entre les différentes tâches : c'est un des avantages de la synchronisation. La fin de l'exécution d'une tâche donne le signal de départ d'une autre tâche.

Plusieurs types de synchronisation (entre deux tâches) sont possibles :

- ASYN-ASYN : les deux systèmes sont indépendants,
- SYN-SYN : les deux systèmes doivent s'attendre à un "point de rendez-vous" pour que l'exécution puisse se poursuivre,
- ASYN-SYN : la première tâche s'exécute librement, et la deuxième est bloquée jusqu'à ce que la première rejoigne le point de rendez-vous,
- SYN-ASYN : les rôles des deux tâches précédentes sont inversés.

Néanmoins, une synchronisation trop lourde peut engendrer des bloquages et des incohérences temporelles. Il existe un autre mécanisme d'interaction : la préemption.

Un système de contrôle est généralement composé de différents types de calculs : les calculs de vérification, rapides et simples, qui peuvent donc être exécutés assez souvent, renforçant ainsi la robustesse du système, et les mises à jour, habituellement plus longues à effectuer, car nécessitant des calculs complexes. Celles-ci doivent donc avoir une faible priorité pour laisser les autres calculs s'exécuter.

Les tâches du système préemptif sont ordonnées par leur priorité, et le déroulement du calcul se fait de la manière suivante : lorsqu'une tâche avec une grande priorité devient exécutable, celle-ci effectue son cycle de calcul, et toutes les tâches moins prioritaires stoppent leur calcul. A la fin du cycle, les tâches exécutables venant immédiatement après dans l'ordre des priorités reprennent leur exécution là où elles s'étaient arrêtées.

2.2 Réseaux de Petri

Les réseaux de Petri sont des graphes qui permettent de modéliser les tâches. On utilise une version légèrement modifiée du réseau de Petri, qui permet de prendre le temps en compte, pour représenter la durée d'exécution des différentes tâches.

Définition 1 (Réseau de Petri temporisé). $\mathcal{G} = (\mathcal{Q}, \mathcal{P}, M, \sigma)$ *est un réseau de Petri si :*

- $\mathcal{G}$ *est un graphe orienté dont les nœuds forment l'ensemble* $\mathcal{P} \cup \mathcal{Q}$*,*
- $\mathcal{Q}$ *est un ensemble de transitions numérotées de taille* Q*,*
- $\mathcal{P}$ *est un ensemble de places numérotées de taille* P*,*
- *Une transition ne peut être reliée qu'à des places, et inversement.*
- $\sigma = (\sigma_1, ..., \sigma_Q)$ *est l'ensemble des temps de franchissement des transitions*
- $\mu(p)$ *est le nombre de jetons dans la place* p *à l'instant initial.*

On note aussi $^\bullet q$ *et* $q^\bullet$ *les places d'entrée et sortie de la transition q.*

L'évolution d'un réseau de Petri est alors la suivante : une transition q ne peut être active que si toutes les places la précédant possèdent au moins un jeton. Elle est active dès cet instant, et un jeton disparaît dans chaque place de ${}^{\bullet}q$. Au bout d'un temps σ_q, un jeton supplémentaire apparaît dans chaque place de $q^{\bullet}$. Un jeton ne peut activer qu'une transition à la fois. Mais il peut être revendiqué par plusieurs transitions. Il faut alors faire un choix : il y a conflit.

On se limite au cas particulier des graphes d'événements (chaque place n'admet qu'une transition entrante et une transition sortante). Les conflits sont ainsi éliminés, et le graphe est déterministe.

Pour modéliser l'ensemble des tâches par un graphe d'événement, chaque transition représente une action de la tâche, et la présence de jetons indique que la transition qui la précède a été effectuée, et la suivante en attente d'être activée. σ_q représente le temps nécessaire à la tâche modélisée par la transition q pour s'exécuter.

Cette représentation permet de mettre en évidence les éventuels bloquages du système, et les incohérences temporelles du système, telles que les évolutions non périodiques. Par exemple, si il existe un circuit dans le graphe dont toutes les places sont vides, alors il y a bloquage : il n'y aura jamais de jeton dans ce circuit.

Une tâche simple peut être représentée par un graphe à trois transitions : pour la lecture des données, leur traitement, et l'écriture des résultats. On utilise également une quatrième transition pour que le système soit périodiquement activé par une horloge.

Exemple 1. *τ représente la période de l'horloge et t_1 le temps d'exécution de la tâche. Tous les temps τ la tâche pourra donc s'exécuter, sous réserve que l'exécution de la précedente soit terminée.*

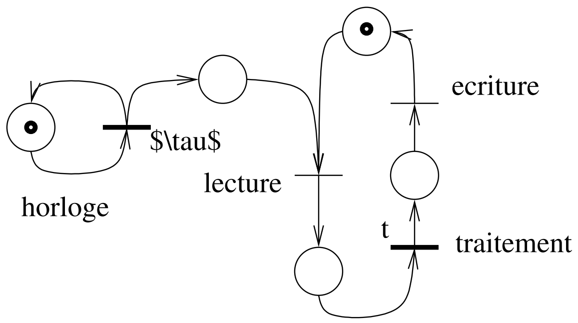

FIG. 1: *Exemple d'une tâche simple.*

Dans toute la suite, nous allons représenter chaque tâche par un réseau de Petri. Une tâche T_i sera représentée par le graphe $\mathcal{G}_i$. De plus, on divise les composantes fortement connexes des graphes en deux parties. I_i est l'ensemble des composantes initiales, aussi appelé horloge. En général, il s'agit d'un simple cycle formé d'une seule transition, mais ce n'est pas obligatoire. O_i est l'ensemble des autres composantes.

On donne une relation d'ordre sur les graphes pour représenter les préemptions. $\mathcal{G}_i \succ \mathcal{G}_j$ signifie qu'à chaque fois qu'une transition de O_i est active, l'évolution des transitions dans O_j est stoppée, et celle-ci reprend dès que toutes les transitions de O_i sont inactives.

