

S. HADDAD

P. MOREAUX

G. CHIOLA

**Distributions de Cox et Hase-type dans les réseaux de Petri stochastiques : une méthode efficace de résolution**

*RAIRO. Recherche opérationnelle*, tome 32, n° 3 (1998), p. 289-323

[http://www.numdam.org/item?id=RO\\_1998\\_\\_32\\_3\\_289\\_0](http://www.numdam.org/item?id=RO_1998__32_3_289_0)

© AFCET, 1998, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

## DISTRIBUTIONS DE COX ET PHASE-TYPE DANS LES RÉSEAUX DE PETRI STOCHASTIQUES : UNE MÉTHODE EFFICACE DE RÉOLUTION

par S. HADDAD <sup>(1)</sup>, P. MOREAUX <sup>(1, 2)</sup> et G. CHIOLA <sup>(3)</sup>

---

**Résumé.** – *Nous étudions l'introduction des transitions avec durée de tir à distributions de Cox ou Phase-type dans les réseaux de Petri stochastiques généralisés. Ces transitions génèrent des accroissements importants en espace mémoire et en temps de calcul des probabilités à l'équilibre de la chaîne de Markov sous-jacente. Nous proposons une nouvelle approche pour limiter ce phénomène en conservant toute la sémantique stochastique des travaux antérieurs sur le sujet. Cette méthode est fondée sur une décomposition structurelle du réseau. Nous établissons les conditions sous lesquelles cette décomposition conduit à une expression tensorielle du générateur de la chaîne. Cette expression tensorielle est utilisée pour résoudre la chaîne par une méthode itérative.* © Elsevier, Paris

**Mots clés :** Distributions de Cox, distributions Phase-type, réseaux de Petri stochastiques, algèbre tensorielle.

**Abstract.** – *We study the introduction of transitions with Coxian or Phase-type distribution firing time in Generalized Stochastic Petri nets (GSPNs). Such transitions produce large increases of both space and time complexity for the computation of the steady state probabilities of the underlying Markov chain. We propose a new approach to limit this phenomenon while keeping full stochastic semantics of previous works. The method is based on a structural decomposition of the net. We establish conditions under which this decomposition leads to a tensor expression of the generator of the chain. The tensor expression is used to solve the chain with an iterative method.* © Elsevier, Paris

**Keywords:** Coxian distributions, Phase-type distributions, Stochastic Petri Nets, Tensor algebra.

---

<sup>(1)</sup> LAMSADE – URA CNRS 825, Université Paris Dauphine, place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France, E-mail: {haddad, moreaux}@lamsade.dauphine.fr.

<sup>(2)</sup> LERI-RS, Université de Reims Champagne-Ardenne, BP 1039 – 51687 Reims Cedex 2, France.

<sup>(3)</sup> DISI, Università di Genova, via Dodecaneso 35, 16146 Genova, Italy, E-mail: chiola@disi.unige.it.

**TABLE DES MATIÈRES**

<b>INTRODUCTION</b> .....	291
<b>1. TRANSITIONS-COX ET PHASE-TYPE DANS LES RÉSEAUX DE PETRI STOCHASTIQUES</b> .....	292
1.1. Distributions de Cox .....	292
1.2. Sémantique stochastique des réseaux de Petri stochastiques .....	293
1.2.1. La politique de mémoire .....	293
1.2.2. La sémantique du service .....	294
1.2.3. La politique d'interruption/reprise .....	295
1.2.4. Complexité du modèle stochastique .....	296
1.3. Travaux antérieurs .....	296
<b>2. DÉFINITION DES RÉSEAUX DE PETRI STOCHASTIQUES GÉNÉRALISÉS À TRANSITIONS-COX</b> .....	297
2.1. GSPNs à transitions-Cox .....	297
2.2. Exemple .....	298
2.3. Graphe d'accessibilité d'un Coxian-GSPN .....	299
2.4. Chaîne de Markov d'un Coxian-GSPN .....	300
2.4.1. Descripteurs pour politiques à clients ordonnés .....	301
2.4.2. Descripteurs pour politiques à clients banalisés .....	301
2.4.3. Descripteurs pour politiques à serveur unique .....	302
2.5. Transitions de la chaîne de Markov .....	303
<b>3. DÉCOMPOSITION STRUCTURELLE DES COXIAN-GSPNs</b> .....	305
3.1. Graphes d'accessibilité locaux étendus .....	309
3.2. Expression tensorielle du générateur .....	310
3.2.1. Calcul des $Q_i^0$ .....	312
3.2.2. Calcul des $C_j(t)$ .....	314
<b>4. ÉVALUATION DE LA MÉTHODE</b> .....	314
4.1. Politique serveur-unique .....	314
4.2. Politique à clients ordonnés .....	315
<b>5. INTRODUCTION DES TRANSITIONS IMMÉDIATES</b> .....	316
5.1. Sémantique stochastique des transitions-Cox immédiates .....	316
5.2. Décomposition du réseau .....	317
5.3. Résolution de la chaîne de Markov .....	318
5.3.1. Calcul des $\overline{TRG}_i$ .....	318
5.3.2. Expression tensorielle du générateur .....	320
<b>CONCLUSIONS ET PERSPECTIVES</b> .....	321

## INTRODUCTION

Les réseaux de Petri stochastiques généralisés (GSPNs) (voir par exemple [5] pour une introduction détaillée) comportent deux types de transitions, immédiates et exponentielles. Bien que ceci permette de modéliser une gamme étendue de systèmes, l'emploi de durées de tir à distributions plus générales reste nécessaire pour de nombreux phénomènes réels. Deux directions ont été suivies dans ce but : la première [6, 11] introduit des distributions générales mais sous des conditions spécifiques, qui permettent d'extraire des chaînes de Markov dites subordonnées, et d'en déduire les valeurs de probabilités transitoires et à l'équilibre. La seconde approche [19, 13, 8] introduit des durées de tir à loi de Cox ou Phase-type sans autre restriction, et calcule la chaîne de Markov à temps continu (CTMC) incluse. Ceci exige de définir précisément la sémantique stochastique de ces nouvelles transitions et de gérer la complexité « bi-dimensionnelle » qu'elles induisent : exponentielle par rapport à la taille du réseau et exponentielle (sous des hypothèses spécifiques) par rapport au nombre d'étages des distributions de Cox ou Phase-type. Dans le contexte de cette dernière approche, les travaux antérieurs ont pris en compte les problèmes de sémantique stochastique, soit par des GSPNs standard, soit *via* des CTMCs augmentées. Malheureusement, aucune solution au problème de l'accroissement de complexité n'a été proposée.

Notre approche prend en compte non seulement la sémantique stochastique en autorisant les situations les plus générales, mais encore l'accroissement de complexité de la chaîne de Markov résultante. Nous exploitons la structure de concurrence (propriété qualitative) des réseaux de Petri pour obtenir une expression tensorielle du générateur de la CTMC sous-jacente. Pour autant que l'impact des transitions à loi de Cox ou Phase-type demeure local du point de vue qualitatif, l'accroissement de complexité reste « gérable » : des exemples caractéristiques montrent que l'on passe d'une complexité exponentielle à une complexité polynomiale.

L'organisation de l'article est la suivante : dans la section 1 nous examinons les problèmes que soulève l'introduction des transitions à durée de tir avec distributions de Cox (transitions-Cox) ou Phase-type (transitions-Phase-type) dans les GSPNs. Nous définissons ensuite formellement la syntaxe et la sémantique de telles transitions dans la section 2. Dans la section 3 nous expliquons comment décomposer le réseau pour obtenir une expression du générateur de sa CTMC sous-jacente. Nous étudions les effets de notre méthode sur la complexité de calcul dans la section 4 et nous

introduisons les transitions immédiates dans la section 5, avant de conclure sur l'évolution possible de ce travail.

## 1. TRANSITIONS-COX ET PHASE-TYPE DANS LES RÉSEAUX DE PETRI STOCHASTIQUES

### 1.1 Distributions de Cox

Les distributions de Cox [12] généralisent les distributions exponentielles en série (lois d'Erlang) et en parallèle (lois hyper-exponentielles). Elles sont constituées d'une succession d'étages exponentiels (*fig. 1*) :  $\alpha_0$  est la probabilité de quitter immédiatement le service ; en fin de service de l'étage  $i$ , on peut quitter définitivement le service avec une probabilité  $\alpha_i$  ou poursuivre à l'étage suivant (si  $i < s$ ) avec une probabilité  $\beta_i = 1 - \alpha_i$ . La durée moyenne du service  $i$  est  $\frac{1}{\mu_i}$ .

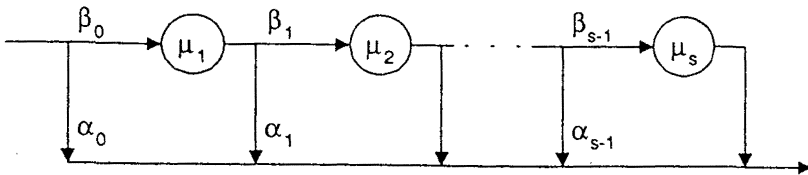


Figure 1. - Modèle d'une distribution de Cox.

L'intérêt des distributions de Cox est double. D'une part, de nombreux phénomènes peuvent être modélisés comme des suites d'étages exponentiels : on parle alors d'approximation par construction de la loi du phénomène. Par exemple, considérons une machine d'une ligne de production dont la durée de traitement proprement dit est de loi exponentielle de moyenne  $\mu_1$  secondes et telle que 10% des pièces traitées (choisies au hasard) subissent un contrôle de qualité dont la durée suit également une loi exponentielle de moyenne  $\mu_2$ . L'étage de fabrication complet est alors modélisé par un traitement de loi de Cox à deux étages où  $\beta_1 = 0,1$ . D'autre part, on peut prendre une distribution de Cox comme approximation de la distribution du phénomène étudié. Il s'agit alors d'identifier les paramètres de l'approximation.

Les lois Phase-type [20] constituent une famille plus générale de distributions pouvant approcher (selon différents critères) une distribution quelconque. Une loi Phase-type est la distribution du temps d'arrivée dans l'unique état absorbant d'une CTMC à  $s + 1$  états. On peut aussi l'interpréter comme un réseau de serveurs exponentiels (*fig. 2*), la distribution initiale de la chaîne étant les  $(C_{0,j})_{1 \leq j \leq s}$  et l'état absorbant étant l'état 0.

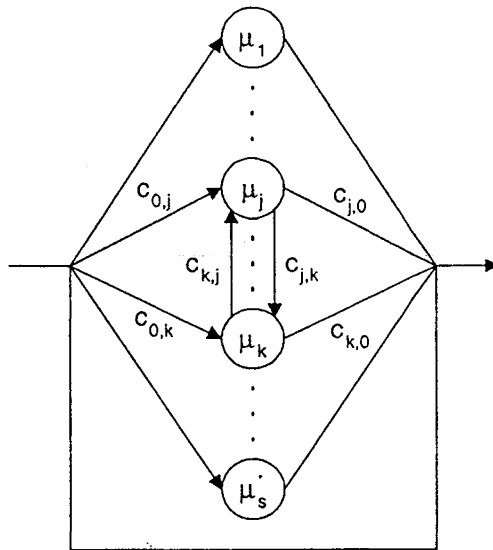


Figure 2. – Modèle d'une distribution Phase-type.

Dans la suite, nous nous restreignons aux lois de Cox. L'application de notre méthode aux transitions à loi Phase-type suit une démarche analogue et ne présente pas de difficulté théorique supplémentaire. Elle nécessite cependant l'adaptation des descripteurs (section 2) et un examen précis des nouvelles transitions possibles entre états de la chaîne de Markov.

## 1.2. Sémantique stochastique des réseaux de Petri stochastiques

Rappelons [2, 1] que la *politique de course* (race policy) choisit comme transition à tirer parmi les transitions franchissables, celle qui possède la plus petite durée de tir restante. Pour les réseaux de Petri stochastiques (SPNs), deux éléments principaux [1] définissent la sémantique stochastique, aussi appelée *politique d'exécution* du modèle; pour les distributions de Cox, un troisième élément doit être ajouté.

