

Calcul des probabilités de visite dans les chaînes de Markov réductibles selon les composantes fortement connexes

Hélène Le Guen, Raymond Marie

► **To cite this version:**

Hélène Le Guen, Raymond Marie. Calcul des probabilités de visite dans les chaînes de Markov réductibles selon les composantes fortement connexes. [Rapport de recherche] RR-4296, INRIA. 2001. inria-00072291

HAL Id: inria-00072291

<https://hal.inria.fr/inria-00072291>

Submitted on 23 May 2006

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

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

*Calcul des probabilités de visite
dans les chaînes de Markov réductibles
selon les composantes fortement connexes*

Hélène Le Guen et Raymond Marie

N°4296

octobre 2001

————— THÈME 1 —————



*Rapport
de recherche*

Calcul des probabilités de visite dans les chaînes de Markov réductibles selon les composantes fortement connexes

Hélène Le Guen et Raymond Marie*

Thème 1 — Réseaux et systèmes
Projet Armor

Rapport de recherche n° 4296 — octobre 2001 — 14 pages

Résumé : Dans le cadre de l'étude du test de logiciel nous avons été amenés à modéliser le comportement du logiciel par une chaîne de Markov à temps discret (CMTD) et à nous intéresser à la probabilité qu'un état de la chaîne soit visité au cours d'une exécution du logiciel. Pour pouvoir adapter ce calcul à des chaînes comportant un grand nombre d'états, nous avons développé une technique de décomposition basée sur l'utilisation des composantes fortement connexes. Ce rapport présente la méthodologie employée et des exemples numériques démontrant l'efficacité de l'approche.

Mots-clé : Chaîne de Markov, Probabilité de visite, Composante fortement connexe

(Abstract: pto)

* {Hélène.Le_Guen}{Raymond.Marie}@irisa.fr

Visiting probabilities in non irreducible Markov chains with strongly connected components

Abstract: For the purpose of software testing, we have been conducted to make use of Markov Chains for modeling software behaviour, and to look for the probability of visiting an arbitrary state during one execution of the software. In order to be able to cope with large models, we have developed a decomposition technique using the potential existence of strongly connected components. This report presents the proposed methodology based on the utilisation of strongly connected components. Some numerical examples illustrate the efficiency of this approach.

Key-words: Markov Chain, Visiting probability, Strongly connected component

1 Introduction

Dans le cadre de l'étude de mesures d'efficacité de tests de logiciel, nous avons été amené à modéliser le comportement du logiciel par une chaîne de Markov à temps discret (CMTD). A partir du modèle markovien élaboré pour représenter la dynamique des actions que l'environnement peut effectuer sur le logiciel, il est possible de générer des cas de tests représentatifs de l'utilisation future du logiciel. Cette technique a déjà été utilisée dans le passé (*cf.* [WT94, Gut97]) ; en effet elle permet de tester le logiciel de manière satisfaisante tout en permettant l'estimation non-biaisée de la fiabilité.

Nous nous sommes intéressés, en particulier, à la probabilité qu'un état de la chaîne soit visité au cours d'une exécution du logiciel.

Les modèles représentatifs du problème correspondent à des chaînes non irréductibles mais apériodiques et de dimension finie. Aussi les états sont ici désignés par les entiers de 1 à N et de façon générale les chaînes comportent une ou plusieurs classes transitoires ainsi qu'une ou plusieurs classes récurrentes. Le vecteur de distribution initial est supposé connu.

Après un rappel présenté dans le prochain paragraphe sur les méthodes existantes permettant de calculer les probabilités de visite, nous proposons au paragraphe 3 une amélioration de ce calcul basée sur l'utilisation des classes d'équivalence, puis nous illustrons (paragraphe 4) au moyen de résultats numériques le gain obtenu pour des matrices de grandes tailles.

2 Méthodes classiques

On illustrera sur un exemple deux méthodes différentes permettant de fournir ce résultat. Comme dans les autres exemples présentés dans ce rapport la distribution initiale est choisie telle que $\alpha = (1, 0, 0, \dots, 0)$. Cet exemple de chaîne est représenté à la figure 1. Cette chaîne ne possède qu'une seule classe récurrente, réduite à l'état absorbant 8.