Exemple 2. *Donnons un exemple de sychronisation de deux tâches ASYN-SYN :*

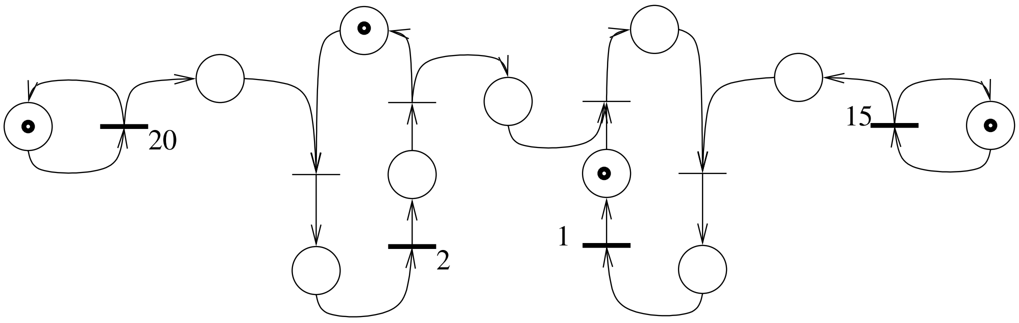

FIG. 2: *Deux tâches synchronisées*

et un exemple de tâches obéissant à un schéma de préemption :

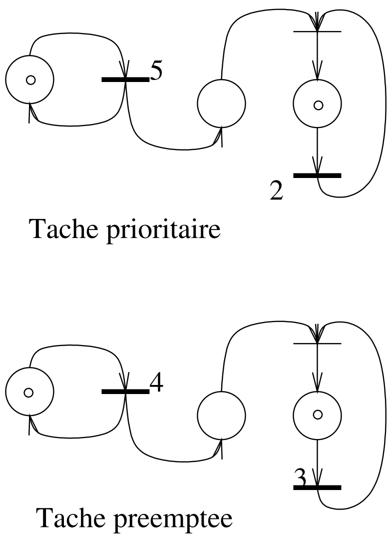

FIG. 3: *Exemple de préemption*

Tous les 5 coups d'horloge, la première tâche est exécutée, et l'exécution dure une unité de temps. La deuxième tâche s'effectue tous les 4 coups d'horloge, mais l'exécution ne dure pas trois unités de temps : elle peut être interrompue si la première tâche doit s'exécuter.

2.3 Position du problème

Le problème est de savoir si les contraintes de temps sont respectées, c'est-à-dire si chaque tâche est toujours exécutée dans le temps qui lui est attribué par l'horloge : c'est le problème le plus général, noté P1 dans la suite. On s'attachera aussi à résoudre le problème moins restrictif (P2) : la contrainte de temps n'est violée qu'un nombre fini de fois.

On peut formuler ces deux problèmes de manière un peu plus formelle :

– Problème P1 : pour tous les graphes $\mathcal{G}_i$, le nombre de jetons dans les places joignant une transition de I_i à une transition de O_i est borné par un,

– Problème P2 : pour tous les graphes $\mathcal{G}_i$, le nombre de jetons dans les places joignant une transition de I_i à une transition de O_i est borné par un, sauf pour un nombre fini de fois.

3 Algèbre (Max, +)

Nous allons introduire l'algèbre (Max,+), qui est adaptée à la modélisation du problème.

3.1 Définition

Définition 2 (Semi-anneau (Max,+)). *C'est le triplet* $(\mathbb{R}_{Max}, \oplus, \otimes)$ *où :*

– $\mathbb{R}_{Max} = \mathbb{R} \cup \{-\infty\}$,

– $\oplus$ *est l'opérateur Max,*

– $\otimes$ *est l'addition.*

L'élément neutre est $-\infty$ que l'on note ε et l'élément unité est 0. $\oplus$ et $\otimes$ sont associatives et commutatives, mais l'opérateur $\oplus$ n'est pas inversible. On peut néanmoins étendre ces opérations au calcul matriciel :

Soient $A, B, C, D \in \mathbb{R}_{Max}^{n \times n}$

1. $C = A \oplus B$ est définie par $C_{i,j} = A_{i,j} \oplus B_{i,j}\ \forall i, j \in \mathbb{N}_n$
2. $D = A \otimes B$ est définie par $D_{i,j} = \bigoplus_{k \in N_n} A_{i,k} \otimes B_{k,j}\ \forall i, j \in \mathbb{N}_n$

En outre, on munit la structure d'une loi de composition externe : $a \otimes A = (a \otimes A_{i,j})_{i,j \in \mathbb{N}_n}$.

Ainsi, on dispose donc d'une algèbre (Max,+), qui conserve certaines propriétés d'une algèbre classique.

De plus, on peut établir une correspondance entre les graphes et les matrices dans l'algèbre (Max,+) : on associe un poids à chaque arc, et on définit la matrice A associée par :

$$A_{i,j} = \begin{cases} p_{i,j} & \text{le poids de la transition entre } j \text{ et } i \text{ si elle existe,} \\ \varepsilon & \text{sinon.} \end{cases} \tag{1}$$

On peut également faire la transformation inverse.

Les puissances n-ièmes de la matrice A représenteront donc les poids maximum des chemins de longueur n joignant les différents sommets.

On note $\mathcal{G}(A)$ le graphe associé à la matrice A.

3.2 Quelques éléments de théorie spectrale

Définition 3. *Une matrice A de dimension n est irréductible si et seulement si pour tout i et tout j de $\mathbb{N}_n$ il existe $n_0 \in \mathbb{N}$ tel que $\forall n \geq n_0 \ \ A^n_{i,j} > \varepsilon$.*

Remarquons qu'une matrice est irréductible si et seulement si son graphe associé est fortement connexe.

Définition 4. *Sous réserve d'existence, on définit les matrices A^* et A^+ par :*

- $A^* = \bigoplus_{n=0}^{\infty} A^n$*, et*
- $A^+ = \bigoplus_{n=1}^{\infty} A^n$.

I est la matrice unité.

Lemme 1. *Soit A une matrice représentant un graphe $\mathcal{G}(A)$ dont tous les circuits sont de poids négatif ou nul. Alors A^* et A^+ sont bien définies dans $\mathbb{R}^{n\times n}_{max}$.*

Démonstration : $A^*_{i,j}$ représente le poids maximum de tous les chemins joignant i à j. Une condition nécessaire et suffisante pour que cette quantité soit bornée pour tout (i, j) est que $\mathcal{G}(A)$ n'ait pas de chemin de poids strictement positif. Sinon, en passant par un tel circuit, on pourrait obtenir un chemin arbitrairement lourd. □

Lemme 2. *Soit A une matrice de dimension n et $\mathcal{G}(A)$ le graphe associé. Si $\mathcal{G}(A)$ n'a pas de chemins de poids strictement positif, alors*

$$A^* = I \oplus A \oplus A^2 \oplus ... \oplus A^{(n-1)} \tag{2}$$

Démonstration : Par argument de cardinalité, tous les chemins de longueur supérieure ou égale à n dans le graphe sont formés d'un circuit au moins et d'un chemin simple de longueur inférieure strictement à n. D'après les hypothèses, le poids du chemin simple sera supérieur au poids du chemin entier. □

Théorème 1. *Soit A une matrice, telle que son graphe associé n'ait que des circuits de poids négatif ou nul. Alors l'équation $x = A \otimes x \oplus b$ admet une plus petite solution, $x = A^*b$. En outre, si les poids des circuits sont tous strictement négatifs, cette solution est unique.*

Démonstration :

Existence de la solution : Le lemme précédent et les hypothèses nous assurent de l'existence de la quantité A^*b.

