

Estimation de l'erreur de troncature par la méthode de stabilité forte

B. ISSAADI^a, K. ABBAS^b et D. AISSANI^c

Unité de recherche LaMOS (Laboratoires de Modélisation et d'Optimisation des Systèmes)
Université de Bejaïa, Bejaïa 06000, Algérie

Tél. (213) 34 21 51 88,

^a email : issaadi.badredine@yahoo.fr

^b email : karabbas2003@yahoo.fr

^c email : lamos.bejaia@hotmail.com

Résumé Soit $P = (P(i, j))_{i, j \geq 1}$, une matrice stochastique infinie, irréductible et récurrente positive, elle admet donc une distribution stationnaire unique $\pi = (\pi(j))_{j \geq 1}$. Le calcul de cette distribution étant en général difficile sinon impossible, il est souhaitable de disposer d'approximations simples et convergeant rapidement vers cette distribution. Pour cela, une solution consiste à approcher P par une matrice stochastique finie ${}_{(n)}P$.

Dans ce travail, nous traitons l'approximation de la distribution stationnaire pour une chaîne de Markov à espace d'état dénombrable en utilisant la troncature de la matrice de transition.

Mots-clés : Chaînes de Markov, Troncature, Stabilité, Ergodicité uniforme, Ergodicité géométrique, Algorithme, Simulation, Système de files d'attente.

8.1 Application de la méthode de stabilité forte

Considérons un système de files d'attente $M/G/1$. Définissons ainsi un processus stochastique à temps discret : $\{X_n = X(t_n), n = 1, 2, \dots\}$. où t_n est l'instant de départ du n -ième client.

La suite de variables aléatoires $\{X_n, n \geq 1\}$ s'appelle chaîne de Markov induite. Soit P la matrice des probabilités de transition.

Notons par ${}_{(n)}P_0$ et ${}_{(n)}P_n$ les matrices finies des chaînes de Markov induites ${}_{(n)}X_0$ et ${}_{(n)}X_n$ obtenues par augmentation de la troncature de P respectivement à la première et à la dernière colonne.

Nous voulons savoir si le modèle $M/G/1$ définie par la matrice P , peut être estimé par les modèles tronqués définie par les matrices ${}_{(n)}P_0$ et ${}_{(n)}P_n$.

Supposons que le temps de service est réparti suivant une loi exponentielle.

Théorème 8.1 *Supposons que $\lambda/\mu < 1$, alors pour tout β tel que $1 < \beta < \mu/\lambda$, la chaîne de Markov induite définie par la matrice ${}_{(n)}P_0$ est fortement V -stable pour une fonction $V(k) = \beta^k$.*

8.1.1 Déviation du noyau de transition

Lemme 8.1 Soit ${}_{(n)}P_0$ et P les noyaux de transition des chaînes de Markov induites ${}_{(n)}X_0$ et X . Alors, pour tout β tel que $1 < \beta < \beta_0$, on a :

$$\|P - {}_{(n)}P_0\|_V \leq \left(\frac{1}{\beta^{N+1}} + \frac{\mu}{\beta(\lambda + \mu - \beta\lambda)} \right) \left(\frac{\beta\lambda}{\lambda + \mu} \right)^2 = {}_{(n)}\Delta_0.$$

8.1.2 Inégalités de stabilité

Lemme 8.2 Soit π (resp. ${}_{(n)}\pi_0$) la distribution stationnaire de la chaîne induite X (resp. ${}_{(n)}X_0$). Alors, pour tout $1 < \beta < \mu/\lambda$, on a

$$\|{}_{(n)}\pi_0\|_V \leq c_0 = \frac{\beta {}_{(n)}\rho_0}{1 - {}_{(n)}\rho_0} {}_{(n)}\pi_0(0); \quad (8.1)$$

$$\text{avec } {}_{(n)}\rho_0 = \frac{1}{\beta} \left(\frac{\lambda}{\lambda + \mu} \right)^N + \frac{\mu}{\beta(\lambda + \mu - \beta\lambda)} \left(1 - \left(\frac{\beta\lambda}{\lambda + \mu} \right)^N \right).$$

Théorème 8.2 Soit ${}_{(n)}X_0$ (resp. X) la chaîne de Markov induite du système définie par ${}_{(n)}P_0$ (resp. P). Soit ${}_{(n)}\pi_0$ (resp. π) sa mesure invariante. Alors, pour $1 < \beta < \mu/\lambda$, on a l'inégalité :

$$\|\pi - {}_{(n)}\pi_0\|_V \leq c_0 c_{(n)} \Delta_0 (1 - {}_{(n)}\rho_0 - c_{(n)} \Delta_0)^{-1};$$

où c_0 est donné par la formule 8.1, $c = 1 + \|{}_{(n)}\pi_0\|_V$.

Remarque 8.1 De la même façon, nous avons appliqué la méthode de stabilité forte, pour estimer l'erreur de la troncature pour le modèle défini par la matrice de transition ${}_{(n)}P_n$.