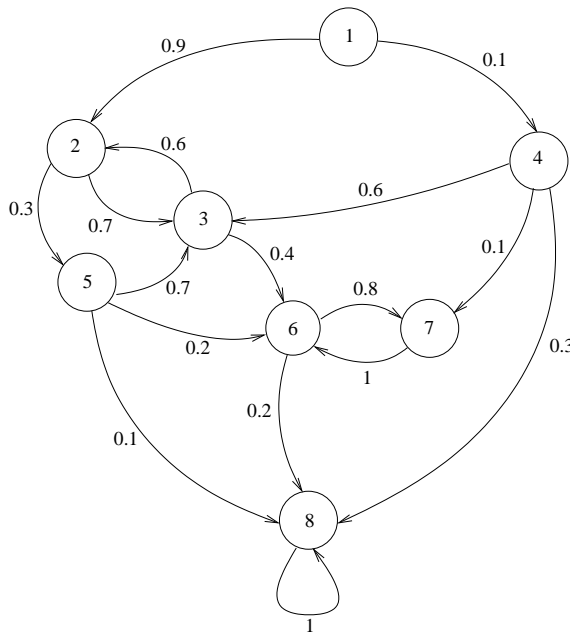


FIG. 1 – Chaîne de Markov

La matrice de transition correspondante s'écrit :

$$\mathbf{P} = \begin{bmatrix} 0 & 0.9 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0.6 & 0 & 0 & 0 & 0.1 & 0.3 \\ 0 & 0 & 0.7 & 0 & 0 & 0.2 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2.1 Première approche

On considère la matrice de transition \mathbf{P} de cette chaîne non irréductible. En adoptant un ordre sur les états qui énumère d'abord les états transitoires puis les états récurrents, \mathbf{P} s'exprime sous la forme suivante :

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{0} & \mathbf{P}_{22} \end{bmatrix}$$

Avec \mathbf{P}_{11} la matrice relative aux états transitoires et \mathbf{P}_{22} la matrice relative aux états récurrents.

Dans cette approche on considèrera que la matrice \mathbf{P} ne comporte que des états transitoires et des états absorbants. Nous traiterons le cas des états appartenant à des classes récurrentes comprenant plusieurs états au paragraphe 2.2.

Considérons tout d'abord un état i transitoire. On note $\mathbf{P}_{11}^{R_i}$ la matrice obtenue en supprimant la ligne et la colonne relatives à l'état i à la matrice \mathbf{P}_{11} ; on désigne par $\mathbf{P}_{11,i}^R$ la colonne de \mathbf{P}_{11} relative à l'état i réduite de sa composante i .

On note α le vecteur ligne des probabilités d'état à l'instant initial, α_{11} le vecteur réduit aux composantes des états transitoires et $\alpha_{11}^{R_i}$ le vecteur réduit de la composante i de valeur α_i et des états absorbants.

On peut alors exprimer la probabilité de visite V_i de l'état i sous la forme :

$$V_i = \alpha_i + \alpha_{11}^{R_i} (\mathbf{I} - \mathbf{P}_{11}^{R_i})^{-1} \mathbf{P}_{11,i}^R \quad (1)$$

Si l'on considère maintenant un état i absorbant, en notant $\mathbf{P}_{12,i}$ la colonne de \mathbf{P}_{12} relative à l'état i , on obtient :

$$V_i = \alpha_i + \alpha_{11} (\mathbf{I} - \mathbf{P}_{11})^{-1} \mathbf{P}_{12,i} \quad (2)$$

Cherchons sur l'exemple de base la probabilité de visite de l'état transitoire 2. La matrice $\mathbf{P}_{11}^{R_2}$ et le vecteur $\mathbf{P}_{11,2}^R$ s'écrivent :

$$\mathbf{P}_{11}^{R_2} = \begin{bmatrix} 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 \\ 0 & 0.6 & 0 & 0 & 0 & 0.1 \\ 0 & 0.7 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{et} \quad \mathbf{P}_{11,2}^R = \begin{bmatrix} 0.9 \\ 0.6 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Sachant que $\alpha = (1, 0, 0, 0, 0, 0, 0, 0)$ on a alors $\alpha_2 = 0$ et $\alpha^{R_2} = (1, 0, 0, 0, 0, 0)$.

Par application de la relation 1 on obtient $V_2 = 0.936$.

Cette première solution est coûteuse en temps de calcul si l'on souhaite calculer les probabilités de visite de tous les états de la chaîne, car elle nécessite N_T inversions de taille $(N_T - 1)$ et une inversion de taille N_T , si N_T désigne le nombre d'états transitoires. Si l'on souhaite connaître les probabilités de visite pour plusieurs états de la chaîne, on choisira alors la méthode décrite dans [Cin75], et expliquée dans le paragraphe suivant.