A^*b est bien solution de l'équation : $A(A^*b) \oplus b = (I \oplus AA^*)b = A^*b$

Unicité de la solution : Soit x une solution de l'équation. On a alors :

$$x = b \oplus Ab \oplus A^2 b \oplus ... \oplus A^{k-1} b \oplus A^k x \tag{3}$$

Donc $x \geq A^* b$. Or puisque les poids des circuits sont strictement négatifs, $A^k \overset{k \to \infty}{\longrightarrow} \varepsilon$. Donc pour k assez grand, on aura bien $x = A^* b$.

□

Définition 5. *Soit A une matrice. λ est une valeur propre de A s'il existe un vecteur x non nul tel que $Ax = \lambda x$.*

Théorème 2. *Si A est une matrice irréductible, elle admet une unique valeur propre, égale au poids moyen maximum des circuits de $\mathcal{G}(A)$:*

$$\lambda = \max_{i \in \{0 \dots n-1\}} \max_{l \in \mathbb{N}_n} \frac{A^l_{i,i}}{l} \tag{4}$$

Démonstration :

- Existence d'un couple propre : Soit λ défini comme ci-dessus, et $B = \lambda^{-1} A$. Le poids maximum des circuits de B est donc nul. B^* et B^+ existent donc, et il existe un i_0 tel que $B_{i_0 i_0} = 0$. Pour cet i_0, on a, puisque pour $i \neq i_0$, $I_{i,i_0} = \varepsilon$, $B^+_{.,i_0} = B^*_{.,i_0}$. Par suite, $BB^*_{.,i_0} = B^+_{.,i_0} = B^*_{.,i_0} \Rightarrow AB^*_{.,i_0} = \lambda B^*_{.,i_0}$. Donc $(\lambda, B^*_{.,i_0})$ est un couple propre de la matrice A.

- Pour tout vecteur propre x, aucune de ses coordonnées n'est égale à ε. En effet, il existe nécessairement une coordonnée x_{i_0} non nulle. Or pour tout j, il existe n tel que $A^n_{j,i_0} \neq \varepsilon$ (A étant irréductible). Mais alors, $x_j \geq \lambda^n x_{i_0} > \varepsilon$.

- Si λ est une valeur propre, alors c'est le poids moyen d'un cycle : il existe un x_{i_1} non nul, et donc un x_{i_2} tel que $A_{i_1,i_2} x_{i_2} = \lambda x_{i_1}$. x_{i_2} est non nul, et donc on peut itérer ce processus : pour tout j, il existe i_{j+1} tel que $A_{i_j,i_{j+1}} x_{i_{j+1}} = \lambda x_{i_j}$. Or on obtient deux indices égaux (i_l par exemple), le nombre de valeurs i_j étant fini. On a donc, entre les indices i_l : $A_{i_l,i_{l+1}} A_{i_{l+1},i_{l+2}} ... A_{i_k,i_l} = A^{k-l+1}_{i_l,i_l} = \lambda^{k-l+1}$. λ est donc le poids moyen d'un circuit.

- Unicité de la valeur propre : Soit $\gamma = (i_1, ..., i_l)$ un circuit ($i_1 = i_l$) du graphe $\mathcal{G}(A)$, λ une valeur propre et x un vecteur propre. $\forall j \in \mathbb{N}_l$, on a $A_{i_{j+1},i_j} x_{i_j} \leq \lambda x_{i_{j+1}}$, et donc, $A^l x_{i_1} \leq \lambda^l x_{i_1}$ et le poids moyen de γ est inférieur à λ. Le point précédent permet de conclure.

□

Définition 6. *Soit A une matrice et $\mathcal{G}$ son graphe associé. Un circuit est dit critique s'il est de poids moyen maximal. Le graphe critique associé à $\mathcal{G}$ est le sous-graphe de $\mathcal{G}$ formé de ses circuits critiques. La matrice critique de A est obtenue en remplaçant $A_{i,j}$ par ε si l'arête (i, j) n'appartient pas au graphe critique.*

Définition 7. *On définit la cyclicité par la formule :*

$$c = ppcm_{\mathcal{C} \in cfc_{crit}} pgcd_{a \in \mathcal{C}} l(a) \tag{5}$$

cfc_{crit} est l'ensemble des composantes fortement connexes du graphe critique, a parcourt les circuits, et $l(a)$ représente la longueur de a.

Le théorème suivant va être très utile par la suite :

Théorème 3. *Soit A une matrice irréductible, c sa cyclicité, et λ son unique valeur propre. Il existe alors un entier naturel n_0 appelé temps de couplage tel que :*

$$\forall n \geq n_0 \quad A^{n+c} = \lambda^c \otimes A^n \tag{6}$$

Nous démontrerons une propriété plus précise en Section 6.

3.3 Modélisation des graphes d'événements

On cherche maintenant à représenter l'évolution des graphes d'évenements par des équations dans l'algèbre (Max,+). On suppose en outre qu'à l'instant initial il y a au plus 1 jeton dans les places (en ajoutant des transitions et des places supplémentaires, on peut se ramener à ce cas).

3.3.1 Cas des systèmes isolés

On dispose d'un graphe d'événement temporisé $\mathcal{G}$ de taille n dont les durées de franchissement des transitions sont représentées par le vecteur σ, et le marquage dans les jetons est supposé être 0 ou 1 à l'instant initial. $X_q(k)$ est le temps de k-ième tir dans la transition q. C'est l'évolution de X que l'on veut trouver.