### 1.2.1. La politique de mémoire

Puisque les lois de Cox ne sont pas « sans mémoire », on doit indiquer ce qu'il advient du travail déjà réalisé par une transition-Cox lorsqu'une transition est franchie.

Il existe trois « politiques de mémoire » (le terme « politique d'interactions de services » serait plus approprié mais nous nous en tiendrons dans la suite de cet article aux dénominations anglo-saxonnes ou dérivées, en usage

dans la communauté scientifique), qui correspondent aux situations les plus classiques :

**Resampling (interruption)** le temps déjà passé est perdu et un échantillon sera recalculé lorsque la transition sera à nouveau franchissable. Ceci permet de modéliser des activités en conflit dont le changement d'état de l'une affecte celui des autres, comme une panne de serveur vis-à-vis des travaux en cours d'exécution ;

**Enabling memory (poursuite conditionnelle)** le temps passé est conservé *si la transition reste franchissable*. On modélise ainsi les transitions des chiens de garde (time-out) par exemple : les activités sont arrêtées <sup>(1)</sup> lorsque le chien de garde expire, mais le chien de garde est réinitialisé lorsque, après avoir été inhibé (sa transition n'est plus franchissable) il est de nouveau activé (sa transition est franchissable) ;

**Age memory (poursuite inconditionnelle)** le temps déjà passé est conservé, quelque soit l'évolution du système : la prochaine fois que la transition sera franchissable, c'est le temps *résiduel* et non un nouvel échantillon, qui sera utilisé comme délai de franchissement éventuel. Avec cette politique, on modélise des systèmes en temps partagé, l'allocation de processeur à une tâche  $y$  ne faisant pas perdre le travail déjà effectué par une tâche  $x$  qui n'a plus de processeur.

En pratique, la politique Resampling n'a de signification que si les transitions concernées sont en conflit [2], et dans ce cas elle peut être modélisée par une politique Enabling memory et des transitions immédiates.

### 1.2.2. La sémantique du service

Elle indique comment la transition (le service) se comporte lorsque le degré de franchissabilité de la transition (ED) est supérieur à 1 (plusieurs clients en attente) : pour une transition  $t$  à *loi exponentielle*, grâce à sa propriété d'être sans mémoire, c'est sa vitesse dans le marquage  $\mathbf{m}$  qui rend compte de cette sémantique. Trois modes sont fréquemment employés :

**Single server (Serveur-unique)** un seul client est en service à chaque instant : la vitesse de  $t$  est sa vitesse « nominale »  $\theta(t)$ . C'est la sémantique par défaut dans les GSPNs ;

**Multiple ( $K$ ) servers (K-serveurs)** au plus  $K$  clients peuvent être servis simultanément : la vitesse de  $t$  en  $\mathbf{m}$  est  $\min(K, ED(t, \mathbf{m})) \cdot \theta(t)$  ;

---

<sup>(1)</sup> La politique de mémoire pour les activités interrompues peut être Resampling ou non.

**Infinite-servers (Serveurs-infinis)** autant de clients que l'on veut peuvent être servis en même temps : la vitesse de  $t$  en  $\mathbf{m}$  est  $ED(t, \mathbf{m}).\theta(t)$ .

Il est clair qu'une politique de service autre que serveur-unique, oblige à préciser la sémantique des politiques de mémoires et d'interruption/reprise.

Notons également que si la politique de service peut être modélisée par la vitesse effective pour une transition exponentielle, il n'en est plus de même pour une transition-Cox : on doit *mémoriser les étages dans lesquels les clients sont servis*.

### 1.2.3. La politique d'interruption/reprise

Si plusieurs clients sont en service dans les étages d'une loi de Cox, il faut préciser lequel est interrompu si nécessaire. On peut envisager au moins les deux choix suivants : le client le plus récent (noté  $F$  en raison de sa similitude avec la stratégie « FIFO » (Premier entré, premier sorti)) ou le client le moins avancé dans les étages de la loi (noté  $L$ ), est interrompu. Si, de plus, la politique de mémoire est Age memory, il faut définir la stratégie de reprise : quel est le client ré-introduit parmi ceux interrompus ? Ce choix doit être cohérent avec le précédent (exemples : reprise du plus ancien, reprise du plus avancé). Notons que pour une extension aux lois Phase-type, les stratégies d'interruption/reprise doivent être affinées puisque la notion d'ordre entre les étages (au sens de la progression des clients) n'a plus de signification.

Nous résumons les différentes politiques d'exécution possibles dans le tableau I : chaque ligne donne les trois paramètres nécessaires ( $sp(t)$ ,  $mp(t)$  et  $ip(t)$ ) pour chaque transition-Cox.

TABLEAU I  
Tableau des politiques d'exécution.

Politique de service $sp(t)$	Politique de mémoire $mp(t)$	Politique d'interruption $ip(t)$
Single Server ( $S$ )	Enabling ( $E$ )	-
	Age ( $A$ )	
K-Servers ( $K$ ) ou Infinite Servers ( $I$ )	Enabling ( $E$ )	dernier engagé ( $F$ )
	Age ( $A$ )	moins avancé ( $L$ )
		dernier engagé ( $F$ )
		moins avancé ( $L$ )



#### 1.2.4. Complexité du modèle stochastique

La taille du modèle markovien des GSPNs est approximativement la même que celle du générateur du réseau, c'est-à-dire exponentielle par rapport aux cardinaux des ensembles de places et transitions. Avec l'introduction des distributions de Cox et de leurs différentes sémantiques, la taille de la nouvelle CTMC est maintenant le *produit* de la taille du modèle précédent par les cardinaux de *chacun* des espaces d'états des transitions-Cox. Pour les politiques Multiple ou Infinite Servers (et plus encore avec la politique Age Memory, et la stratégie FIFO) ceci accroît considérablement la taille du modèle, ce qui conduit à trouver une méthode efficace pour sa résolution.

### 1.3. Travaux antérieurs

Depuis le début des travaux sur les réseaux de Petri stochastiques, diverses recherches ont essayé de contourner la restriction exponentielle en introduisant des lois plus générales. Deux grandes directions sont possibles.

1. On introduit des lois générales et on étudie le processus stochastique sous-jacent [11]. Si deux transitions à loi générale sont franchissables en un même marquage et que le franchissement de l'un ne réinitialise pas le temps de service de l'autre (politique Age memory ou loi déterministe par exemple le processus n'est plus markovien. Sous certaines hypothèses [15, 10] on peut cependant définir des *points de régénération* du processus qui est alors semi-régénératif. La résolution à l'équilibre est fondée sur la mise en évidence d'une *chaîne de Markov incluse* et sur l'étude du processus entre deux points de régénérations.
2. On se restreint aux réseaux dont les processus stochastiques sous-jacents sont des CTMCs.

Dans le deuxième cas, qui est le cadre de ce travail, deux méthodes ont été proposées.

La méthode de transformation du GSPN initial [19, 8] : chaque transition-Cox est remplacée dans le réseau par un sous-réseau qui traduit sa sémantique stochastique. Dans [8] les auteurs proposent des réseaux de remplacement d'une telle transition possédant un nombre fini de places et transitions quelque soit le nombre d'étages de la loi de Cox contrairement à [19] : par exemple une transition-Cox avec politique Age memory, serveur-unique est expansée en un sous-réseau à 8 places et 7 transitions. La procédure de remplacement peut être automatisée et intégrée dans un outil. Notons cependant que seule la politique de service serveur-unique est étudiée. De plus chaque transition-Cox doit posséder une seule place d'entrée et une seule place de sortie.

La méthode d'expansion de la chaîne de Markov incluse [13] : l'auteur traite le cas des lois Phase-type mais le principe est inchangé pour des lois de Cox. Le réseau n'est ici pas modifié ; on indique simplement les propriétés des transitions-Cox. C'est au moment du calcul du graphe d'accessibilité que l'existence de ces lois est prise en compte. D'une part chaque marquage contient, en plus des informations habituelles, toutes les informations nécessaires sur l'état courant de chaque transition-Cox (clients en service, leurs étages). D'autre part, lors du calcul des franchissements à partir d'un marquage, on met à jour ces informations en tenant compte de la sémantique stochastique des transitions. Cette méthode permet de gérer toutes les combinaisons des trois caractéristiques sémantiques que nous avons indiquées ci-dessus. Là encore, il est possible d'intégrer la méthode dans un outil (au niveau du solveur et non plus à celui d'une étape précédant le solveur).

## 2. DÉFINITION DES RÉSEAUX DE PETRI STOCHASTIQUES GÉNÉRALISÉS À TRANSITIONS-COX

Dans cette section, nous donnons la définition formelle d'un réseau de Petri stochastique généralisé à transition-Cox et son interprétation stochastique, c'est-à-dire sa chaîne de Markov sous-jacente.

### 2.1. GSPNs à transitions-Cox

DEFINITION 2.1 : *Un réseau de Petri stochastique généralisé à transition-Cox (Coxian-GSPN) est un tiple  $\mathcal{N} = (\mathcal{N}', \theta, \theta_C)$  où :*

- $\mathcal{N}' = (P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{Inh}, \mathbf{pri})$  est un réseau de Petri :
  - $P$  est l'ensemble des places ;
  - $T$  est l'ensemble des transitions ;
  - $\mathbf{Pre}, \mathbf{Post}$  et  $\mathbf{Inh} : P \times T \rightarrow \mathbb{N}$  sont les fonctions d'incidence :  $\mathbf{Pre}$  est la fonction d'incidence avant,  $\mathbf{Post}$  la fonction d'incidence arrière et  $\mathbf{Inh}$  la fonction d'inhibition ;
  - $\mathbf{pri} : T \rightarrow \mathbb{N}$  est la fonction de priorité.
- et  $(^2) T = T_E \uplus T_I \uplus T_C$  où  $T_E, T_I$  et  $T_C$  sont les ensembles de transitions exponentielles, immédiates et Cox.
- $\theta$  et  $\theta_C$  définissent les propriétés stochastiques des transitions :

---

(<sup>2</sup>)  $A \uplus B$  signifie  $A \cup B$  avec  $A \cap B = \emptyset$ .

- $\theta : T \times \text{Bag}(P)^{(3)} \rightarrow \mathbb{R}^+$ .  $\theta(t, \mathbf{m})$  est la vitesse (si  $t$  est exponentielle) ou le poids (si  $t$  est immédiate ou Cox) de  $t$  en  $\mathbf{m}$ .
- $\theta_C = (s, \mu, \alpha, mp, sp, ip)$  est définie uniquement pour les transitions-Cox. Elle donne pour chaque transition  $t$  toutes les informations nécessaires pour décrire de manière non ambiguë, sa sémantique stochastique :
  - $s(t)$  : le nombre d'étages de  $t$ ;
  - $\alpha_0(t)$  : la probabilité de la fin de service immédiate de  $t$ ;
  - $\forall 1 \leq i \leq s(t)$ ;
  - $\mu_i(t)$  : la vitesse de service de l'étage  $i$  de  $t$ ;
  - $\alpha_i(t)$  : la probabilité de fin de service de  $t$  après la fin de service de l'étage  $i$  ( $\alpha_{s(t)} = 1$ );
  - $mp(t)$  : la politique de mémoire de  $t$  ( $E$  ou  $A$ );
  - $sp(t)$  : la politique de service de  $t$  ( $S, K$  ou  $I$ );
  - $ip(t)$  : la politique d'interruption de  $t$ , si nécessaire ( $F$  ou  $L$ ).

La syntaxe et la sémantique des composants standard des GSPNs sont inchangées. Pour  $0 \leq i \leq s(t)$ , nous notons  $\beta_i(t) = 1 - \alpha_i(t)$  la probabilité d'entrer dans l'étage  $(i+1)$  après la fin de service de l'étage  $i$  de  $t$  ( $\beta_{s(t)} = 0$ ).

Le choix d'un marquage initial  $\mathbf{m}_0$  définit un Coxian-GSPN marqué aussi appelé *Coxian-GSPN system*  $\mathcal{S} = (\mathcal{N}, \mathbf{m}_0)$ .