2.2 Deuxième approche

Soit $r_{i,j}$ l'espérance du nombre de visites à l'état j en partant de l'état i . L'ensemble de ces espérances forme la matrice \mathbf{R} , également nommée la matrice des potentiels. Un résultat de base établit que $r_{i,j} = \sum_{m=0}^{\infty} p_{i,j}^m$ et donc :

$$\mathbf{R} = \sum_{m=0}^{\infty} \mathbf{P}^m \quad (3)$$

Les probabilités de visite se calculent différemment suivant le type des états i et j .

On désigne par $f_{i,j}$ la probabilité conditionnelle de visiter l'état j dans un temps fini sachant que la chaîne est initialement dans un état i .

Tout d'abord, si i est un état récurrent et j un état transitoire, la probabilité d'aller de i en j est alors nulle, et donc $f_{i,j} = 0$.

Si, maintenant, nous considérons i et j deux états récurrents, alors deux cas se présentent : soit ils appartiennent à la même classe de récurrence et donc $f_{i,j} = 1$, soit ils appartiennent à deux classes récurrentes différentes et $f_{i,j} = 0$.

Le cas où i et j sont tous deux transitoires est légèrement plus complexe. Soit \mathbf{S} la sous-matrice de \mathbf{R} relative aux états transitoires, de composantes $s_{i,j}$. En notant \mathbf{A} la matrice $(\mathbf{I} - \mathbf{P}_{11})$, on obtient :

$$\mathbf{S} = \sum_{m=0}^{\infty} \mathbf{P}_{11}^m = \mathbf{A}^{-1}$$

La connaissance de cette matrice \mathbf{S} permet de trouver les probabilités de visite dans ce cas puisqu'il est possible de montrer que :

$$f_{i,j} = \frac{s_{i,j}}{s_{j,j}} \quad (4)$$

Intéressons nous maintenant au dernier cas, celui où i est transitoire et j est récurrent.

Si la chaîne ne comporte qu'une seule classe récurrente, et que l'état j est un élément de cette classe alors $f_{i,j} = 1$. Par contre si la chaîne comporte plusieurs classes de récurrence, il n'est pas possible de conclure immédiatement.

On peut tout d'abord se souvenir que les probabilités de visite des états appartenant à une même classe de récurrence sont égales. Il est donc possible de voir chaque classe d'équivalence comme un macro-état et, de ce fait, de ne pas tenir compte de la topologie à l'intérieur de la classe de récurrence.

On s'intéresse donc à la probabilité de visiter les classes récurrentes à partir de chaque état transitoire. Ceci peut être exprimé à l'aide de la matrice $\hat{\mathbf{P}}$ définie de la façon suivante :

$$\hat{\mathbf{P}} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{b}_1 & \mathbf{b}_2 & \dots & \dots & \dots & \mathbf{b}_m \\ 0 & 1 & 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & 1 & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \quad (5)$$

où les vecteurs \mathbf{b}_j correspondent aux probabilités de visite des classes récurrentes à partir des états transitoires. On note $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m]$. En désignant par \mathbf{b}_h le vecteur correspondant à la classe d'équivalence notée C_h , on a pour chaque état transitoire i :

$$b_h(i) = \sum_{k \in C_h} p_{i,k} \quad (6)$$

Cherchons, tout d'abord, à calculer la $n^{\text{ème}}$ puissance de la matrice $\hat{\mathbf{P}}$. Compte tenu de la structure de $\hat{\mathbf{P}}$, on peut facilement vérifier que $\hat{\mathbf{P}}^n$ s'écrit de la façon suivante :

$$\hat{\mathbf{P}}^n = \begin{bmatrix} \mathbf{P}_{11}^n & \mathbf{B}_n \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (7)$$

avec

$$\mathbf{B}_n = (\mathbf{I} + \mathbf{P}_{11} + \dots + \mathbf{P}_{11}^{n-1})\mathbf{B}$$

Définissons la matrice \mathbf{G} telle que

$$\mathbf{G} = \lim_{n \rightarrow \infty} \mathbf{B}_n \quad (8)$$

alors

$$\mathbf{G} = \left(\sum_{k=0}^{\infty} \mathbf{P}_{11}^k \right) \mathbf{B} = \mathbf{S}\mathbf{B}. \quad (9)$$

Par définition, $G_{i,k}$ est la probabilité asymptotique que partant de l'état i , la chaîne soit dans le macro-état k . Comme cet état est récurrent c'est encore la probabilité, que partant de i , la chaîne visite l'état k .

Ainsi, la probabilité de visite de tout état j appartenant à une classe C_k est exprimée par :

$$f_{i,j} = G_{i,k}. \quad (10)$$

Enfin, disposant des probabilités $f_{i,j}$, la probabilité V_i s'obtient immédiatement par :

$$V_i = \sum_j \alpha_j f_{j,i}. \quad (11)$$

Ces deux approches vues dans cette première partie permettent d'obtenir les probabilités de visite pour chaque état de la chaîne. Les probabilités de visite des états de la chaîne représentée à la figure 1 sont données par le tableau 1.