Définissons les matrices $A(k)$ pour $k \in \{0, 1\}$:

$$A(k)_{i,j} = \begin{cases} \sigma_j & \text{si il existe une place entre } i \text{ et } j \text{ ayant } k \text{ jetons à l'instant initial} \\ \varepsilon & \text{sinon.} \end{cases} \tag{7}$$

$A(0)$ représente le sous-graphe de $\mathcal{G}$, où l'on garde uniquement les places sans jeton initialement, et $A(1)$ le sous-graphe où l'on garde uniquement les places avec jeton initialement. Dans une transition, le temps de n-ième tir est obtenu en calculant le maximum des temps de n-ième tir plus le temps de franchissement des transitions précédant la transition donnée, pour les places n'ayant pas de jetons au temps 0, et du $n-1$-ième temps de tir plus le temps de franchissement pour celles qui ont un jeton à l'instant initial.

$$X_q(n+1) = \bigoplus_{i \in \mathbb{N}_N} A_{q,i}(0) X_i(n+1) \oplus \bigoplus_{i \in \mathbb{N}_N} A_{q,i}(1) X_i(n) \tag{8}$$

donc, on a :

$$X(n+1) = A(0)X(n+1) \oplus A(1)X(n) \tag{9}$$

Soit $A = A(0)^*A(1)$. Cette matrice correspond au graphe $\mathcal{G}$ dans lequel toutes les places sans jeton à l'instant initial ont été court-circuitées. Le graphe est modifié, mais son évolution reste inchangée : on remplace des chemins passant par une place avec un jeton, puis sans jeton par une transition du poids total. Les poids des circuits restent inchangés. Les temps de tir dans les transitions restent les mêmes. L'équation 9 donne alors :

$$X(n+1) = A \otimes X(n) \tag{10}$$

3.3.2 Systèmes non isolés

On suppose l'existence d'une source qui envoie des jetons aux dates $(u(n))$ dans des transitions de $\mathcal{C}$. Soit B le vecteur défini par : $B_i = 0$ si la transition i est reliée à la place source, et ε sinon. Alors l'évolution du système sera :

$$X(n+1) = A \otimes X(n) \oplus B \otimes u(n) \tag{11}$$

4 Formulation algébrique des problèmes à résoudre

On considère d'abord le cas de deux systèmes, puis on fera une généralisation à N systèmes. On suppose que tous les temps de tir sont des entiers.

Considérons $\mathcal{G}_1$ et $\mathcal{G}_2$ deux graphes, tels que $\mathcal{G}_1 \succ \mathcal{G}_2$.

4.1 Calcul des temps de cycles dans les graphes

Pour faire des calculs dans les graphes, le lemme suivant est très utile :

Lemme 3. *Pour toute composante fortement connexe isolée $\mathcal{C}$ il existe un temps de cycle $\lambda_\mathcal{C}$ réel positif, une cyclicité $s_\mathcal{C}$ entier strictement positif et une période de transition $k_\mathcal{C}$ entier positif ou nul tel qur pour toute transition $q \in \mathcal{C}$ et tou $k \geq k_\mathcal{C}$ on a : $V_q(k + s_\mathcal{C}) = V_q(k) + s_\mathcal{C}\lambda_\mathcal{C}$.*

Ce lemme nous permet de trouver facilement les temps de cycle pour les composantes fortement connexes d'un graphe fermé : si le maximum des temps de cycle des composantes qui précèdent une composante $\mathcal{C}$ est supérieur au temps de cycle de $\mathcal{C}$ prise isolée, alors $\mathcal{C}$ dans le graphe a pour temps ce cycle de maximum. Sinon, le système est dit instable, et le temps de cycle pour cette composante est la même que si elle étaie isolée.

4.2 Modélisation en temps contracté

Activité dans le premier graphe Introduisons une fonction d'activité du premier graphe :

$$S_1(t) = \begin{cases} 1 & \text{s'il y a une transition de } O_1 \text{ active à la date t,} \\ 0 & \text{sinon.} \end{cases} \tag{12}$$

On sait que l'évolution du système est périodique, de période T. On définit alors le cœfficient $\Gamma = \int_0^T 1 - S_1(\tau)\, d\tau$ qui représente la durée d'inaction du graphe 1 pendant une période, et la fonction F de $\mathbb{R}$ dans $\mathbb{R}$ définie par $F(t) = \int_0^t 1 - S_1(\tau)\, d\tau$ qui représente le temps contracté par l'activité de $\mathcal{G}_1$: c'est le temps pendant lequel le graphe $\mathcal{G}_2$ peut être actif. F est pseudo-périodique de période T, d'incrément Γ.

Deuxième graphe en temps contracté Soit $\mathcal{H}$ une copie de $\mathcal{G}_2$ mais non préemptée par $\mathcal{G}_1$. On introduit les suites $\{X_q(n)\}$ et $\{V_q(n)\}$ des temps de tir dans la transition q dans respectivement $\mathcal{G}_2$ et $\mathcal{H}$.

Le dernier théorème appliqué au graphe devient alors :

Lemme 4. *Pour toute composante fortement connexe isolée $\mathcal{C}$ il existe un temps de cycle $\lambda_\mathcal{C}$ réel positif, une cyclicité $s_\mathcal{C}$ entier strictement positif et une période de transition $k_\mathcal{C}$ entier positif ou nul tel que pour toute transition $q \in \mathcal{C}$ et tout $k \geq k_\mathcal{C}$ on a : $V_q(k + s_\mathcal{C}) = V_q(k) + s_\mathcal{C}\lambda_\mathcal{C}$.*

On définit l'évolution du deuxième système en temps contracté : $X'_q(k) = F(X_q(k))$. On vérifie que l'évolution des temps de tir des transitions est toujours pseudo-périodique en temps contracté.

Grâce au lemme précédent, on a le résultat suivant :

Lemme 5. *Pour toute composante fortement connexe $\mathcal{C}$ du graphe $\mathcal{G}_2$ prise séparément, on a :*

- *si $\mathcal{C} \in I_2$, alors pour toute transition $q \in \mathcal{C}$ et tout $k \geq n_\mathcal{C}$, $X'_q(n + s_\mathcal{C}T) = X'_q(n) + s_\mathcal{C}\lambda_\mathcal{C}\Gamma$*
- *si $\mathcal{C} \in O_2$, alors pour toute transition $q \in \mathcal{C}$ et tout $k \geq n_\mathcal{C}$, $X'_q(n + s_\mathcal{C}) = X'_q(n) + s_\mathcal{C}\lambda_\mathcal{C}$*

L'évolution du système $\mathcal{G}_2$ dans O_2 est donc la même en temps contracté que ce qu'elle serait en temps réel sans la préemption de $\mathcal{G}_1$. C'est dans I_2 que celle-ci est visible : l'arrivée des jetons dans les places entre I_2 et O_2 est accélérée. Ceci simplifie les calculs, dans la mesure où, en général, I_2 est une simple horloge. L'horloge est toujours pseudo-périodique, et donc l'évolution globale de $\mathcal{G}_2$ est encore pseudo-périodique.