Comme nous traitons ici l'introduction des transitions-Cox, nous nous restreignons dans cette section aux Coxian-GSPNs sans transition immédiate et aux transitions-Cox sans franchissement immédiate (i.e.  $\alpha_0 = 0$ ). Le cas général sera étudié dans la section 5 où l'on discutera aussi de la signification exacte de **pri** et  $\theta$  pour les transitions-Cox.

## 2.2. Exemple

Pour illustrer notre propos, nous présentons un exemple élémentaire de Coxian-GSPN, extension d'un exemple présenté dans [1], que nous emploierons tout au long de ce travail. Le réseau  $\mathcal{S}_1$  de la figure 3 modélise un système où des activités (assemblage, etc.) sont effectuées à l'aide de l'une ou l'autre de deux lignes de machines (transitions  $t_1, t_2$ ) mises en fonctionnement dès la présence de pièces à traiter (place  $p_1$ ). Si la pièce est traitée par une ligne, le travail de préparation déjà réalisé par l'autre est

<sup>(3)</sup>  $\text{Bag}(P)$  est l'ensemble des multi-ensembles sur  $P : \text{Bag}(P) = \{x = (x(p))_{p \in P} \mid \forall p \in P, x(p) \in \mathbb{N}\}$ .

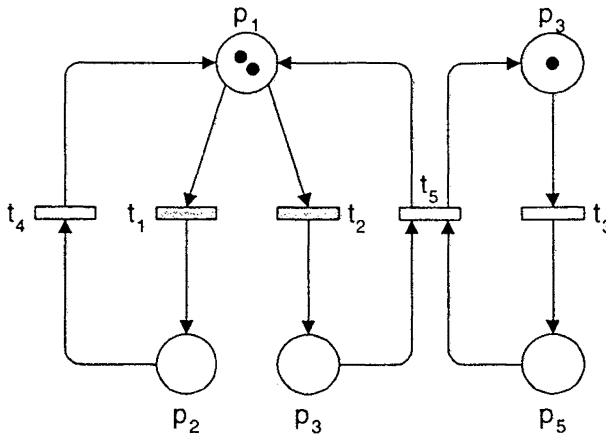


Figure 3. - Exemple de Coxian-GSPN ( $S_1$ ).

conservé pour la pièce suivante. Après la phase de préparation, les pièces sont effectivement traitées ( $t_4$ ,  $t_5$ ) en un temps suivant une loi exponentielle. Le temps de préparation de la ligne 1 ( $t_1$ ) suit une loi de Cox à deux étages tandis que celui de la ligne 2 ( $t_2$ ) est de loi d'Erlang à trois étages. De plus, la ligne 2 nécessite un autre traitement (transition  $t_3$ ) dont le temps de service suit une loi exponentielle. Le système contient en permanence deux pièces à traiter (marquage initial de deux jetons en  $p_1$ ). Les hypothèses sur les pré-traitements conduisent à adopter  $sp(t_1) = sp(t_2) = 2$  (service 2-servers) et  $pm(t_1) = mp(t_2) = A$  (mémoire Age-memory). Nous choisissons une politique d'interruption/reprise FIFO pour  $t_1(ip(t_1) = F)$  et « moins avancé » pour  $t_2(ip(t_2) = L)$ .

### 2.3. Graphe d'accessibilité d'un Coxian-GSPN

C'est le graphe d'accessibilité standard d'un GSPN où une transition-Cox est considérée comme une transition exponentielle. Pendant sa construction, les valeurs du degré de franchissabilité sous contrainte de service

$$EDS(t, \mathbf{m}) = \begin{cases} \min(ED(t, \mathbf{m})) & \text{si } sp(t) = K \text{ ou } S \\ ED(t, \mathbf{m}) & \text{si } sp(t) = I \end{cases}$$

sont utilisées pour calculer le plus grand nombre  $e(t)$  de clients qui peuvent effectivement être dans l'ensemble des étages d'une transition-Cox en un marquage quelconque de l'ensemble d'accessibilité  $RS$  du réseau :

$$e(t) = \max_{\mathbf{m} \in RS} \{EDS(t, \mathbf{m})\}$$

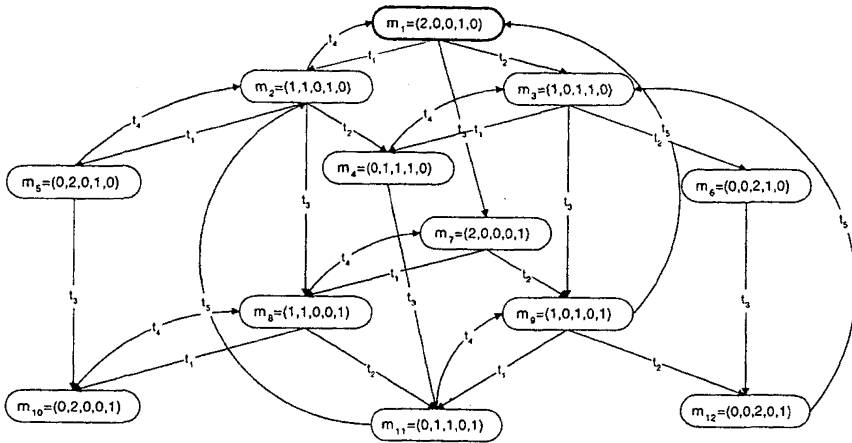


Figure 4. – Graphe d’accessibilité du réseau  $\mathcal{S}_1$ .

Nous supposons aussi que  $EDS(t, \mathbf{m})$  est toujours disponible pendant les étapes suivantes des analyses qualitative et quantitative du réseau, sans nous préoccuper d’avantage de ce problème dans le présent article.

La figure 4 présente le graphe d’accessibilité du réseau  $\mathcal{S}_1$ . Nous avons  $e(t_1) = e(t_2) = 2$ .

**2.4. Chaîne de Markov d’un Coxian-GSPN**

Un état de la chaîne est un marquage *étendu* qui, en plus du marquage ordinaire, donne pour chaque transition-Cox, son état interne, c’est-à-dire dans quel(s) état(s) exponentiel(s) se trouvent les clients en service.

Nous appelons *descripteur* de la transition, l’information à mémoriser, qui dépend des propriétés de sa sémantique stochastique.

Étant donné les descripteurs  $d(t)$  des transitions-Cox, un état de la chaîne de Markov est le *marquage étendu* compatible avec un marquage  $\mathbf{m}$  du GSPN :

$$\bar{\mathbf{m}} = (\mathbf{m}, (d(t))_{t \in T_C})$$

Nous déduisons de  $RS$  un ensemble de marquages étendus en ajoutant toutes les valeurs possibles des descripteurs. Ceci peut générer des états inaccessibles si  $mp(t) = A$  pour certaines transitions.

Avant de décrire les transitions entre états de la chaîne de Markov, donnons trois exemples de descripteurs adaptés à des cas classiques de politiques d’exécution.

### 2.4.1. Descripteurs pour politiques à clients ordonnés

Nous avons une politique à serveurs multiples ( $sp(t) = K$  ou  $I$ ), Enable memory ou Age memory. Chaque client est repéré par son instant d'arrivée dans le service d'où le qualificatif « ordonné » (par rapport au temps). La priorité de service est au client le plus ancien ( $ip(t) = F$ ).

Le descripteur de ces politiques (descripteur-1) donne pour chaque client son étage de service, qu'il soit interrompu ou non (si  $mp(t) = A$ ). En sachant que  $EDS(t, \mathbf{m}) = 0$  ssi aucun client n'est en service, nous avons :

$$d(t) = (d_1, \dots, d_i, \dots, d_{e(t)}) \in CD(t)$$

avec

$$CD(t) = \{1, \dots, s(t)\}^{e(t)}$$

Les clients sont rangés dans l'ordre d'arrivée de gauche à droite.  $d_i$  est l'étage de service courant du client  $i$  si  $i \leq EDS(t, \mathbf{m})$  et l'étage où il est interrompu (et dans lequel il sera éventuellement repris) si  $i > EDS(t, \mathbf{m})$ ; il est inutile de différencier dans les descripteurs, un client interrompu en étage 1 et l'absence de ce client, donc, pour un client non présent, on a  $d_i = 1$ .

Exemples :

$d(t)$	$EDS(t, \mathbf{m})$	situation
$(3, 1, 2, 7, 1, 1, \dots, 1)$	2	clients 1 et 2 en service. clients 3 et 4 interrompus
$(3, 1, 2, 7, 1, 1, \dots, 1)$	5	clients 1 à 5 en service
$(3, 1, 2, 7, 1, 1, \dots, 1)$	0	clients 1 à 4 interrompus
$(1, 1, \dots, 1)$	0	aucun client en service

Dans le réseau  $S_1$ , la préparation de la ligne 1 est à clients ordonnés en accord avec  $ip(t_1) = F$ . Un descripteur de  $t_1$  sera donc  $d(t_1) = (d_1, d_2)$  puisque  $e(t_1) = 2$ ; enfin, comme  $t_1$  possède deux étages,  $d_1 = d_2 = 1$  ou  $2$ .

Ces descripteurs sont les plus fins possibles et autorisent l'emploi de toute politique d'exécution.

### 2.4.2. Descripteurs pour politiques à clients banalisés

La politique de service est toujours  $K$  ou  $I$ ,  $mp(t) = E$ . Les clients ne sont connus que par l'étage dans lequel ils sont servis. Nous avons alors le descripteur (descripteur-2)

$$d(t) = (d_1, \dots, d_{s(t)}) \in SD(t)$$

avec

$$SD(t) = \{1, 2, \dots, e(t)\} \times \{0, 1, \dots, e(t)\}^{s(t)-1}$$

Pour  $EDS(t, \mathbf{m}) > 0$ ,  $d_i$  est le nombre de clients en service dans l'étage  $i$ . Si  $EDS(t, \mathbf{m}) = 0$ ,  $d_i = 0, \forall i > 1$  et  $d_1 = 1$ ; il est encore inutile de différencier dans les descripteurs, un client interrompu en étage 1 et l'absence de ce client, donc, pour un client non présent, on a  $d_1 = 1$ . Exemples :

$d(t)$	$EDS(t, \mathbf{m})$	situation
$(1, 0, \dots, 0, 2, 0, 3, 0, 0)$	6	6 clients en service en étages 1, $s-4$ et $s-2$
$(1, 0, \dots, 0, 2, 0, 3, 0, 0)$	5	5 clients en service en étages $s-4$ et $s-2$
$(1, 0, \dots, 0)$	1	1 client en service en étage 1
$(1, 0, \dots, 0)$	0	aucun client en service

Dans le réseau  $\mathcal{S}_1$ , la préparation de la ligne 2 est à clients banalisés en conformité avec  $ip(t_2) = F$ . Un descripteur de  $t_2$  est donc de la forme  $d(t_2) = (d_1, d_2, d_3)$  puisque  $t_2$  possède trois étages; de plus,  $d_1 = d_2 = d_3 = 0, 1$  ou  $2$  car  $e(t_2) = 2$ .

#### 2.4.3. Descripteurs pour politiques à serveur unique

Si la politique d'exécution de  $t$  est serveur-unique, Enabling memory ou Age memory, nous prendrons simplement comme descripteur (descripteur-3) le fait qu'un client est ou a été en service dans l'étage  $r$  :

$$d(t) = n \in SSD(t)$$

avec

$$SSD(t) = \{1, \dots, s(t)\}$$

Si  $EDS(t, \mathbf{m}) = 1$ ,  $d(t)$  est l'étage du client en service. Si  $EDS(t, \mathbf{m}) = 0$ ,  $d(t)$  est l'étage du client interrompu.

Exemples :

$d(t)$	$EDS(t, \mathbf{m})$	situation
6	1	client en service en étage 6
6	0	client interrompu en étage 6
1	1	client en service en étage 1
1	0	aucun client

Chacun de ces descripteurs peut être modifié (simplifié) lorsque la politique d'exécution de la transition le permet : par exemple, en Enabling memory, il est inutile de conserver l'information sur les états des clients interrompus. Naturellement, il est possible d'avoir des politiques différentes pour diverses transitions-Cox d'un même réseau de Petri stochastique. Dans ce cas, on emploiera pour chacune, le descripteur approprié.