V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
1	0.936	0.879	0.1	0.4841379	0.9081498	0.7285198	1

TAB. 1 – Probabilités de visite des états de la chaîne de l'exemple

Dans cet exemple, l'état initial étant fixé à l'état 1, on a directement $V_i = f_{1,i}$.

Malheureusement, même cette dernière approche n'est pas adaptée pour résoudre des problèmes de grande taille car il est nécessaire de réaliser une inversion de matrice de taille N_T .

3 Utilisation des classes d'équivalences

Ce paragraphe montre l'intérêt d'une décomposition en classes d'équivalences de la matrice pour le calcul des probabilités de visite V_i .

Dans la méthode de calcul des probabilités de visite V_i présentée au paragraphe 2.2, il est nécessaire d'inverser la matrice \mathbf{A} d'ordre N_T ; ce qui constitue la limite de l'approche tant du point de vue du temps d'exécution que du point de vue de l'espace mémoire. Néanmoins, pour repousser la limite imposée vis à vis de la taille des problèmes traités, nous proposons maintenant une approche où l'inversion de la grande matrice \mathbf{A} est remplacée, au prix de traitements préliminaires, par l'inversion de plusieurs petites matrices. La technique repose sur l'existence supposée de plusieurs classes d'équivalence pour la relation de communication entre les états de la CMTD.

Si la chaîne comporte plusieurs classes récurrentes de cardinalité supérieure à l'unité, il convient de remplacer chaque sous-chaîne récurrente par un macro-état. Les probabilités de visite de chaque état de la sous-chaîne récurrente seront alors égales à la probabilité de visite du macro-état.

3.1 Identification et rangement des classes d'équivalences selon un semi-ordre

On considère les classes d'équivalences $\{C_1, C_2, \dots, C_K\}$ définies sur l'espace d'état par la relation de communication entre les états. La figure 2 montre les différentes classes d'équivalence de l'exemple déjà utilisé.

On considère le semi-ordre tel que $j < i$ signifie que C_j n'est pas accessible à partir de C_i .

En réordonnant les états de façon à respecter le semi-ordre on obtient une matrice de transition triangulaire supérieure par blocs :

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \cdots & \mathbf{P}_{1K} \\ \mathbf{0} & \mathbf{P}_{22} & \mathbf{P}_{23} & \cdots & \mathbf{P}_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{KK} \end{pmatrix} \quad (12)$$

Pour ne pas compliquer inutilement les notations, on décide de renommer les classes selon le semi-ordre. De plus, on fait le choix de construire la matrice de telle sorte que l'on ait d'abord toutes les classes transitoires et ensuite toutes les classes absorbantes.

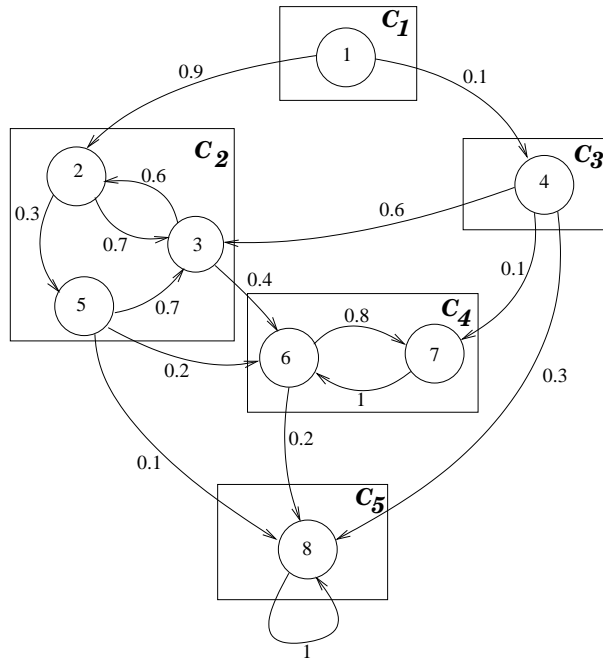


FIG. 2 – Identification des classes d'équivalence

Pour l'exemple, le seul rangement possible selon le semi-ordre est C_1, C_3, C_2, C_4, C_5 . Après rangement selon le semi-ordre, on obtient la matrice de transition suivante :

$$\mathbf{P} = \begin{array}{c|ccc|cc|c} \hline 0 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0.6 & 0 & 0 & 0.1 & 0.3 \\ \hline 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 \\ \hline 0 & 0 & 0.6 & 0 & 0 & 0.4 & 0 & 0 \\ \hline 0 & 0 & 0 & 0.7 & 0 & 0.2 & 0 & 0.1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0.2 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