Nous allons maitenant voir si le système est stable. Pour cela, définissons les quantités $\lambda'_\mathcal{C} = \lambda_\mathcal{C}\Gamma/T$ si $\mathcal{C} \in I_2$ et $\lambda'_\mathcal{C} = \lambda_\mathcal{C}$ si $\mathcal{C} \in O_2$. Deux cas se présentent :

1. $\min_{\mathcal{C}\in I} \lambda'_\mathcal{C} \geq \max_{\mathcal{C}\in O} \lambda'_\mathcal{C}$ et le système est stable. Toutes les transitions atteignent en un temps fini un régime pseudo-périodique tel que $X'_q(n + s_{\mathcal{C}_0}mT) = X'_q(n) + s_{\mathcal{C}_0}m\lambda_{\mathcal{C}_0}\Gamma$ où m est un entier tel que mT et $m\lambda_{\mathcal{C}_0}$ en soient aussi. Le nombre de jetons dans les places est borné. Le processus d'activité devient périodique, en temps réel et en temps contracté.

2. $\min_{\mathcal{C}\in I} \lambda'_{\mathcal{C}} < \max_{\mathcal{C}\in O} \lambda'_{\mathcal{C}}$ et le système est instable. Dans certaines places le nombre de jetons croît vers l'infini.

Exemple 3. *Dans le premier exemple, l'application du premier lemme suffit à montrer que le système n'est pas stable. En effet, la première tâche s'effectue toutes les 20 unités de temps, et impose donc à la seconde de s'effectuer aussi à cette vitesse. Or l'horloge de la deuxième tâche a une période de 15 unités de temps. Donc le nombre de jetons dans la place située dans la seconde tâche, juste avant le point de rendez-vous tend vers l'infini avec le temps.*

Considérons un système formé d'une composante initiale, $\mathcal{C}_1$, et d'une autre composante, $\mathcal{C}_2$. La place de transition entre $\mathcal{C}_1$ et $\mathcal{C}_2$ est notée q_1 (et appartient à $\mathcal{C}_2$).

Introduisons (t_n) la suite des dates d'arrivée d'un jeton en q_1 en temps réel et $\{u(n)\} = \{F(t_n)\}$ celle en temps contracté. $\{u(n)\}$ est pseudo-périodique et $u(n+T) = u(n) + \sigma\Gamma$

On a donc l'équation en temps contracté : $X(n+1) = A \otimes X(n) \oplus B \otimes u(n)$, où A est la matrice associée au graphe $\mathcal{C}_2$ pris isolément, et B le vecteur dont les composantes sont toutes égales à ε, sauf celles reliées à la place q_1 , qui sont nulles.

4.3 Problème P2

Nous avons déjà montré comment savoir si un système est stable globalement ou non. Introduisons tout d'abord de nouvelles notations. On notera désormais T le plus petit multiplicateur commun (ppcm) de la période de u et de celle de l'horloge. Γ est alors modifié. γ est la plus grande valeur propre de A, et σ le temps de cycle de l'horloge.

Nous supposerons désormais que le système est stable ($\sigma\Gamma/T \geq \gamma$). Il nous faut donc maintenant affiner notre analyse afin de savoir si une tâche a été effectuée presque à chaque fois dans le temps qui lui a été imparti, c'est-à-dire à chaque fois, une fois le régime périodique atteint.

Définissons le vecteur $Z(n) = X(n) - u(n+1)$. X et u sont toutes les deux pseudo-périodiques de période T et d'incrément $\sigma\Gamma$ à partir d'un certain rang, donc l'évolution de Z devient elle aussi périodique de période T , et on a :

$$Z(kT+s) = Z((k-1)T + s \; \forall k \geq k_0, \; \forall 0 \leq s \leq T-1 \tag{13}$$

Or $(X(n))_1$ représente la date de n-ième tir dans la transition 1 (celle qui est reliée à la place q_1 et $u(n+1)$ est la date d'arrivée du $n+1$-ième jeton dans cette place. Pour que la propriété P2 soit satisfaite, il faut donc que, pour presque tout n, $Z(n)_1 \leq 0$.

Grâce à la périodicité de Z, il faut et il suffit de montrer que $Z(kT+s) \leq 0$ pour un $k \geq k_0$ et tout $s \in \{0, ..., T-1\}$.

Or,

$$
\begin{aligned}
Z(kT+s) &= X(kT+s) - u(kT+s+1) \\
&= [A \otimes X(kT+s-1) \oplus B \otimes u(kT+s)] - u(kT+s+1) \\
&= A \otimes X(kT+s-1) - u(kT+s) - u(kT+s+1) + u(kT+s) \\
&\quad \oplus B \otimes u(kT+s) - u(kT+s+1) \\
&= -\tau(kT+s) \otimes A \otimes Z(kT+s-1) \oplus -\tau(kT+s) \otimes B \\
&= ... \\
&= \sigma\Gamma \otimes A^{\otimes T} \otimes Z((k-1)T+s) \\
&\quad \bigoplus_{i=0}^{T-1} (-\tau(kT+s) - ... - \tau(kT+s-i)) \otimes A^{\otimes i} \otimes B
\end{aligned}
$$

Posant $M = \sigma\Gamma \otimes A^{\otimes T}$ et $C(s) = \bigoplus_{i=0}^{T-1}(-u(kT+s+1) + u(kT+s-i)) \otimes A^{\otimes i} \otimes B$, on a $Z(kT+s) = M \otimes Z(kT+s) \oplus C(s)$ et donc $Z(kT+s) = M^* \otimes C(s)$.

La propriété P2 est donc vérifiée si et seulement si $M^* \otimes C(s) \leq 0 \ \forall 0 \leq s \leq T-1$.

4.4 Propriété 2 pour toutes les phases initiales

A chaque fois qu'une exécution est lancée, il se peut que ce soit avec une phase initiale différente, c'est-à-dire qu'il y a un décalage temporel entre les deux systèmes à l'instant initial. Pour cette raison, on voudrait s'assurer que la propriété P2 est satisfaite quelque soit l'état de départ du système.

Dans la partie périodique de l'évolution du système, il y a T phases différentes. Pour une phase ϕ donnée, $\phi \in [1...T-1]$, les temps de tir en temps réel sont alors décalés de ϕ.

Soient $u_\phi(n) = F(t_n + \phi)$ la date du n-ième tir en temps contracté, $\tau_(\phi) = u_\phi(n+1) - u_\phi(n)$ et C_ϕ la valeur obtenue en remplaçant τ par τ_ϕ dans la formule donnant C.

Il s'agit alors de savoir si $K = \bigoplus_{\phi=0}^{T-1} \bigoplus_{s=0}^{T-1} M^* \otimes C_\phi(s)$ a sa première coordonnée négative ou nulle. Mais cette formule se calcule avec une complexité en $\mathcal{O}(T^3)$, ce qui est trop. On cherche donc une autre formule.