8.1.3 Algorithme de stabilité forte

Initialisation : Choisir l'ordre de la troncature N ;
 Introduire le taux d'arrivée λ ;
 Introduire le taux de service μ ;

Début

étape 1
 Vérification de la stabilité;
si $\lambda/\mu \geq 1$ **alors**
 le système est instable;
 Quitter l'algorithme;
sinon
 Déterminer $\beta_0 = \max\{\beta : 1 < \beta < \mu/\lambda \text{ et } (n)\rho_0 < 1\}$;
 Avec $(n)\rho_0 = \frac{1}{\beta} \left(\frac{\lambda}{\lambda + \mu} \right)^N + \frac{\mu}{\beta(\lambda + \mu - \beta\lambda)} \left(1 - \left(\frac{\beta\lambda}{\lambda + \mu} \right)^N \right)$;

fin si

fin étape 1

étape 2
 Déterminer $[\beta_{min}, \beta_{max}]$ dans lequel $(n)\Delta_0 \leq \frac{1 - (n)\rho_0}{c}$;
 Avec;
 $(n)\Delta_0 = \left(\frac{1}{\beta^{N+1}} + \frac{\mu}{\beta(\lambda + \mu - \beta\lambda)} \right) \left(\frac{\beta\lambda}{\lambda + \mu} \right)^2$;
 $c = 1 + \|(n)\pi_0\|_V$;
 $c_0 = \frac{\beta(n)\rho_0}{1 - (n)\rho_0} (n)\pi_0(0)$;
 $\|(n)\pi_0\|_V \leq c_0$;

fin étape 2

étape 3
 Déterminer $\beta_{opt} \in [\beta_{min}, \beta_{max}]$ qui correspond à une erreur d'approximation $\|\pi - (n)\pi_0\|_V$ minimale;
 Avec;
 $\|\pi - (n)\pi_0\|_V \leq \frac{c_0 c(n)\Delta_0}{1 - (n)\rho_0 - c(n)\Delta_0}$;

fin étape 3

Fin

Algorithme 1 : Erreur de $\|\pi - (n)\pi_0\|_V$

De la même façon, nous construisons l'algorithme 2 qui nous donne l'erreur d'approximation de la distribution stationnaire de la chaîne de Markov induite X , par la distribution stationnaire de la chaîne de Markov induite ${}_{(n)}X_n$.

8.2 Application de l'approche sur les chaînes stochastiquement monotones

Nous pouvons facilement remarquer que la chaîne de Markov induite définie par la matrice P est stochastiquement monotone. Soit ${}_{(n)}P_n$ la matrice finie obtenue par augmentation de la troncature de P à la dernière colonne, donc ${}_{(n)}P_n$ est aussi monotone. Soit $V(i) = \beta^i$, où $\beta \geq 1$, Un simple calcul montre que la condition de Lyapunov est vérifiée, c'est à dire,

$$PV \leq \delta V + b\mathbb{1}_0, \quad \delta < 1, b < \infty. \quad (8.2)$$

Donc, la chaîne est géométriquement ergodique, et D'après [3], on a :

$$\|{}_{(n)}\pi - \pi\|_V \leq 4\delta^m b / (1 - \delta) + D \left[{}_{(n)}\pi(n)V(n) + \sum_w {}_{(n)}\pi(w) \left[\sum_j P(w, j)V(j) - \sum_{j \leq n} P(w, j)V(j) \right] \right]$$

où :

$$D = \sum_{s=0}^{m-1} (\delta + b)^s = \left(\frac{\lambda + \mu}{\lambda - \mu} \right) \left(1 - \left(\frac{2\mu}{\lambda + \mu} \right)^m \right).$$

Ainsi,

$$\sum_w {}_{(n)}\pi(w) \left[\sum_j P(w, j)V(j) - \sum_{j \leq n} P(w, j)V(j) \right] \leq \frac{\delta}{2^{n+2}} \sum_w {}_{(n)}\pi(w)V(w)2^w + \left(\frac{\delta + b}{2^{n+1}} - \delta \right) {}_{(n)}\pi(0)$$

Et cela, pour $V(i) = \left(\frac{\lambda + \mu}{2\lambda} \right)^i$, $\delta = \frac{4\lambda\mu}{(\lambda + \mu)^2} = \frac{4\rho}{(1 + \rho)^2}$, et $b = \frac{2\mu}{\lambda + \mu} - \frac{4\lambda\mu}{(\lambda + \mu)^2}$, où $\rho = \lambda/\mu$.

8.2.1 Algorithme d'approximation

Nous avons alors le programme d'estimation suivant :

Nous avons alors le programme d'estimation suivant :

Initialisation : Trouver V , δ , b pour que (8.2) soit vérifiée;

Choisir ε ;

Début

étape 1
Choisir m suffisamment grand tel que $4\delta^m b/[1 - \delta] \leq \varepsilon/2$;

fin étape 1

étape 2
Avec m évaluer $D = \sum_{s=0}^{m-1} [\delta + b]^s$;

fin étape 2

étape 3
Pour chaque ${}_{(n)}P_n$, calculer ${}_{(n)}\pi$ et évaluer;

$${}_{(n)}\pi(n)V(n) + \sum_w {}_{(n)}\pi(w) \left[\sum_j P(w, j)V(j) - \sum_{j \leq n} P(w, j)V(j) \right] \leq \varepsilon/2D.$$

fin étape 3

Fin

Algorithme 3 : Erreur de $\|{}_{(n)}\pi - \pi\|_V$.