L'identification des composantes fortement connexes est faite au moyen de l'algorithme de Tarjan [Tar72]. Cet algorithme possède la propriété de ranger les classes selon l'ordre inverse du semi-ordre. Le principe consiste à parcourir la chaîne en profondeur d'abord, en empilant successivement les premiers successeurs avec un marquage des états. Lorsque l'on retrouve dans l'exploration un état déjà présent dans la pile, on en déduit l'existence d'un circuit. Dès l'instant que l'exploration de tous les successeurs des états présents dans ce chemin est terminée, l'ensemble des états appartenant à la composante est identifié. Les composantes sont rangées dans l'ordre de leur découverte. Cet algorithme est linéaire en temps, car chaque état est empilé exactement une fois.

3.2 Probabilités d'entrée dans une classe

On désigne par $\Gamma(k)$, l'ensemble des classes antécédentes à la classe C_k : $C_j \in \Gamma(k)$ si le bloc \mathbf{P}_{jk} n'est pas un bloc nul, $j \neq k$.

On note γ_k le vecteur ligne tel que γ_{kh} désigne la probabilité d'entrer dans la classe C_k par l'état h . Ce vecteur est obtenu par la formule suivante :

$$\gamma_k = \alpha_k + \sum_{j \in \Gamma(k)} \gamma_j (\mathbf{I} - \mathbf{P}_{jj})^{-1} \mathbf{P}_{jk} \quad (13)$$

Où α_k correspond aux composantes du vecteur de distribution initiale α relatives à la classe k .

Si C_j est composée d'un seul état, le vecteur γ_j est un scalaire et il faut remplacer le second terme de l'expression matricielle par le produit $\gamma_j \mathbf{P}_{jk}$ où le vecteur ligne \mathbf{P}_{jk} peut aussi se réduire à un scalaire si la classe C_k ne contient également qu'un seul état.

3.3 Calcul des probabilités de visite

On définit le vecteur colonne \mathbf{e} ayant toutes ses composantes égales à l'unité. Soit \mathbf{q}_k le vecteur ligne tel que q_{kh} désigne la probabilité de quitter la classe C_k à l'étape $(n+1)$ sachant que la chaîne est dans l'état h à l'étape n . On a sous forme matricielle :

$$\mathbf{q}_k = \mathbf{e} - \mathbf{P}_{kk} \mathbf{e} \quad (14)$$

Considérons la CMTD irréductible composée de la classe d'équivalence C_k et d'un état absorbant supplémentaire s relié aux états de C_k selon les probabilités de transition \mathbf{q}_k . Autrement dit, à condition de classer dernier l'état supplémentaire s , la chaîne construite admet la matrice de transition suivante :

$$\widehat{\mathbf{P}}_{kk} = \begin{bmatrix} \mathbf{P}_{kk} & \mathbf{q}_k \\ 0 & 1 \end{bmatrix} \quad (15)$$

Ainsi, pour $k = 3$ et toujours pour l'exemple (C_3 est l'ancienne classe C_2 composée des états 2, 3 et 5), on obtient :

$$\widehat{\mathbf{P}}_{33} = \begin{bmatrix} 0 & 0,7 & 0,3 & 0 \\ 0,6 & 0 & 0 & 0,4 \\ 0 & 0,7 & 0 & 0,3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

La figure 3 illustre la construction de cette matrice $\widehat{\mathbf{P}}_{33}$.

L'étude de cette CMTD permet de déterminer les probabilités conditionnelles de visite d'un état i sachant que l'on est dans la classe d'équivalence C_k . On peut utiliser le même algorithme que celui présenté au paragraphe précédent, à condition de remplacer le vecteur α par le vecteur γ_k .

Ainsi, pour la classe C_3 de l'exemple on a, en utilisant la relation 13, (ici α_3 est nul) :

$$\gamma_3 = \gamma_1 (\mathbf{I} - \mathbf{P}_{11})^{-1} \mathbf{P}_{13} + \gamma_2 (\mathbf{I} - \mathbf{P}_{22})^{-1} \mathbf{P}_{23}$$