Remplaçons C_ϕ par sa formule. On a donc :

$$
M^* \otimes C_\phi(s) = \bigoplus_{i=0}^{T-1} (M^* \otimes A^i)_{1,1} \otimes -(u_\phi(kT+s+1) - u_\phi(kT+s-i)) \tag{14}
$$

et alors,

$$
\begin{aligned}
K &= \bigoplus_{\phi=0}^{T-1}\bigoplus_{s=0}^{T-1} M^* \otimes C_\phi(s) \\
&= \bigoplus_{i=0}^{T-1}(M^* \otimes A^i)_{1,1} \bigoplus_{\phi=0}^{T-1}\bigoplus_{s=0}^{T-1} -(u_\phi(kT+s-1) - u_\phi(kT+s+i)) \\
&= \bigoplus_{i=0}^{T-1}(M^* \otimes A^i)_{1,1} \bigoplus_{\phi=0}^{T-1}\bigoplus_{s=0}^{T-1} -(u_\phi(s+i+1) - u_\phi(s))
\end{aligned}
$$

Pour chaque i, cherchons le minimum de $u_\phi(s) - u_\phi(s+i-1)$. Pour ce faire, on choisit s^* une fonction de i telle que pour tout $s \in [0...T]$ on ait $t_{i+1+s^*} - t_{s^*} \leq t_{i+1+s} - t_s$ puis ϕ^* tel que $F(t_{i+1+s^*} + \phi^*) - F(t_{s^*} + \phi^*) \leq F(t_{i+1+s^*} + \phi) - F(t_{s^*} + \phi)$. On obtient bien l'inégalité voulue, et l'égalité est obtenue pour $s = s^*$ et $\phi = \phi^*$.

$$
K = \bigoplus_{i=0}^{T-1}(M^* \otimes A^i)_{1,1} + (u_{\phi^*}(s^*) - u_{\phi^*}(s^*+i+1)) \tag{15}
$$

Pour chaque i, le calcul de s^* et de ϕ^* se fait en $\mathcal{O}(T)$ et donc le calcul complet en $\mathcal{O}(T^2)$.

4.5 Propriété 1

La propriété 2 nous assure que dans la partie périodique les contraintes de temps sont respectées. Ainsi, pour la propriété 1, il suffit de vérifier ces contraintes pour la partie transitoire. Mais au lieu de vérifier à chaque fois si la partie périodique est atteinte, on cherche plutôt à majorer la durée de la période de transition.

A et u sont pseudo-périodiques à partir d'un certain rang. Il faut chercher leur temps de couplage. Pour cela, on regarde à chaque étape si la partie périodique est atteinte. La période de transition est donnée par la formule suivante :

$$
t_0 + \frac{e \otimes (I \oplus A)^{2t_0+p} \otimes (B \oplus (X(1) - u(1)))}{-\gamma/c + \sigma\Gamma/T} \tag{16}
$$

p est le ppcm du temps de cyclicité de u et A, t_0 le maximun de leurs périodes de transition, e est le vecteur ligne formé de 0.

Pour chercher cette période de transition, il faut chercher le temps de couplage de A, mais on peut se demander s'il existe un majorant s'exprimant en fonction de A, de sorte que l'on n'ait plus à calculer les puissances de A. Une fois la partie de programmation terminée, un des objectifs du stage était de trouver un tel majorant. C'est ce que nous verrons en partie 6.

4.6 Exemple

Reprenons le deuxième exemple.

On a les valeurs suivantes : T = 5, $\Gamma = 4$, $\lambda_I = 4$, $\lambda_O = 3$ et $\lambda'_I = 16/5 < 3$. Le système est stable globalement.

Intéressons nous maintenant aux propriétés P1 et P2 :

Remarquons d'abord que le système est périodique : si P2 est vérifiée, alors P1 aussi.

On a en temps contracté :

$$u(k) = 4k + \lceil \frac{4k}{5} \rceil \tag{17}$$

La matrice A est :

$$\begin{bmatrix} \varepsilon & 3 \\ \varepsilon & \varepsilon \end{bmatrix}$$

Pour la transition reliée à l'horloge, on a donc l'évolution :

$$x(k) = 3 \otimes x(k-1) \oplus u(k).$$

et

$$z(k) = x(k) - u(k+1)$$

Supposons que le système est stable jusqu'au rang $k-1$. Alors, on a :

$$z(k) = x(k) - u(k+1) = x(k) - u(k) + u(k) - u(k+1)$$

$u(k+1) - u(k) \geq 3$ et $x(k-1) \leq u(k)$. Donc, puisque $x(k) = 3 \otimes x(k-1) \oplus u(k)$,

- soit $x(k) = u(k)$, auquel cas $z(k) \leq -3$,
- soit $x(k) = 3 + x(k-1)$ et $z(k) \leq 0$.

Donc la propriété 2 est satisfaite au rang k.

La propriété 2 est donc satisfaite.

4.7 Généralisation à un nombre arbitraire de systèmes

On suppose maintenant qu'il y a N systèmes, ordonnés selon leur priorité par l'ordre partiel $\succ$. Pour un système i donné, soit J l'ensemble des systèmes qui le préemtent.

La période des systèmes préemptifs est alors $T = ppcm_{j \in J} T_j$ et le processus d'activité $S(t) = \max_{j \in J} S_j(t)$.

Pour savoir si les propriétés sont vérifiées pour un système donné, il suffit d'effectuer les calculs précédents en remplaçant T_1 et S_1 par les formules ci-dessus.

Ainsi, pour savoir si les contraintes de temps sont vérifiées, on procède par induction sur l'ordre des priorités, pour toutes les tâches.

5 Aspects algorithmiques

5.1 Plan du programme

Le but du stage était de faire un programme permettant de savoir si les systèmes étaient stables, et si les propriétés étaient vérifiées, à partie du graphe d'événement des tâches.

À partir du graphe, on peut facilement trouver la matrice d'adjacence du graphe M ($M_{i,j} = 0$ s'il existe un arc de i à j et ε sinon), la matrice A et le vecteur des durées de tir dans les transitions.

Il a aussi fallu trouver un système pour représenter l'ordre des priorités des différentes tâches. On considère qu'il y a des différents niveaux de préemption : on représente ceux-ci par des entiers. Plus la tâche est prioritaire, plus le niveau est élevé. À chaque transition du graphe, on associe alors un entier qui symbolise le niveau de préemption de la tâche.

Une première partie du programme consiste à trouver et trier dans l'ordre topologique les différentes composantes composantes fortement connexes. Il faut aussi séparer les différents systèmes disjoints (les différentes tâches).

Une fois les systèmes définis, on vérifie la cohérence des préemptions : toutes les transitions d'un même système doivent avoir le même niveau de priorité.

Ainsi, on connaît l'ordre des tâches, et on peut tester les différentes propriétés en prenant les systèmes dans l'ordre décroissant des niveaux de préemption.