Les descripteurs-1 sont naturellement très « coûteux » en mémoire de données puisque  $|CD(t)| = (s(t))^{e(t)}$ . Les descripteurs-2 consomment moins de mémoire ( $|SD(t)| = e(t).(e(t) + 1)^{s(t)-1}$ ) et les descripteurs-3 contiennent le minimum d'informations (pour pouvoir gérer la politique Age memory). On voit que, pour un nombre fixé d'étages de  $t$ , l'occupation mémoire de  $CD(t)$  peut croître exponentiellement avec le cardinal de l'ensemble d'accessibilité du réseau, alors que cette croissante est polynomiale pour les descripteurs-2.

## 2.5. Transitions de la chaîne de Markov

Ces transitions (avec leur vitesse) entre états de la chaîne de Markov se déduisent, comme les GSPNs standard, d'un graphe que nous appelons *graphe d'accessibilité étendu* (noté  $\overline{RG}$ ) : ses nœuds sont les marquages étendus, et un arc entre  $\overline{m}$  et  $\overline{m}'$  correspond au franchissement d'une transition du réseau. Pour décrire les modifications du descripteur d'une transition-Cox  $t'$  provoquées par le franchissement d'une autre transition

### Algorithme 2.1 (Construction de $\overline{m} \xrightarrow{t'} \overline{m}'$ , $t' \in T_E$ )

$m' = m + \text{Post}(\cdot, t') - \text{Pre}(\cdot, t')$

pour chaque  $t' \in T_C$  faire

$d'(t') = d(t')$

(\* interruptions éventuelles \*)

si  $(EDS(t', m') - EDS(t', m) < 0) \wedge (mp(t') = E)$  alors

(\* interromptre définitivement  $|EDS(t', m') - (EDS(t', m))|$  clients \*)

pour  $i$  de  $EDS(t', m')$  + 1 à  $EDS(t', m)$  faire

$d'_i(t') = 1$

finpour

fin si (\* rien à faire si  $mp(t') = A$  \*)

finpour

ajouter l'arc  $\overline{m} \xrightarrow{t'} \overline{m}'$  à  $\overline{RG}$



$t$ , nous devons distinguer trois types de franchissements de  $t$  (pour un marquage étendu  $\bar{\mathbf{m}}$  donné) :

1.  $t$  est exponentielle : nous notons  $\bar{\mathbf{m}} \xrightarrow{t} \bar{\mathbf{m}}'$  un tel franchissement ;
2.  $t$  est une transition-Cox et son franchissement est une fin de service de l'étage  $r$  avec sortie du service de la transition. Nous appelons «  $\alpha$ -franchissement » un tel franchissement et nous le notons  $\bar{\mathbf{m}} \xrightarrow{t_r^\alpha} \bar{\mathbf{m}}'$  ;
3.  $t$  est une transition-Cox et son franchissement est une fin de service de l'étage  $r$  avec branchement vers l'étage suivant. Nous nommons «  $\beta$ -franchissement » un tel tir, et nous le notons  $\bar{\mathbf{m}} \xrightarrow{t_r^\beta} \bar{\mathbf{m}}'$ . Le marquage  $\mathbf{m}$  est inchangé dans ce cas.

Nous donnons maintenant les algorithmes détaillés pour calculer  $\overline{RG}$  dans le cas où toutes les transitions-Cox ont des descripteurs-1. Remarquons que ces descripteurs n'ont pas à être modifiés lorsque de nouveaux clients sont engagés. Soit  $\bar{\mathbf{m}} = (\mathbf{m}, (d(t))_{t \in T_C})$  un marquage étendu. Le franchissement  $\bar{\mathbf{m}} \xrightarrow{t}$  d'une transition exponentielle  $t$  produit un unique marquage étendu  $\bar{\mathbf{m}}' = (\mathbf{m}', (d'(t))_{t \in T_C})$  calculé par l'algorithme 2.1. La vitesse de cette transition élémentaire d'état est  $\theta(t, \mathbf{m})$ .

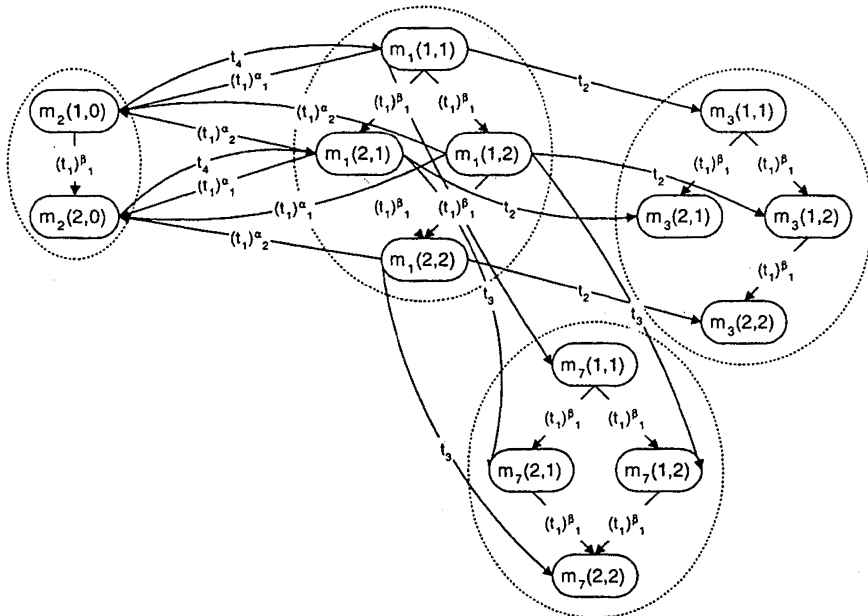


Figure 5. – Extrait du graphe des transitions de la chaîne de Markov de  $S'_1$ .

Le franchissement  $\overline{\mathbf{m}} \xrightarrow{t_r^\beta}$  produit EDS  $(t, M)$  marquages  $\overline{\mathbf{m}}'$  (un pour chaque client possible quittant le service). Il est calculé par l'algorithme 2.2. La vitesse correspondante est  $\mu_r(t) \cdot \alpha_r(t)$ .

Le franchissement  $\overline{\mathbf{m}} \xrightarrow{t_r^\alpha}$  produit au plus EDS  $(t, M)$  marquages  $\overline{\mathbf{m}}'$  (un au plus pour chaque client éventuel engagé) (algorithme 2.3) et chaque transition élémentaire a pour vitesse  $\mu_r(t) \cdot \beta_r(t)$ .

Exemple du réseau  $\mathcal{S}_1$  :

Les figures suivantes se rapportent à des versions simplifiées de  $\mathcal{S}_1$  où sont isolés les effets de  $t_1$  ( $\mathcal{S}'_1$ ) et  $t_2$  ( $\mathcal{S}''_1$ ). Ces effets, qui se conjuguent dans  $\mathcal{S}_1$ , génèrent un graphe si peu lisible que nous avons préféré présenter ces versions qui illustrent déjà le phénomène de complexité « bi-dimensionnelle » mentionné dans l'introduction.

La figure 5 présente uniquement *un extrait* du graphe de la chaîne de Markov du réseau  $\mathcal{S}'_1$ , version *simplifiée* (en termes de chaîne de Markov) de  $\mathcal{S}_1$  dans lequel  $t_2$  est remplacée par une transition exponentielle : seuls les marquages étendus de marquage  $\mathbf{m}_1$ ,  $\mathbf{m}_2$ ,  $\mathbf{m}_3$  et  $\mathbf{m}_7$  sont représentés et seules les transitions entre marquages étendus de marquage  $\mathbf{m}_1$  et un autre marquage étendu représenté sont indiquées.

On donne en figure 6 *un extrait* (dans le même sens qu'au-dessus) du graphe de la chaîne de Markov du réseau  $\mathcal{S}''_1$ , version *simplifiée* (en termes de chaîne de Markov) de  $\mathcal{S}_1$  dans lequel  $t_1$  est remplacée par une transition exponentielle.

### 3. DÉCOMPOSITION STRUCTURELLE DES COXIAN-GSPNs

Les algorithmes ci-dessus démontrent l'impact de l'introduction des transitions-Cox sur les besoins en mémoire et le coût du calcul des probabilités à l'équilibre de la chaîne de Markov d'un Coxian-GSPN. La décomposition tensorielle que nous introduisons maintenant est l'élément clé pour pallier cette difficulté. La décomposition tensorielle [14, 7, 16, 17] d'une chaîne de Markov se fonde sur l'expression de l'espace d'états de la chaîne comme *produit cartésien* d'espaces plus petits et sur la distinction entre les changements locaux d'état, c'est-à-dire modifiant une seule composante de l'état global, et les changements globaux qui modifient plusieurs composantes de l'état global. Puisqu'un marquage étendu est constitué d'un marquage standard et d'un tuple de descripteurs des transitions-Cox, nous devons introduire une *partition* de l'ensemble des places qui nous permette également d'isoler les effets des franchissements sur les descripteurs. Comme nous

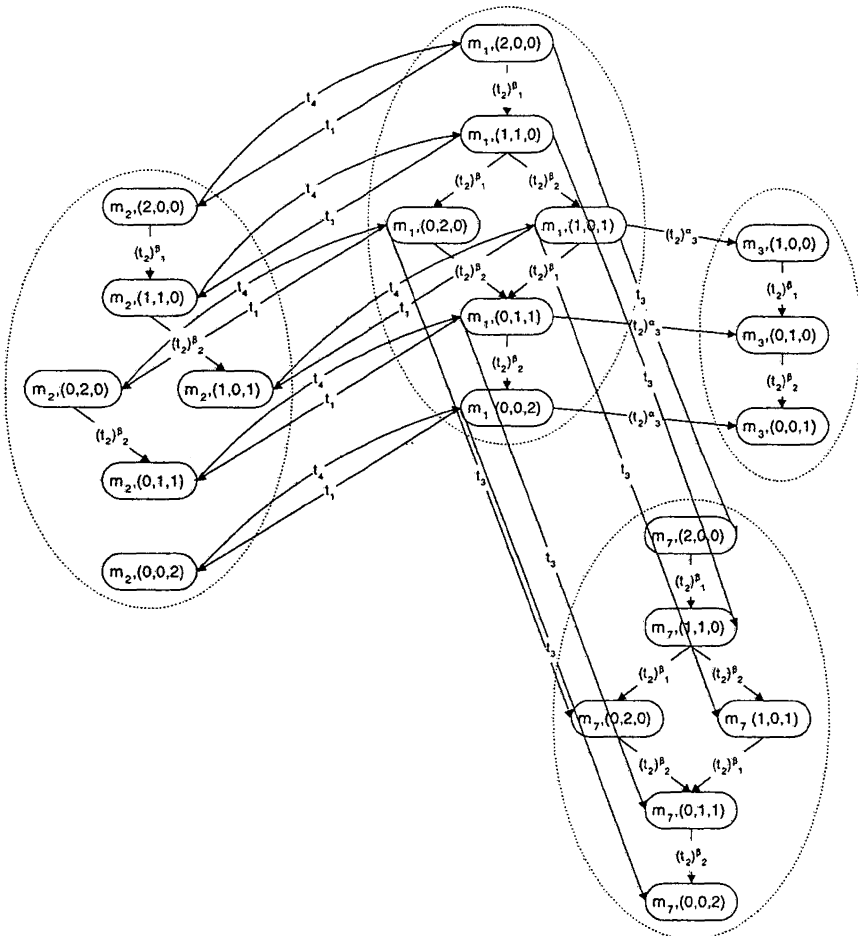


Figure 6. – Extrait du graphe des transitions de la chaîne de Markov de  $S_1''$ .

l'avons indiqué dans la section 2.5, le descripteur d'une transition  $t$  donnée peut être modifié par le franchissement de  $t'$  si et seulement si  $t'$  peut changer  $EDS(t, \cdot)$ .