soit

$$\gamma_3 = (0,9, 0,06, 0)$$

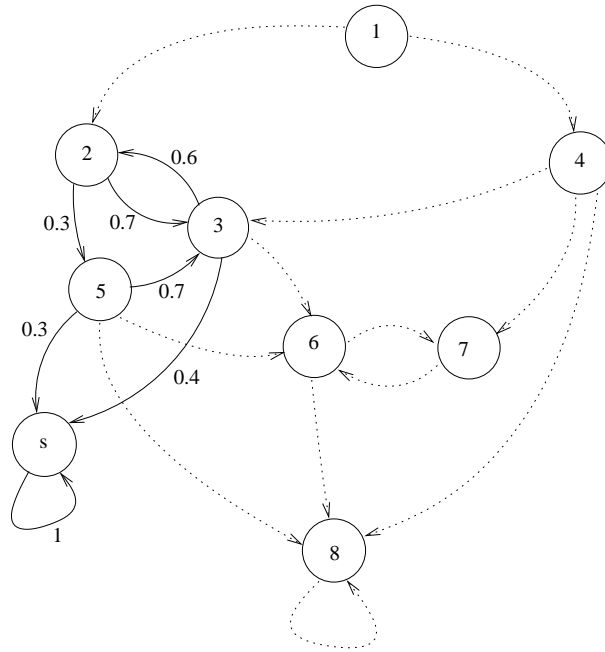


FIG. 3 – Transformation de la classe 3 en chaîne irréductible

De façon similaire à la relation 1, mais en remplaçant la composante du vecteur α par la composante du vecteur γ , on obtient la relation suivante :

$$V_i = \gamma_{3i} + \gamma_3^{R_i} (\mathbf{I} - \mathbf{P}_{33}^{R_i})^{-1} \mathbf{P}_{33,i}^R \quad (17)$$

où la matrice $\mathbf{P}_{33}^{R_i}$ et le vecteur $\mathbf{P}_{33,i}^R$ sont obtenus à partir de la matrice $\widehat{\mathbf{P}}_{33}$. Ce qui nous donne $V_2 = 0.936$, $V_3 = 0.879$ et $V_5 = 0.4841$; on retrouve bien les valeurs obtenues par la méthode globale du paragraphe précédent.

4 Application numérique

4.1 Description des algorithmes

L'algorithme \mathcal{A}_1 correspond à la mise en œuvre de la méthode classique présentée au paragraphe 2.2.

L'algorithme \mathcal{A}_2 utilise la technique de décomposition basée sur la notion de classe d'équivalence. Cet algorithme est plus complexe car il comporte une succession d'étapes différentes. Rappelons que nous avons choisi d'utiliser l'algorithme de Tarjan [Tar72] pour réaliser l'identification des composantes fortement connexes.

Pour comparer les algorithmes, nous avons construit plusieurs chaînes de Markov. Chaque chaîne correspond à un motif qui peut être multiplié à l'infini.

Le premier motif représenté à la figure 4 est formé par plusieurs "lignes", chaque ligne étant constituée de 5 classes d'équivalence, chacune formée de 5 états.

Le second motif, illustré par la figure 5, est une succession de classes transitoires, c'est à dire que lors d'une réalisation du processus toutes les classes de la chaîne sont visitées. Les classes d'équivalence sont toutes composées de 10 états.