On teste d'abord la stabilité des systèmes, indépendamment des préemptions. Pour cela, il faut calculer les temps de cycle des composantes fortement connexes, et les comparer à ceux des précédentes : on calcule la valeur propre de la restriction de A aux composantes fortement connexes.

Ensuite, il faut calculer la fonction d'activité et la période associée aux systèmes prioritaires sur celui qui est traité. On obtient ainsi le coefficient Γ et on peut calculer les temps de cycle en temps contracté, ce qui permet de savoir si le système est globalement stable ou non.

Si le système est stable, on regarde si les autres propriétés sont vérifiées : dans l'ordre, P2, puis P2 pour toutes les phases initiales, et enfin P1.

Enfin, on calcule l'activité du système comme suit : le système est actif si une de ses transitions non initiales est active, ou si un des systèmes plus prioritaires que lui est actif.

On lit les résultats dans un fichier de sortie.

5.2 L'outil ERS

La dernière partie du stage était consacrée à l'intégration de ce programme à l'outil ERS.

ERS est un ensemble d'outils servant à la modélisation de graphes d'événements. Il est conçu comme un système modulaire : il est formé de programmes indépendants, ayant chacun une fonction précise.

Les programmes peuvent communiquer entre eux : les fichiers d'entrée doivent avoir un format précis. L'utilisation de plusieurs modules permet de faire une analyse assez complexe des graphes.

Il y a aussi une interface graphique qui facilite l'entrée des fichiers utilisés par les outils d'analyse.

6 Temps de couplage d'une matrice irréductible dans l'algèbre (Max,+)

Revenons au théorème 3 et à la propriété P1. Le théorème donne l'existence d'un rang k_0 à partit duquel on a la formule :

$$\forall k \geq k_0,\ A^{\otimes(k+c)} = \lambda^{\otimes c} \otimes A^{\otimes k} \tag{18}$$

6.1 Introduction

Aucune indication n'est donné quant à son ordre de grandeur. On cherche ici à trouver un majorant de k_0.

Hypothèses faites sur la matrice A

On considère une matrice $A \in \mathbb{R}_{max}^{N\times N}$, irréductible.

On suppose également que sa valeur propre est 0 : quitte à multiplier la matrice par λ^{-1}, on peut considerer sans perte de généralité que la valeur propre est nulle. On suppose que A est acyclique ($c = 1$). Il suffit de prendre en compte la matrice A^c, et de multiplier le temps trouvé par c.

Soit $\mathcal{G}$ le graphe associé à A. $A^k_{i,j}$ donne les poids maximum des chemins de i à j de longueur k. L'hypothèse faite sur la valeur propre et la cyclicité nous permet de chercher le rang à partir duquel $(A^n_{i,j})_n$ est stationnaire.

6.2 Tous les nœuds de la matrice sont critiques

6.2.1 La matrice critique est irréductible

Lemme 6. *Sous ces hypothèses, pour tous i, j il y a un chemin de poids maximum de i à j qui est un chemin du graphe critique.*

Démonstration : le graphe critique est fortement connexe, donc tout chemin de i à j dans celui-ci est sous chemin d'un circuit critique, donc de poids maximum. En effet, en considérant le chemin de j à i dans le graphe critique, et un chemin de poids supérieur à celui du graphe critique, on obtiendrait un poids strictement positif pour un cycle. □

Théorème 4. *Soient $\mathcal{G}$ un graphe fortement connexe, acyclique à N nœuds et i un nœud du graphe. Supposons que par i passe un cycle de longueur n. Alors, pour tout nœud j de $\mathcal{G}$, il existe un chemin de i à j de longueur k pour tout $k \geq (N-1)n$.*

Démonstration : Posons tout d'abord $S(0) = \{i\}$ et

$$\forall k \geq 1, S(k) = \{j | \text{il existe un chemin de } i \text{ à } j \text{ de longueur } nk.\} \tag{19}$$

Puisque le nœud i appartient à un circuit de longueur n, $S(k+1)$ est la réunion de $S(k)$ et des circuits formés d'un chemin de $S(k)$ et d'un chemin de longueur n ($S'(k)$, qui ne dépend par construction que de $S(k)$). On aura donc :

$$\forall k \in \mathbb{N},\ S(k+1) \supseteq S(k). \tag{20}$$

et même :

$$S(k+1) = S(k) \cup S'(k) \tag{21}$$

S est donc une suite croissante dans les parties de N,qui est fini, récurrente d'ordre 1. Il existe donc un entier k^* tel que S soit strictement croissante jusqu'au rang k^* puis stationnaire.

Or, $\mathcal{G}$ est apériodique, donc il existe un rang à partir duquel il existe des chemins joignant tous nœuds pour toutes les longueurs de chemin. Donc $S(k^*) = \mathbb{N}_N$.

De plus, $k^* \leq N-1$: S est strictement croissante jusqu'au rang k^* et $|S(0)| = 1$. Par conséquent, il existe des chemins de longueur $(N-1)n$ de i à j. Le graphe étant fortement connexe, pour tout nœud j, il existe un nœud l tel que (l, j) soit une arête du graphe. Il y a donc des chemins de longueur $(N-1)n+1$ de i à j, et en itérant, de longueur $m,\ m \geq (N-1)n$. □

Corollaire 1. *Soit $\mathcal{G}$ un graphe fortement connexe, acyclique à N nœuds. Soit n la longueur de son plus petit cycle. Alors pour tous i et j, il existe un chemin joignant i à j de longueur k pour tout $k \geq k_0$ où*

$$k_0 = N + (N-2)n. \tag{22}$$

Démonstration : Le plus court chemin de i à un nœud l d'un cycle de longueur n a au plus $N-n$ arcs. On applique le théorème précédent au nœud l et on trouve $k_0 \leq (N-n) + (N-1)n = N + (N-2)n$. On peut donc prendre cette valeur pour k_0. □

Appliquons ce corollaire au graphe critique. Le lemme 6 permet de ne considérer que les chemins du graphe critique. Il est acyclique, et tous ses chemins sont critiques. Donc à partir du rang $N + (N-2)n$, le graphe critique permet à lui seul de trouver les chemins de poids maximum, et donc de trouver les coefficients des matrices A^k.

Notons aussi que n est le plus petit entier k tel qu'un élément de la diagonale de A^k soit nul.

6.2.2 La matrice critique n'est pas irréductible

Il y a maintenant K composantes fortement connexes dans le graphe critique. Un chemin de poids maximum passe par au plus $K-1$ arcs qui ne sont pas dans le graphe critique. Le reste du chemin se fait donc à l'intérieur des composantes du graphe critique.