Rappelons que deux transitions  $t$  et  $t'$  sont en conflit structurel symétrique (Symmetric Structural Conflict, SSC) ssi le franchissement de  $t$  peut rendre  $t'$  non franchissable ou *vice versa*. Les ensembles de conflit étendu (ECSs) du réseau sont les classes d'équivalence de la fermeture transitive de la relation SSC [3].

**Algorithme 2.2 (Construction des  $\overline{m} \xrightarrow{t_r^\alpha} \overline{m}'$ , à partir de  $\overline{m} \xrightarrow{t_r^\alpha}$ )** $m' = m + \text{Post}(\cdot, t) - \text{Pre}(\cdot, t)$ **pour** chaque  $t' \in T_C$ ,  $t' \neq t$  **faire**(\* calculer  $d'(t')$  comme dans le cas  $\xrightarrow{t}$  \*) $d'(t') = d(t')$ 

(\* interruptions éventuelles \*)

**si**  $(\text{EDS}(t', m') - \text{EDS}(t', m) < 0) \wedge (\text{mp}(t') = E)$  **alors**(\* interrompre définitivement  $|(\text{EDS}(t', m') - (\text{EDS}(t', m)|$  clients \*)**pour**  $i$  de  $(\text{EDS}(t', m') + 1$  à  $(\text{EDS}(t', m)$  **faire** $d'_i(t') = 1$ **finpour****fin**si (\* rien à faire si  $\text{mp}(t') = A$  \*)**finpour**(\* supprimer les clients en étage  $r$  de  $t$  \*) $i = 1$ **tant que**  $i \leq \text{EDS}(t, m)$  **faire****si**  $(d_i(t) = r)$  **alors**

(\* supprimer ce client \*)

 $d'(t) = d(t)$ **pour**  $j$  de  $i + 1$  à  $e(t)$  **faire** $d'_{j-1}(t) = d'_j(t)$ **finpour** $d'_{e(t)}(t) = 1$ ajouter l'arc  $\overline{m} \xrightarrow{t_r^\alpha} \overline{m}'$  à  $\overline{RG}$ **fin**si $i = i + 1$ **fin**tanque

Pour garantir qu'une partition permettra une décomposition tensorielle, nous lui imposons des conditions fondées sur les ensembles de conflit étendu (ECSs) du réseau.

DEFINITION 3.1 (Partition compatible) : Soient  $(ECS_j)_{1 \leq j \leq n}$  les ensembles de conflit étendu du réseau. Une partition  $(P_i)_{1 \leq i \leq m}$  de l'ensemble des places  $P$  est compatible par rapport aux transitions-Cox ssi

$$\forall 1 \leq j \leq n,$$

$$(ECS_j \text{ contient une transition-Cox} \Rightarrow \exists i_j, \bullet ECS_j \cup {}^\circ ECS_j \subseteq P_{i_j})$$

où  $\bullet ECS_j = \bigcup_{t \in ECS_j} \text{Pre}^{-1}(\cdot, t)$  et  ${}^\circ ECS_j = \bigcup_{t \in ECS_j} \text{Inh}^{-1}(\cdot, t)$ .

**Algorithme 2.3 (Construction des  $\overline{m} \xrightarrow{t_r^\beta} \overline{m}'$ , à partir de  $\overline{m} \xrightarrow{t_r^\beta}$ )**

```

m' = m
pour chaque  $t' \in T_C, t' \neq t$  faire
     $d'(t') = d(t')$ 
finpour
(* avancer les clients qui sont en étage r *)
i = 1
tant que  $i \leq EDS(t, m)$  faire
    si  $(d_i(t) = r)$  alors
        (* avancer ce client *)
         $d'(t) = d(t)$ 
         $d'_i(t) = r + 1$ 
        ajouter l'arc  $\overline{m} \xrightarrow{t_r^\beta} \overline{m}'$  à  $\overline{RG}$ 
    fin
    i = i + 1
fintantque
    
```

Les marquages  $\mathbf{m}_i$  sont les marquages sur les  $P_i$  :

$$\forall 1 \leq i \leq m, \mathbf{m}_i = \mathbf{m}(P_i)$$

Dans le réseau  $\mathcal{N}_1$ , nous avons  $ECS_1 = \{t_1, t_2\}$ ,  $ECS_2 = \{t_3\}$ ,  $ECS_3 = \{t_4\}$  et  $ECS_4 = \{t_5\}$ . Nous pouvons donc choisir  $P_1 = \{p_1\}$  et  $P_2 = \{p_2, p_3, p_4, p_5\}$ .

Dans l'exemple plus complexe du réseau  $S_2$  de la figure 7 (les transitions-Cox sont en grisé), nous avons deux ECSs avec transitions-Cox (entourés d'un trait discontinu) :  $ECS_1 = \{t_{c1}, t_{c2}\}$  et  $ECS_2 = \{t_{e5}, t_{c3}\}$ . Nous devons donc choisir une partition telle que  $t_{c1}$  et  $t_{c2}$  (resp.  $t_{e5}$  et  $t_{c3}$ ) appartiennent au même  $P_i$ .

On remarquera que ces conditions autorisent un calcul automatique d'une partition compatible en temps polynomial, à savoir exactement les

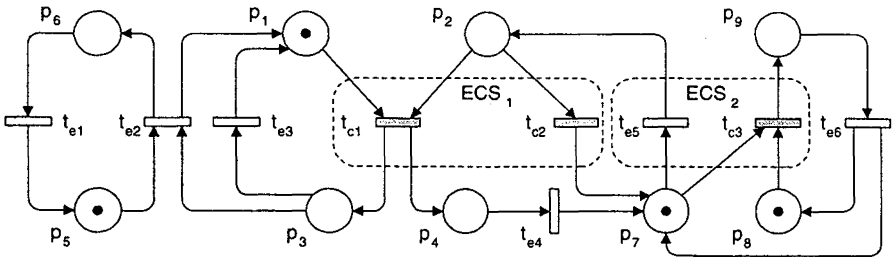


Figure 7. - Autre exemple de Coxian-GSPN ( $S_2$ ).

$\bullet ECS_j = \bigcup^\circ ECS_j$ , puisque les ECSs d'un réseau peuvent être obtenus en temps polynomial par rapport aux tailles de  $P$  et  $T$ .

Pour différencier les franchissements locaux et globaux, nous introduisons les ensembles  $T_i$  de transitions qui sont les seules pouvant modifier le marquage d'une place de  $P_i$  :

$$T_i = \bullet P_i \bigcup P_i \bullet$$

De plus, l'ensemble des transitions-Cox dont les descripteurs dépendent de  $P_i$  est

$$T_{C,i} = \{t \in T_C \mid \bullet ECS(t) \bigcup^\circ ECS(t) \subseteq P_i\}$$

Soulignons que  $T_i$  peut contenir des transitions-Cox qui *ne sont pas dans*  $T_{C,i}$  : ce sont celles qui placent des jetons dans  $P_i$  (fonctions **Post**) sans en prendre (fonctions **Pre**).

Pour le réseau  $\mathcal{N}_1$ , nous obtenons  $T_1 = \{t_1, t_2, t_4, t_5\}$  et  $T_2 = T$ , l'ensemble de toutes les transitions.

Si nous définissons les  $(P_i)$  du réseau exemple  $\mathcal{N}_2$  par  $P_1 = \{p_1, p_2, p_3, p_4, p_5, p_6\}$  et  $P_2 = \{p_7, p_8, p_9\}$ , nous avons :

$$T_1 = \{t_{e1}, t_{e2}, t_{e3}, t_{e4}, t_{e5}, t_{C1}, t_{C2}\} \text{ et } T_2 = \{t_{e4}, t_{e5}, t_{e6}, t_{C2}, t_{C3}\}$$

(les autres ECSs de  $\mathcal{N}_2$  sont :  $ECS_3 = \{t_{e2}, t_{e3}\}$ ,  $ECS_4 = \{t_{e1}\}$ ,  $ECS_5 = \{t_{e4}\}$  et  $ECS_6 = \{t_{e6}\}$ ).

### 3.1. Graphes d'accessibilité locaux étendus

Soit  $(P_i)_{1 \leq i \leq m}$  une partition compatible. Au lieu de construire le graphe d'accessibilité étendu  $(\overline{RG})$  du réseau tout entier, nous construisons des graphes d'accessibilité *locaux* étendus, c'est-à-dire restreints à  $(P_i, T_i)$  et aux arcs les joignant. Nous calculons tout d'abord le graphe d'accessibilité du réseau complet.  $\forall 1 \leq i \leq m$ , l'ensemble des valeurs possibles des  $\mathbf{m}_i$  est noté  $RS_i$ .

**DEFINITION 3.2** (Marquage local étendu, graphe d'accessibilité local étendu) : *Un marquage local étendu d'indice  $i$  ( $1 \leq i \leq m$ ) d'un marquage  $\mathbf{m}$  est*

$$\overline{\mathbf{m}}_i = (\mathbf{m}_i, d(t)_{t \in T_{C,i}}) \text{ avec } \mathbf{m}_i \in RS_i$$

et  $d(t)$  une valeur du descripteur de  $t$  compatible avec sa politique d'exécution.

On note  $\overline{RS}_i$  l'ensemble des marquages  $\overline{m}_i$ .

Le  $i$ ème graphe d'accessibilité local étendu de  $S$ , noté  $\overline{RG}_i$ , est le graphe d'accessibilité étendu calculé comme en section 2.5 mais restreint aux marquages locaux étendus de  $\overline{RS}_i$  et aux transitions de  $T_i$ .

En appliquant ces définitions à notre réseau exemple simplifié  $S'_1$ , nous obtenons les graphes  $\overline{RG}_1$  et  $\overline{RG}_2$  entièrement représentés sur les figures 8 et 9. On remarquera en particulier que  $\overline{RG}_2$  est indépendant de la loi de  $t_1$ .

### 3.2. Expression tensorielle du générateur

Chaque franchissement d'une transition du réseau peut être local, c'est-à-dire modifiant uniquement un marquage local étendu, ou global, en modifiant

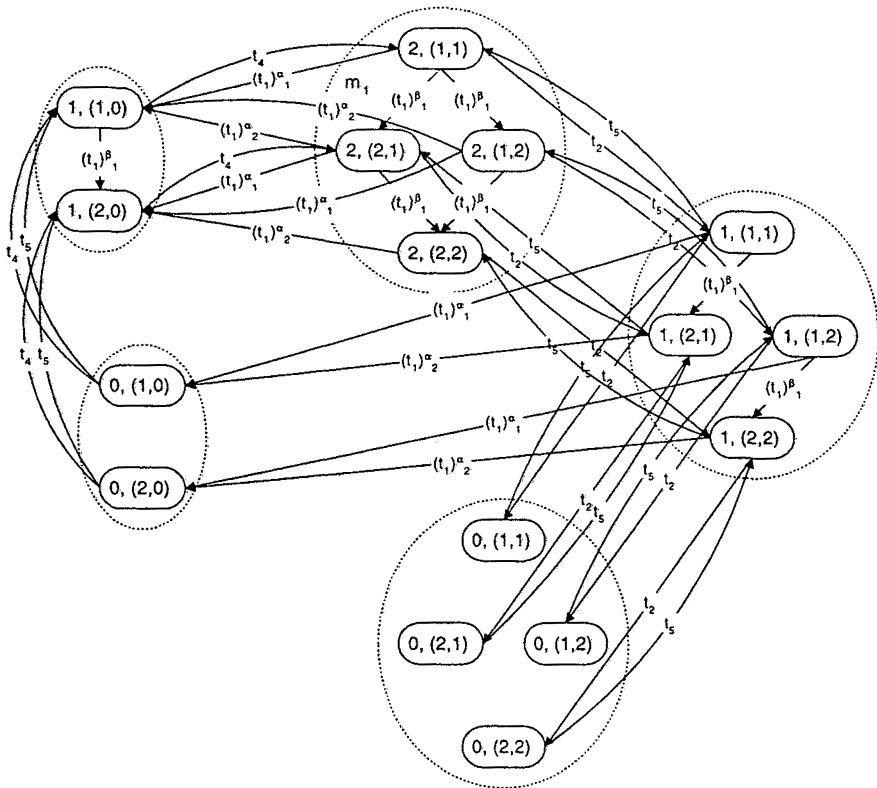


Figure 8. – Graphe d'accessibilité local étendu 1 du réseau  $S'_1$ .