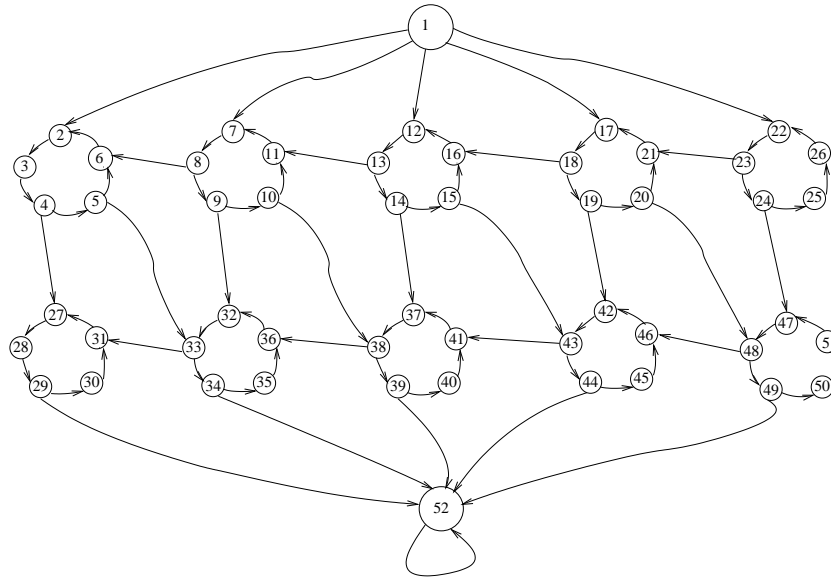


FIG. 4 – Motif 1

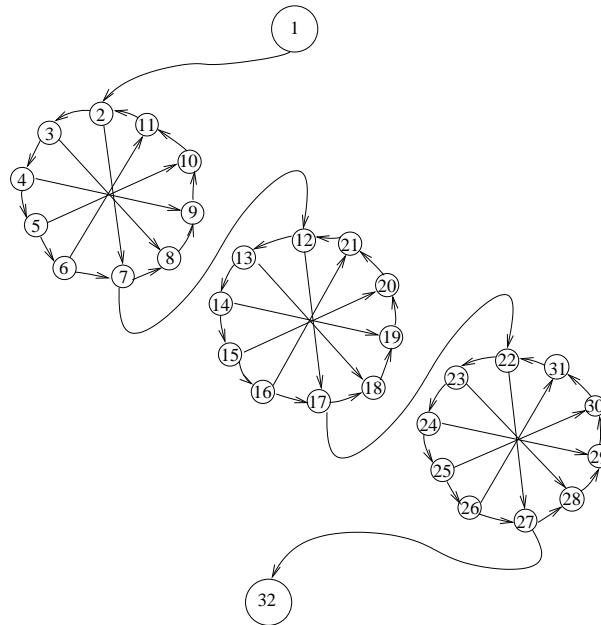


FIG. 5 – Motif 2

Le troisième motif, représenté par la figure 6, est constitué de 10 classes. A la différence des deux motifs précédents, c'est le nombre d'états dans chaque classe qui varie et non plus le nombre de classes.

Ces algorithmes ont été implémentés sous Matlab.

4.2 Détermination des complexités respectives

On désigne à nouveau par N le nombre d'états alors que K désigne ici le nombre de classes d'équivalence.

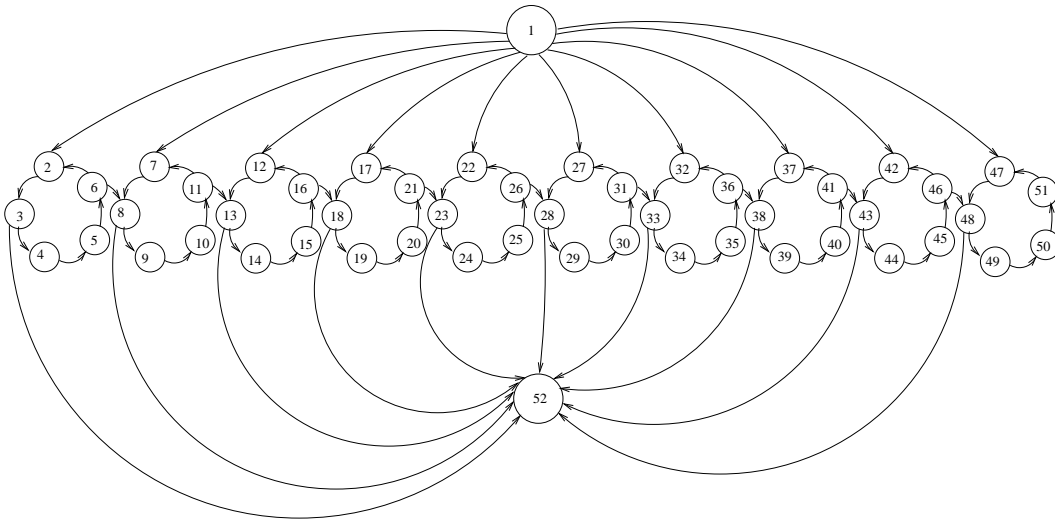


FIG. 6 – Motif 3

On se place dans l’hypothèse où les classes d’une chaîne sont sensiblement de même taille.

L’algorithme \mathcal{A}_1 n’effectue lui qu’une seule inversion de matrice qui correspond à l’opération la plus complexe. Sa complexité est en $O(N^3)$.

L’algorithme \mathcal{A}_2 est de complexité $O\left(\frac{N^3}{K^2}\right)$. En effet le calcul des probabilités d’entrées dans une classe, ainsi que le calcul des probabilités de visite connaissant la classe correspond à K inversions de matrice de taille $\frac{N}{K}$.

4.3 Analyse des résultats

Pour chaque motif et chaque algorithme nous avons noté dans les tables 2, 3 et 4 le temps cpu (en secondes) nécessaire pour calculer les probabilités de visite.