Pour chaque composante k, notons N_k son nombre de nœuds, et n_k la longueur de son plus court cycle. À l'aide de la formule précédente, on connaît le rang à partir duquel tout nœud est joignable à tout autre, dans une même composante. donc, on a :

$$k_0 \leq (K-1) + \sum_{k \in K} (N_k - 2) n_k \tag{23}$$

Soit n le maximum des n_k.

$$\begin{aligned} k_0 &\leq (K-1) + \sum_{k \in K} (N_k - 2) n_k \\ &\leq (K-1) + N + (\sum_{k \in K} (N_k - 2)) n \\ &\leq (K-1) + N + (N - 2K) n \end{aligned}$$

On peut donc prendre $k_0 = (K-1) + N + (N-2K)n$

6.3 Tous les nœuds ne sont pas critiques

Le poids maximum pour un trajet sera atteint lorsque le chemin aura tous ses circuits critiques. Mais il se peut qu'il y ait un circuit de poids moyen très proche de 0 par lequel passent les chemins de poids maximum pour certaines longueurs.

Exemple 4. *On a* $A^n_{k,j} = -n\varepsilon$ *jusqu'à ce que* $n\varepsilon > 10000$ *et donc jusqu'au rang* $10000/\varepsilon$

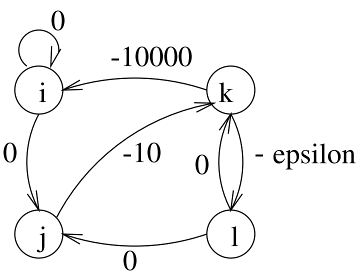

FIG. 4: *Graphe dont tous les nœuds ne sont pas critiques.*

Soit A' la matrice dans laquelle on a remplacé tous les coefficients par ε si une des deux coordonnées représentait un nœud critique. On ne garde donc que les chemins ne passant par aucun nœud critique. Soit B la matrice définie par : $\bigoplus_{k=1}^{N} \frac{A'^k}{k}$.Cette matrice donne pour tout nœud le poids moyen maximal des circuits ne contenant aucun nœud critique y passant. On pose pour tou nœud i non critique $\varepsilon_i = B_{i,i}$

On divise l'ensemble des nœuds en 2 :

- $\mathcal{C}$ l'ensemble des nœuds critiques,
- $\mathcal{C}'$ les autres.

Fixons i et j.

Calculons tout d'abord le rang à partir duquel on peut trouver un chemin critique pour toutes longueurs. Il faut en fait trouver un chemin de poids maximal qui passe par au moins un nœud critique. Supposons qu'il y a N_K nœuds critiques. La longueur maximale d'un chemin passant par un nœud critique est donc $(N - N_K) + (N - 1)$. En modifiant légèrement la formi=ule trouvée précédemment, on trouve un rang $k_0 = (N - N_K) + (N - 1) + N_K + (N_K - 2K)n = 2N - 1 + (N_K - 2K)n$.

Il reste maintenant à trouver la longueur à partir de laquelle le chemin critique est maximal.

Posons $P^k_{i,j} = A^*_{i,k} \otimes A^*_{k,j}$ et $P_{i,j} = \max_{k \in \mathcal{C}} A^*_{i,k} \otimes A^*_{k,j}$. $P^k_{i,j}$ est le chemin de poids maximum de i à j passant par k et $P_{i,j}$ est le chemin de poids maximum de i à j passant par au moins un nœud critique.

À partir de la longueur $\max(\frac{(P^k_{i,j} - P_{i,j})^+}{-\varepsilon_k} + l_{i,j}, k - 0)$ ($l_{i,j}$ est la longueur du chemin simple de poids maximum passant par un nœud critique), les chemins de poids maximum ne passeront pas par k.

Mais alors, tous les circuits des chemins de poids maximum pour une certaine longueur seront critiques à partir de :

$$\max(\max_{k \in \mathcal{C}'} \frac{(P^k_{i,j} - P_{i,j})^+}{-\varepsilon_k} + l_{i,j}, k_0) \tag{24}$$

Enfin, on majore sur i et j. Soit k_1 la longueur minimale pour que les seuls circuits par lesquels passent les chemins soient critiques. On a aussi : $l_{i,j} \leq 2(N - 1)$.

$$k_1 \leq \max(\max_{i \in \mathbb{N}_N} \max_{j \in \mathbb{N}_N} \max_{k' \in \mathcal{C}'} \frac{(P^{k'}_{i,j} - P_{i,j})^+}{-\varepsilon_{k'}} + 2(N - 1), k_0) \tag{25}$$

On peut simplifier cette formule :

Prenons le maximum de ε_k pour les nœuds non critiques, et notons le ε.

On a alors :

$$k_1 \leq \max(\max_{i,j \in \mathbb{N}_N} \max_{k' \in \mathcal{C}'} \frac{(P^{k'}_{i,j} - P_{i,j})^+}{-\varepsilon} + 2(N - 1), k_0) \tag{26}$$

En fait, pour $P^{k'}_{i,j}$, on peut ne prendre en compte que les chemins ne passant pas aucun nœud critique, et donc ces coefficients ne dépendent que de A'. On a donc en prenant la norme du maximum $|.|$:

$$\begin{aligned}
k_1 &\leq \max(\frac{1}{-\varepsilon}(|A'^*| - |P^*|)^+ + 2(N - 1), k_0) && (27)\\
&\leq \max(\frac{1}{-\varepsilon}(|A'^*| - |P^*|)^+ + 2(N - 1), 2N - 1 + (N_K - 2K)n) && (28)\\
&\leq 2N - 1 + \max(\frac{1}{-\varepsilon}(|A'^*| - |P^*|)^+ +, (N_K - 2K)n) && (29)
\end{aligned}$$

7 Conclusion

Grâce à un outil algébrique, on a pu automatiser des calculs et faire une analyse de la stabilité de tâches ordonnancées par une système de préemption et de synchronisation.

Ce stage m'a permis d'approfondir ma connaissance du langage C, et m'a fait découvrir un laboratoire de recherche.

Je tiens à remercier tous les membres de l'équipe TRIO pour leur disponibilité et leur sympathie.

Références

[1] F. Baccelli, B. Gaujal, D. Simon. Analysis of preemptive periodic real time systems using the (max,plus) algebra. Rapport INRIA, 1999.

[2] F. Baccelli, G. Cohen, G. J. Olsder,J.-P. Quadrat. Sychronization and Linearity.Wiley, 1992.

[3] Eric V. Denario. Periods of connected networks and powers if nonnegative matrices. Mathematics of Operation Research, Vol. 2, No. 1, 1977.

Table des matières