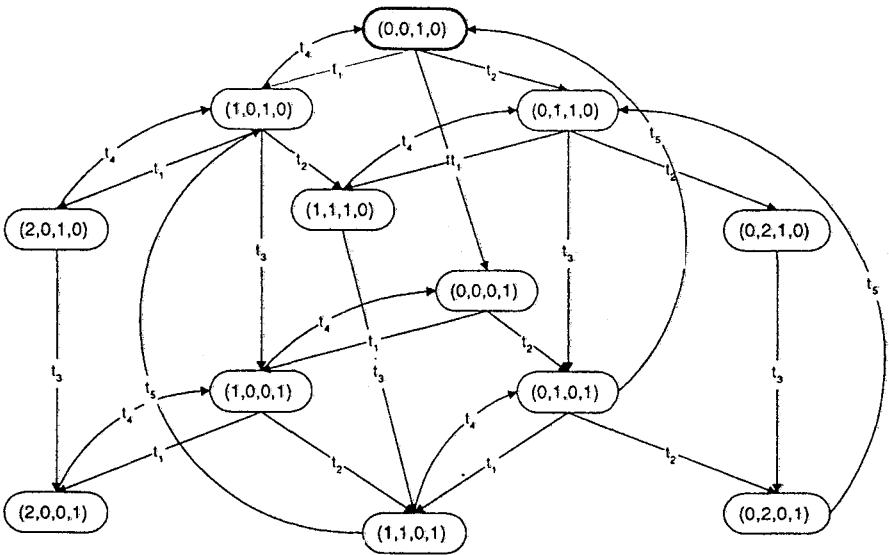


Figure 9. – Graphe d'accessibilité local étendu 2 du réseau  $S_1'$ .

plusieurs simultanément. La décomposition tensorielle s'appuie sur cette classification pour obtenir une expression du générateur de la CTMC. Introduisons tout d'abord les transitions qui ne changent que les marquages locaux étendus de leur propre sous-réseau :

DEFINITION 3.3 (transition locale) : *Étant donné une partition  $(P_i)_{1 \leq i \leq m}$  des places du réseau, une transition  $t$  est dite locale ssi*

$$\exists 1 \leq i \leq m, \quad \bullet t \cup \circ t \subseteq P_i$$

Nous pouvons maintenant décrire les franchissements locaux et globaux :

**les franchissements locaux** sont ceux des transitions locales de  $T_i$  et tous les  $\beta$ -franchissements des transitions-Cox non locales de  $T_{C,i}$ .

**les franchissements globaux** sont ceux des transitions exponentielles non locales de  $T_i$  ainsi que les  $\alpha$ -franchissements des transitions-Cox non locales de  $T_{C,i}$ .

Notons  $T_i^{(L)}$  l'ensemble des transitions locales de  $T_i$ ,  $T_i^{(X)}$  celui des transitions non locales et  $T_{C,i}^{(X)}$  le sous-ensemble des transitions-Cox non locales de  $T_{C,i}$ . Nous avons :

$$\forall 1 \leq i \leq m, \quad T_i = T_i^{(L)} \uplus T_i^{(X)} = T_i^{(L)} \uplus T_{C,i}^{(X)} \uplus T_i^{(X)}$$



où  $T_i^{(X)}$  est l'ensemble des transitions exponentielles non locales de  $T_i$  et des transitions-Cox non locales de  $T_i \setminus T_{C,i}$ . L'ensemble de toutes les transitions exponentielles externes (i.e. non locales) est alors

$$T' = \left( \bigcup_{i=1}^m T_i^{(X)} \right) \setminus T_C$$

Avec ces notations, nous pouvons écrire :

$$T = \left( \biguplus_{i=1}^m T_i^{(L)} \right) \biguplus \left( \biguplus_{i=1}^m T_{C,i}^{(X)} \right) \biguplus T'$$

Dans le réseau  $S_2$  par exemple, nous avons :  $T_i^{(L)} = \{t_{e1}, t_{e2}, t_{e3}, t_{C1}\}$ ,  $T_{C,1}^{(X)} = \{t_{C2}\}$ ,  $T_1^{(X)} = \{t_{e4}, t_{e5}\}$  et  $T_2^{(L)} = \{t_{e6}, t_{C3}\}$ ,  $T_{C,2}^{(X)} = \emptyset$ ,  $T_2^{(X)} = \{t_{e4}, t_{e5}, t_{C2}\}$ .

En distinguant les franchissements locaux et globaux, on en déduit, comme pour la composition synchrone des GSPNs [14], que le générateur de la CTMC sous-jacente au réseau est une sous-matrice de :

$$Q' = \bigoplus_{i=1}^m Q_i^0 + \sum_{i=1}^m \sum_{t \in T_{C,i}^{(X)}} \left[ \bigoplus_{j=1}^m C_j(t) - \bigoplus_{j=1}^m A_j(t) \right] + \sum_{t \in T'} \left[ \bigoplus_{j=1}^m C_j(t) - \bigoplus_{j=1}^m A_j(t) \right] \tag{1}$$

La somme tensorielle correspond aux franchissements locaux tandis que les produits des deuxième et troisième termes correspondent aux franchissements globaux.

### 3.2.1. Calcul des $Q_i^0$

Ce sont les contributions de tous les franchissements des transitions locales exponentielles ou Cox de  $T_i$ , et des  $\beta$ -franchissements des transitions-Cox non locales de  $T_{C,i}$  :

$$q_{i, \bar{\mathbf{m}}_i, \bar{\mathbf{m}}'_i}^0 = \sum_{\substack{t \in T_i^{(L)} \cap T_E \\ \bar{\mathbf{m}}_i \xrightarrow{t} \bar{\mathbf{m}}'_i}} \theta(t, \mathbf{m}_i) + \sum_{\substack{t \in T_i^{(L)} \cap T_C \\ \bar{\mathbf{m}}_i \xrightarrow{t, \alpha} \bar{\mathbf{m}}'_i}} \mu_r(t) \cdot \alpha_r(t) + \sum_{\substack{t \in T_{C,i} \\ \bar{\mathbf{m}}_i \xrightarrow{t, \beta} \bar{\mathbf{m}}'_i}} \mu_r(t) \cdot \beta_r(t)$$

pour  $\bar{\mathbf{m}}_i \neq \bar{\mathbf{m}}'_i$ .

### 3.2.1. Calcul des $C_j(t)$

Il est donné par l'algorithme 3.1 où l'on a séparé le cas des transitions-Cox (deuxième terme de  $Q'$ ) et celui des transitions de  $T'$  (troisième terme de  $Q'$ ). Signalons que dans le premier cas, nous écrivons  $\overline{\mathbf{m}}_j \xrightarrow{t} \overline{\mathbf{m}}'_j$  car nous n'utilisons pas le fait que nous ayons un  $\alpha$ -franchissement de  $t$ .

## 4. ÉVALUATION DE LA MÉTHODE

Nous étudions les gains qu'apportent notre méthode par rapport aux deux méthodes antérieures [13] et [8]. Notons d'abord que quelque soit la méthode, nous avons à résoudre une CTMC dont l'espace d'états est identique. C'est donc sur ce plan que nous plaçons : quels sont les gains obtenus pendant la résolution de la CTMC sous-jacente en utilisant la méthode de décomposition au lieu de la méthode « standard » ? Comme indicateur de complexité de chaque méthode, nous prenons la somme des cardinaux des espaces d'états manipulés (notée  $S$  pour la méthode standard et  $D$  pour notre méthode) : pour les modèles importants, c'est le facteur le plus limitatif et la complexité temporelle en dépend très fortement. De plus, la méthode de décomposition permet, à complexité de données du même ordre, d'effectuer les traitements en parallèle réel, et donc, en fait, de réduire la complexité temporelle.

La diversité des interactions possibles entre les transitions-Cox et le reste du réseau interdit une comparaison générale. Nous proposons ici une première étude de situation type pour deux des politiques étudiées plus haut : serveur unique et clients ordonnés.

Le réseau (fig. 10) comporte une seule transition-Cox,  $t_x$  contrôlée par une seule place  $p_1$  dont le marquage maximal est  $c$ .  $t_x$  comporte  $s$  étages. Le reste du réseau comporte  $n = |P| - 1$  places, dont aucune ne contrôle les transitions en conflit avec  $t_x$ . Nous pouvons donc choisir  $P_1 = \{p_1\}$  et  $P_2 = P \setminus P_1$  (on a ainsi  $|P_2| = n$ ).

Dans tous les cas, nous approximations le nombre de marquages d'un réseau à  $p$  places et  $j$  jetons par sa borne supérieure  $\binom{p+j-1}{p-1}$ .

### 4.1. Politique serveur-unique

La politique de mémoire est Age memory ou Enabling memory.

Avec la méthode standard, nous avons

$$S \approx s \cdot \left[ \sum_{i=0}^{c-1} \binom{n+i-1}{n-1} \right] + \binom{n+c-1}{n-1}$$

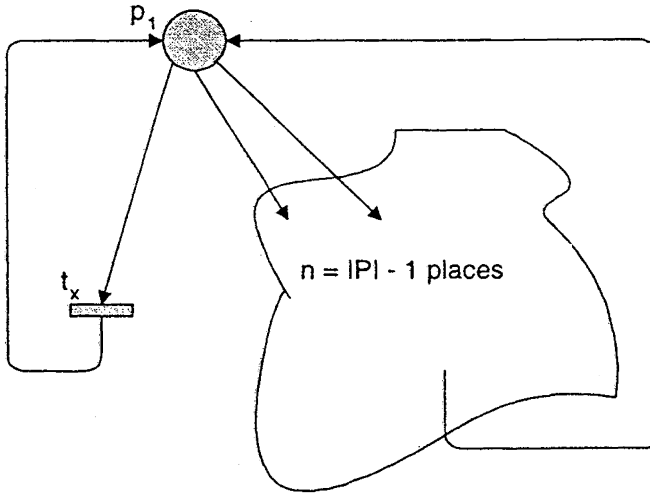


Figure 10. – Modèle pour l'évaluation de la méthode de décomposition.

### Algorithme 3.1 (Calcul des matrices $C_j(t)$ )

1) Cas  $t \in T_{C,i}$

si  $j \neq i$  alors

si  $t \notin T_j$  alors

$$C_j(t) = I_{|\overline{RS}_j|}$$

sinon (\* une transition-Cox peut être dans plusieurs  $T_j$  \*)

$$c_j(t)_{\overline{m}_j, \overline{m}'_i} = \begin{cases} 1 & \text{si } \overline{m}_j \xrightarrow{t} \overline{m}'_i \in \overline{RG}_j \\ 0 & \text{sinon} \end{cases}$$

finsi

sinon

(\*  $\alpha$ -franchissement de  $t$  : fin de service de la loi de Cox \*)

$$c_i(t)_{\overline{m}_i, \overline{m}'_i} = \begin{cases} \mu_r \alpha_r & \text{si } \overline{m}_i \xrightarrow{t} \overline{m}'_i \in \overline{RG}_i \\ 0 & \text{sinon} \end{cases}$$

finsi

2) Cas  $t \in T_i \cap T'$

si  $j \neq i$  alors

si  $t \notin T_j$  alors

$$C_j(t) = I_{|\overline{RS}_j|}$$

sinon

$$c_j(t)_{\overline{m}_j, \overline{m}'_j} = \begin{cases} 1 & \text{si } \overline{m}_j \xrightarrow{t} \overline{m}'_j \in \overline{RG}_j \\ 0 & \text{sinon} \end{cases}$$

finsi

sinon

$$c_i(t)_{\overline{m}_i, \overline{m}'_i} = \begin{cases} \theta(t) & \text{si } \overline{m}_i \xrightarrow{t} \overline{m}'_i \in \overline{RG}_i \\ 0 & \text{sinon} \end{cases}$$

finsi

le premier terme correspondant aux cas où au moins un client est présent dans  $p_1$ . Puisque  $\binom{n+c}{n} = \sum_{i=0}^c \binom{n+i-1}{n-1}$ , nous avons

$$S \approx (s-1) \cdot \left[ \sum_{i=0}^{c-1} \binom{n+i-1}{n-1} \right] + \binom{n+c}{n}$$

Avec la méthode par décomposition, nous avons

$$D = |RS_0| + |\overline{RS}_1| \approx \binom{n+c}{n} + s.c + 1$$

puisque  $P_1$  est réduit à  $p_1$ .