L’algorithme \mathcal{A}_2 étant plus complexe, nous avons choisi de présenter les temps des différentes parties de l’algorithme : c’est à dire la recherche des composantes fortement connexes (C_{tar}), le calcul des probabilités d’entrées dans les classes (C_{gam}) et le calcul des probabilités de visite connaissant la classe (C_{pro}) calculé selon la méthode décrite au paragraphe 2.2. Le total comprend en plus les temps de traitements intermédiaires entre les différents algorithmes.

nb etats	\mathcal{A}_1	\mathcal{A}_2 Total	\mathcal{A}_2		
			C_{tar}	C_{gam}	C_{pro}
252	0.28	1.19	0.23	0.46	0.19
502	1.76	2.27	0.4	0.9	0.42
1002	12.3	4.88	0.86	1.98	0.72
2002	91.24	12.8	2.25	4.71	1.5
3002	310.42	28.13	4.21	8.26	2.18

TAB. 2 – Temps de calcul des 2 algorithmes pour le motif 1

On observe que le temps d’exécution de l’algorithme \mathcal{A}_1 est pratiquement indépendant du motif utilisé. En effet comme nous l’avons vu dans le paragraphe précédent, la taille de la

nb états	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_2		
		Total	C_{tar}	C_{gam}	C_{pro}
252	0.32	0.9	0.21	0.27	0.17
502	1.83	1.82	0.42	0.58	0.29
1002	12.47	3.79	0.94	1.09	0.51
2002	92.47	10.54	2.15	2.75	1.02
3002	308.88	23.69	4.1	4.99	1.52

TAB. 3 – Temps de calcul des 2 algorithmes pour le motif 2

nb états	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_2		
		Total	C_{tar}	C_{gam}	C_{pro}
252	0.4	0.31	0.22	0.1	0.15
502	1.95	1.3	0.34	0.2	0.24
1002	12.41	3	0.75	0.54	0.5
2002	93.69	11.48	2.08	3.15	2.02
3002	315.32	31.24	3.82	8.2	6.12

TAB. 4 – Temps de calcul des 2 algorithmes pour le motif 3

matrice est l'élément déterminant du temps de calcul. L'algorithme \mathcal{A}_2 est quant à lui plus fortement lié au motif. Il est en effet sensible au nombre de points d'entrées dans une classe (pour le calcul des probabilités d'entrée dans une classe γ_k), au nombre moyen d'états dans chaque classe, ainsi qu'au nombre de classes.

En-dessous de 1000 états il semble plus intéressant d'utiliser l'algorithme \mathcal{A}_1 , et de préférer l'algorithme \mathcal{A}_2 lorsque le nombre d'états est supérieur. L'algorithme \mathcal{A}_2 est en effet moins performant pour un nombre faible d'états car il requiert des phases de prétraitement dont le coût prédomine lorsque les problèmes à traiter ne sont pas de taille suffisante.

Cependant, il est important de noter que les motifs utilisés comportent tous plusieurs classes d'équivalence. Bien entendu, dans le cas où la chaîne est constituée d'une composante fortement connexe de forte taille et de quelques composantes fortement connexes de tailles très réduites, l'algorithme \mathcal{A}_2 perd son intérêt.

5 Conclusion

La méthode présentée dans ce rapport permet de calculer de manière efficace les probabilités de visite de chaque état lors de l'exécution du processus d'une chaîne de Markov non irréductible. L'idée principale de la méthode consiste à décomposer le problème global en sous-problèmes correspondants aux composantes fortement connexes, plutôt que d'étudier la chaîne de Markov dans son ensemble.

Cela nécessite de rechercher au préalable les composantes fortement connexes ainsi que les vecteurs des probabilités d'entrée dans ces composantes.

Si la taille du modèle est assez élevée et si les composantes fortement connexes les plus importantes en taille sont multiples et concernent des états transitoires alors cet algorithme est intéressant à la fois en terme de rapidité de calcul, et en terme de besoins en mémoire. De plus,

l'algorithme \mathcal{A}_2 se prête à la parallélisation pour traiter les très gros problèmes sur un système multiprocesseur.

Des applications de ces résultats sont immédiates dans le domaine du test logiciel. En effet, la modélisation du comportement externe d'un logiciel peut être faite sous forme d'une chaîne de Markov, ce qui permet de générer ensuite aléatoirement des séquences de test. Les résultats obtenus peuvent servir à estimer la qualité des tests produits.

Références

- [Cin75] Erhan Cinlar. *Introduction to stochastic Processes*. Prentice Hall, Englewood Cliffs, New Jersey, 1975.
- [Gut97] Walter J. Gutjahr. Importance sampling of test cases in markovian software usage models. *Probability in the engineering and informational sciences*, (11):19–36, 1997.
- [Tar72] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal of Computing*, pages 146–160, June 1972.
- [WT94] James A. Whittaker and Michael G. Thomason. A markov chain model for statistical software testing. *IEEE Transactions on Software engineering*, 20(10), october 1994.



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399