Comparons d'abord les cardinaux  $S$  et  $D$  :

Si  $n \gg c$ ,  $\binom{n+c}{n} \approx \frac{n^c}{c!}$ , donc,  $R = \frac{S}{D} \approx 1 + \sum_{i=0}^{c-1} \frac{\frac{n^i}{i!}}{\frac{n^c}{c!}}$ . Ainsi  $R > 1$ , mais  $\lim_{n \rightarrow \infty} R = 1$  : intuitivement, si  $n$  croît plus vite que  $c$ , l'influence de la transition-Cox devient négligeable et la méthode de décomposition ne peut guère tirer avantage de la partition du réseau.

Si  $c \gg n$ ,  $R \approx 1 + (s-1) \frac{\sum_{i=0}^{c-1} \frac{n^i}{i!}}{\frac{n^c}{c!}}$  et dans ce cas,  $\lim_{n \rightarrow \infty} R = s$  : si le nombre de clients croît plus vite que la complexité de leurs comportements dans le réseau, l'influence de leurs positions dans la transition-Cox augmente et notre méthode en tire partie.

Par ailleurs, pour analyser le comportement de la méthode par décomposition lorsque la taille du système croît, nous utilisons l'indice  $A = \frac{D}{|RS_0|} = 1 + \frac{s.c+1}{\binom{n+c}{n}}$ . On constate que la méthode contrôle bien l'accroissement de complexité dû à la loi de Cox.

#### 4.2. Politique à clients ordonnés

La politique de mémoire est Age memory.

Méthode standard : quelque que soit la valeur de EDS  $(t_x, \mathbf{m})$ , il y a  $s^c$  marquages étendus possibles pour  $p_1$  donc

$$S \approx s^c \cdot \binom{n+c}{n}$$

Avec la méthode par décomposition,

$$D \approx \binom{n+c}{n} + (c+1) \cdot s^c$$

La comparaison entre les deux méthodes donne cette fois un avantage beaucoup plus net à la décomposition.

Si  $n \gg c$ ,  $S \approx 1s^c \cdot \frac{n^c}{c!}$  et  $D \approx \frac{n^c}{c!}$  puisque  $(c+1) \cdot s^c$  est négligeable, d'où  $R \approx s^c$ , qui est très important. Par exemple, pour  $s = 2$ ,  $c = 10$ ,  $R \approx 1000$ .

Si  $c \gg n$ , on a un  $R \approx \frac{c^{n-1}}{n!}$ , qui croît très vite avec  $c$ , à  $n$  fixé : par exemple pour  $n = 10$ ,  $s = 2$ , on obtient pour  $c = 10$ ,  $R \approx 200$ , pour  $c = 20$ ,  $R \approx 10^5$  et pour  $c = 30$ ,  $R \approx 5 \cdot 10^6$ .

L'accroissement de complexité dans la méthode par décomposition est maintenant  $A = 1 + \frac{(c+1)s^c}{\binom{n+c}{c}}$ . Pour  $n \gg c$ ,  $A \approx 1 + (c+1)c! \binom{s}{n}^c$  qui tend vers 1 lorsque  $n$  tend vers  $+\infty$ .

Pour  $c \gg n$ ,  $A \approx 1 + (c+1)n!s^c c^{-n}$  : la croissance due au terme  $s^c$  est atténuée par le coefficient  $c^{-n}$ .

## 5. INTRODUCTION DES TRANSITIONS IMMÉDIATES

Nous étendons la méthode précédente aux Coxian-GSPNs avec transitions immédiates et transitions-Cox à franchissement immédiat, i.e.  $\alpha_0(t) \neq 0$  (appelées transitions-Cox immédiates). Ceci pose deux problèmes : quelle est la sémantique des transitions-Cox immédiates, comment obtenir une décomposition tensorielle de la CTMC sous-jacente ?

### 5.1. Sémantique stochastique des transitions-Cox immédiates

Dans cette section, nous définissons les propriétés supplémentaires du modèle stochastique qui nous permettent de spécifier complètement le processus stochastique sous-jacent au réseau.

Puisqu'une transition-Cox immédiate peut être vue comme une transition immédiate, sa priorité doit être définie de manière cohérente : elle doit être 1 si toutes les transitions immédiates ont pour priorité 1, et toute valeur strictement positive sinon. La priorité d'une transition-Cox non immédiate est 0.

Trois situations de conflit doivent être spécifiées.

1. Conflit avec une transition immédiate au moins : du point de vue modélisation, le conflit doit être résolu entre les seuls franchissements immédiats. Nous attribuons un *poids*  $\theta(t)$  aux transitions-Cox immédiates comme pour les transitions immédiates standard et le conflit est résolu comme dans les GSPNs standard.  $\alpha_0$  n'est pas utilisé dans ce cas.

2. Conflit avec des transitions exponentielles ou Cox non immédiates exclusivement : nous devons pouvoir choisir entre le  $\alpha_0$  et le  $\beta_0$ -franchissement de  $t$  :  $t$  est franchie immédiatement avec une probabilité  $\alpha_0(t)$ .
3. Conflit entre  $L$  transitions-Cox immédiates et des transitions exponentielles ou Cox non immédiates : nous avons  $L + 1$  franchissements possibles ;  $L$  franchissements immédiats et un franchissement où les  $L$  transitions engagent un client dans leur premier étage exponentiel. Ici aussi, les poids  $\theta(t_l)$  permettent de sélectionner le franchissement immédiat ; ce choix est alors pondéré par  $\alpha_0(t_l)$ .

$$\Pr(\text{franchissement immédiat de } t_l) = \frac{\theta(t_l)}{\sum_{l=1}^L \theta(t_l)} \alpha_0(t_l)$$

et

$$\begin{aligned} & \Pr(\text{engagement dans le premier étage des } (t_l)_{1 \leq l \leq L}) \\ &= \frac{\sum_{l=1}^L \theta(t_l) \beta_0(t_l)}{\sum_{l=1}^L \theta(t_l)} \end{aligned}$$

Soulignons que si la définition de l'ensemble de ces paramètres peut sembler constituer un frein à l'utilisation de la méthode, nous exposons ici le cas le plus général, et en pratique, le modélisateur n'aura qu'à préciser un petit nombre d'entre-eux.

## 5.2. Décomposition du réseau

Puisque nous appliquons une décomposition tensorielle au générateur d'une CTMC et comme les franchissements des transitions immédiates et les  $\alpha_0$ -franchissements des transitions-Cox immédiates n'apparaissent pas dans le graphe d'accessibilité tangible (TRG), les modifications résultant de ces franchissements doivent être *locales* aux sous-ensembles  $P_i$ .

Nous imposons donc que *toutes les transitions immédiates et Cox immédiates* soient *locales*. Notons que, étant donné une partition  $(P_i)$ , il est souvent aisé de modifier certains sous-ensembles pour transformer une transition immédiate ou Cox immédiate non locale en une locale, comme sur la figure 11 où, en remplaçant  $P_t$  et  $P_2$ , par  $P_{2t}$ ,  $t$  devient une transition locale à  $P_{2t}$ .

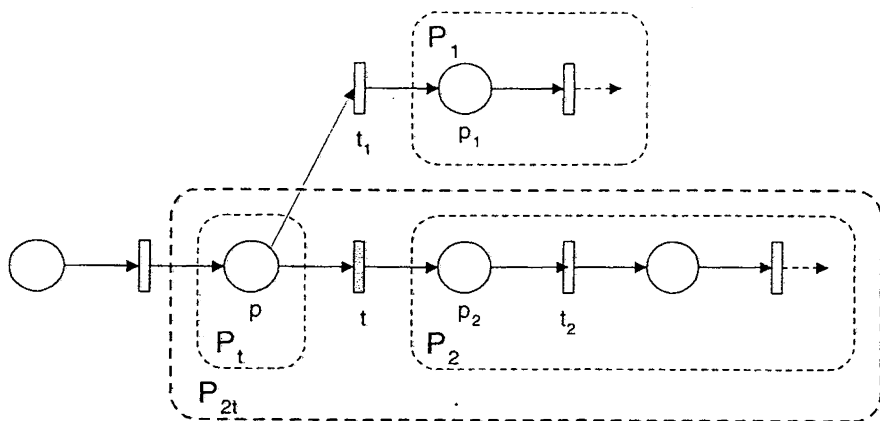


Figure 11. – Rendre locale une transition.

### 5.3. Résolution de la chaîne de Markov

Nous décrivons maintenant les modifications des différentes étapes du cas non immédiat. Nous renvoyons le lecteur à [3, 4] pour une présentation générale du procédé d'extraction de la CTMC incluse d'un GSPN.

La construction du graphe d'accessibilité prend en compte la nouvelle sémantique. nous appellerons encore « tangible » un marquage en lequel il n'y pas de transition immédiate ni Cox immédiate franchissable. Les arcs d'origine non tangible doivent être étiquetés avec les probabilités de transition déduites de la discussion ci-dessus.

#### 5.3.1. Calcul des $\overline{TRG}_i$

$\overline{TRG}_i$  est le graphe des marquages locaux étendus tangibles du sous-réseau  $i$ . Notons que l'on emploie les mêmes descripteurs puisque tous les états sont tangibles.

Les algorithmes sont globalement les mêmes, mais après un franchissement exponentiel ou un  $\alpha$ -franchissement, nous calculons les séquences de franchissement immédiats ou Cox immédiats générés dans chaque sous-réseau : par exemple, dans le réseau de la figure 12, le tir du  $t_{11}$  (exponentielle) produit des franchissements immédiats dans trois sous-réseaux. Quelques séquences possibles sont indiquées sur la figure. Si

$$\overline{M}_i \xrightarrow{t} \overline{M}'_i \xrightarrow{\sigma} \overline{M}_i^{(v)}$$

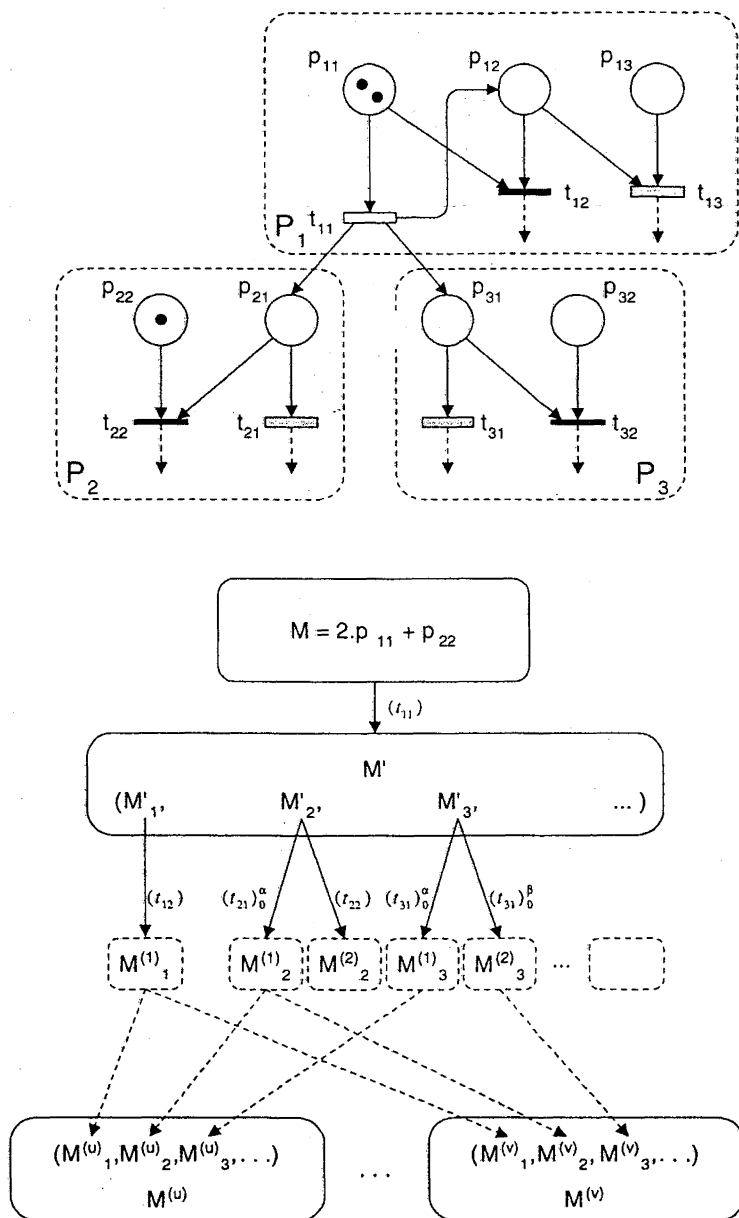


Figure 12. – Franchissement exponentiel suivi de franchissements immédiats.



où  $\sigma$  est une séquence de franchissements immédiats, nous ajoutons à l'étiquette de l'arc  $\overline{M}_i \xrightarrow{t} \overline{M}_i^{(v)}$  la probabilité

$$\sum_{\sigma} \Pr(\overline{M}_i \xrightarrow{\sigma} \overline{M}_i^{(v)})$$

de toutes les séquences de franchissements immédiats entre ces marquages. Bien entendu, les descripteurs des transitions impliquées sont mis à jour pendant ces calculs.

### 5.3.2. Expression tensorielle du générateur

Nous avons là même expression que pour le cas non immédiat. Cependant, les matrices contiennent maintenant les effets des franchissements immédiats : les sommes sur les séquences  $\sigma$  de franchissements immédiats rendent compte de la possibilité de « visiter » une suite d'états transitoires avant d'atteindre le marquage tangible  $\overline{m}'_i$ . Les probabilités correspondantes sont extraites de  $\overline{TRG}_i$ .

Pour les matrices  $Q_i^0$  :

$$\begin{aligned} q_{i, \overline{m}_i, \overline{m}'_i}^0 &= \sum_{\substack{t \in T_i^{(L)} \cap T_E \\ \overline{m}_i \xrightarrow{t} \overline{m}'_i}} \theta(t, \mathbf{m}_i) \sum_{\overline{M}_i'' \xrightarrow{\sigma} \overline{M}'_i} \Pr(\overline{M}_i'' \xrightarrow{\sigma} \overline{M}'_i) \\ &+ \sum_{\substack{t \in T_i^{(L)} \cap T_C \\ \overline{m}_i \xrightarrow{t} \overline{m}'_i}} \mu_r(t) \cdot \alpha_r(t) \sum_{\overline{M}_i'' \xrightarrow{\sigma} \overline{M}'_i} \Pr(\overline{M}_i'' \xrightarrow{\sigma} \overline{M}'_i) \\ &+ \sum_{\substack{t \in T_{C,i} \\ \overline{m}_i \xrightarrow{t} \overline{m}'_i}} \mu_r(t) \cdot \beta_r(t) \end{aligned}$$

Pour les matrices  $C_j(t)$  de l'algorithme 3.1 :

cas 1) première branche sinon :

$$c_j(t)_{\overline{m}_j, \overline{m}'_j} = \begin{cases} \sum_{\overline{M}_j'' \xrightarrow{\sigma} \overline{M}'_j} \Pr(\overline{M}_j'' \xrightarrow{\sigma} \overline{M}'_j) & \text{si } \overline{m}_j \xrightarrow{t} \overline{m}'_j \in \overline{TRG}_j \\ 0 & \text{sinon} \end{cases}$$

cas 1) seconde branche sinon :

$$c_i(t)_{\bar{m}_i, \bar{m}'_i} = \begin{cases} \mu_r \alpha_r \sum_{\bar{M}_i'' \xrightarrow{\sigma} \bar{M}_i'} \Pr(\bar{M}_i'' \xrightarrow{\sigma} \bar{M}_i') & \text{si } \bar{m}_i \xrightarrow[r]{t \alpha} \bar{m}'_i \in \overline{TRG}_i \\ 0 & \text{sinon} \end{cases}$$

Les mêmes modifications sont appliquées dans les deux branches sinon du cas 2).

## CONCLUSIONS ET PERSPECTIVES

Nous avons présenté dans ce papier une nouvelle méthode pour introduire des durées de tir à distribution de Cox et Phase-type dans les réseaux de Petri stochastiques généralisés. Elle autorise la spécification de tous les types de sémantique stochastique habituellement rencontrés, y compris les transitions immédiates et de Cox avec possibilité de franchissement immédiat. Nous avons montré comment définir une décomposition du réseau qui permette de dériver une expression tensorielle du générateur. Cette expression ralentit l'accroissement de complexité induit par les transitions à loi de Cox, le maintenant dans le même ordre de grandeur que celle du graphe d'accessibilité du réseau, lorsque les transitions-Cox sont faiblement couplées avec les autres parties du réseau.

Il ressort de ce travail, comme de tous les travaux fondés sur les méthodes tensorielles (voir par exemple [18]), que si l'accroissement de complexité est maîtrisé par cette approche, une nouvelle difficulté apparaît, conséquence de l'augmentation de la taille des modèles que l'on peut résoudre, à savoir la taille du vecteur de probabilités à l'équilibre lui-même. Des travaux en cours étudient un raffinement de la démarche présentée dans cet article tenant compte de ce problème. Nous analyserons également plus en détail les grains apportés en fonction de la topologie du réseau et évaluerons la possible intégration de la méthode dans un outil comme [9].

Nous souhaiterions cependant souligner l'originalité de notre approche : les travaux précédents s'étaient surtout intéressés au problème de la définition des distributions de Cox dans le contexte des SPNs, sans proposer de moyen de gérer la complexité supplémentaire induite sur la chaîne de Markov sous-jacente. À notre connaissance, notre étude constitue la première tentative pour lier l'introduction de distributions non exponentielles aux techniques de résolution efficaces fondées sur des expressions tensorielles. D'autre part, ces techniques, quoique potentiellement très efficaces, sont habituellement

difficiles à appliquer à des SPNs quelconques. L'utilisation de l'expansion de Cox est un des cas où la partition de l'espace d'états en produit d'espaces presque indépendants est suggérée par le modèle.

#### BIBLIOGRAPHIE

1. M. AJMONE MARSAN, G. BALBO, A. BOBBIO, G. CHIOLA, G. CONTE et A. CUMANI, The effect of execution policies on the semantics and analysis of stochastic Petri nets, *IEEE Transactions on Software Engineering*, July 1989, 15(7), pp. 832-846.
2. M. AJMONE MARSAN, G. BALBO, A. BOBBIO, G. CHIOLA, G. CONTE et A. CUMANI, On Petri nets with stochastic timing, In *Proc. of the International Workshop on Timed Petri Nets*, Torino, Italy, July 1-3 1985, IEEE Computer Society Press, pp. 80-87.
3. M. AJMONE MARSAN, G. BALBO, G. CHIOLA et G. CONTE, Generalized stochastic Petric nets revisited: random switches and priorities. In *Proc. of the International Workshop on Petri Nets and Performance Models*, Madison, Wisconsin, USA, August 24-26 1987, IEEE Computer Society Press, pp. 44-53.
4. M. AJMONE MARSAN, G. BALBO et G. CONTE, A class of generalized of stochastic Petri nets for the performance evaluation of multiprocessor systems, *ACM Transactions on computer systems*, May 1984, 2(2), pp. 93-122.
5. M. AJMONE MARSAN, G. BALBO, G. CONTE, S. DONATELLI et G. FRANCESCHINIS, *Modelling with Generalized Stochastic Petri Nets*, Wiley series in parallel computing, John Wiley & Sons, England, 1995.
6. M. AJMONE MARSAN et G. CHIOLA, On Petri nets with deterministic and exponentially distributed firing times, In G. ROZENBERG, ed., *Advances in Petri Nets 1987*, number 266 in LNCS, Springer-Verlag, 1987, pp. 132-145,
7. P. BUCHHOLZ, Aggregation and reduction techniques for hierarchical GCSPN. In *Proc. of the 5th International Workshop on Petri Nets and Performance Models*, Toulouse, France, October 19-22 1993, IEEE Computer Society Press, pp. 216-225.
8. P. CHEN, S. C. BRUELL et G. BALBO, Alternative methods for incorporating non-exponential distributions into stochastic timed Petri nets. In *Proc. of the thir International Workshop on Petri Nets and Performance Models*, Kyoto, Japan, December 11-13 1989, IEEE Computer Society Press, pp. 187-197.
9. G. CHIOLA, G. FRANCESCHINIS, R. GAETA et M. RIBAUDO, GreatSPN 1.7: graphical editor and analyzer for timed and stochastic Petri nets, *Performance Evaluatin*, 1995, 24(1&2), pp. 47-68.
10. H. CHOI, V. G. KULKARNI et K. S. TRIVEDI, Markov regenerative stochastic Petri nets, *Performance Evaluation*, special issue: Peformance'93, 1994, 20(1-3), pp. 337-357.
11. G. CIARDO, R. GERMAN et C. LINDEMANN, A characterization of the stochastic process underlying a stochastic Petri net, In *Proc. of the 5th International Workshop on Petri Nets and Performance Models*, Toulouse, France, October 19-22 1993, IEEE Computer Society Press, pp. 170-179.
12. D. R. COX, A use of complex probabilities in the theory of stochastic process, In *Proc. Cambridge Philosophical Society*, 1955, pp. 313-319.
13. A. CUMANI, ESP – a package for the evolution of stochastic Petri nets with phase-type distributed transition times. In *Proc. of the International Workshop on Times Petri Nets*, Torino, Italy, July 1-3 1985, IEEE Computer Society Press, pp. 144-151.
14. S. DONATELLI, Superposed generalized stochastic Petri nets: definition and efficient solution. In Robert Valette, editor, *proc. of the 15th International Conference on*

*Application and Theory of Petri Nets*, number 815 in LNCS, Zaragoza, Spain, June 20-24 1994, Springer-Verlag, pp. 258-277.

15. R. GERMAN et C. LINDEMANN, Analysis of stochastic Petri nets by the method of supplementary variables, *Performance Evaluation*, special issue: Performance'93, 1994, 20(1-3), pp. 317-335.
16. S. HADDAD et P. MOREAUX, Evaluation of high level Petri nets by means of aggregation and decomposition. In *Proc. of the 6th international Workshop on Petri Nets and Performance Models*, Durham, NC, USA, October 3-6 1995, IEEE Computer Society Press, pp. 11-20.
17. S. HADDAD, P. MOREAUX, Asynchronous composition of high level Petri nets: a quantitative approach. In *Proc. of the 17th International Conference on Application and Theory of Petri Nets*, number 1091 in LNCS, Osaka, Japan, June 24-28 1996, Springer-Verlag, pp. 193-21.
18. P. KEMPER, Numerical analysis of superposed GSPNs. In *Proc. of the 6th International Workshop on Petri Nets and Performance Models*, Durham, NC, USA, October 3-6 1995, IEEE Computer Society Press, pp. 52-61.
19. M. K. MOLLOY, *On the integration of delay and throughput in distributed processing models*, PhD dissertation, University of California, Los Angeles, CA, USA, September 1981.
20. M. F. NEUTS, *Matrix-geometric solutions in stochastic models – an algorithmic approach*, The John Hopkins University Press, London, 1981.