8.3 Simulation de l'erreur due à l'approximation par rapport à une norme donnée

Afin de simuler l'écart entre les distributions stationnaires du système idéal et perturbé par rapport à une norme donnée, notre simulateur comportera deux procédures dont l'une permettra de simuler la distribution stationnaire du système idéal et l'autre celle du système perturbé. Après obtention des deux distributions stationnaires, il ne reste qu'à utiliser les paramètres de la norme donnée pour calculer la somme des écarts par rapport à cette dernière.

8.4 Application numérique

Dans cette partie, nous présentons les résultats d'application des algorithmes présentés dans la partie précédente. En faisant varier les valeurs de $\rho = \lambda/\mu$, et N qui est l'ordre de

la troncature.

L'algorithme 3 nous permet de calculer la valeur de N et m pour chaque précision ε .

Application de l'algorithme 1

L'implémentation de l'algorithme 1 et du simulateur pour les mêmes valeurs de V ont permis d'obtenir les résultats représentés dans le tableau suivant :

pour $N = 5$

$\rho = \lambda/\mu$	β_0	β_{min}	β_{max}	β_{opt}	$\ \pi - {}_{(n)}\pi_0\ _V$	Erreur Simulée
0.1	9.9900	1.0400	7.2300	1.7800	0.1082	3.8774e-004
0.2	4.9900	1.1500	3.1000	1.6200	0.8166	0.0073
0.28	3.5614	1.3800	1.8314	1.5600	12.3145	0.0290
0.29	3.4383	1.4900	1.6383	1.5600	116.7656	0.0339
≥ 0.3	—	—	—	—	—	

TAB. 5.1 – Tableau comparatif des erreurs. Cas d'application de l'algorithme 1 et du simulateur pour $N = 5$.

pour $N = 50$

$\rho = \lambda/\mu$	β_0	β_{min}	β_{max}	β_{opt}	$\ \pi - {}_{(n)}\pi_0\ _V$	Erreur Simulée
0.1	9.9900	1.0300	7.1400	1.6700	0.1016	3.0669e-039
0.2	4.9900	1.0900	3.0600	1.5200	0.7079	3.2552e-026
0.28	3.5614	1.2400	1.8514	1.4600	6.0248	8.5993e-020
0.29	3.4383	1.2900	1.7083	1.4500	11.9039	3.5294e-019
0.3	3.3233	1.3800	1.5333	1.4500	87.1995	1.8954e-018
≥ 0.31	—	—	—	—	—	

TAB. 5.2 – Tableau comparatif des erreurs. Cas d'application de l'algorithme 1 et du simulateur pour $N = 50$.

D'après le tableau, nous pouvons remarquer que la condition de stabilité n'est pas vérifiée pour $\rho = \lambda/\mu \geq .3$ lorsque $N = 5$ (ainsi pour $\rho = \lambda/\mu \geq .31$ lorsque $N = 50$). D'où l'impossibilité d'obtenir la borne d'approximation.

De plus, on peut remarquer que l'erreur devient importante lorsque ρ croît.

De même, on remarque que l'augmentation de la capacité N du système, n'apporte pas une grande amélioration sur la borne d'approximation.

On constate aussi, que les erreurs obtenues par le simulateur sont toujours inférieures aux erreurs algorithmiques. Ceci signifie que l'erreur numérique est réellement le seuil de l'erreur qu'on peut faire lors de la perturbation de la capacité du système.

Application de l'algorithme 2

L'implémentation de l'algorithme 2 et du simulateur pour les mêmes valeurs de V ont permis d'obtenir les résultats représentés dans le tableau suivant : pour $N = 5$

$\rho = \lambda/\mu$	β_0	β_{min}	β_{max}	β_{opt}	$\ \pi - {}_{(n)}\pi_n\ _V$	Erreur Simulée
0.1	9.9900	1.0400	7.0700	1.8700	0.1627	2.0697e-005
0.2	4.9900	1.2000	2.8600	1.6600	1.6610	6.3380e-004
0.25	3.9900	1.5100	1.7800	1.6300	57.7339	0.0020
≥ 0.26	—	—	—	—	—	

TAB. 5.3 – Tableau comparatif des erreurs. Cas d'application de l'algorithme 2 et du simulateur pour $N = 5$.

pour $N = 50$

$\rho = \lambda/\mu$	β_0	β_{min}	β_{max}	β_{opt}	$\ \pi - {}_{(n)}\pi_n\ _V$	Erreur Simulée
0.1	9.9900	1.0400	7.1500	1.8700	0.1627	2.1271e-037
0.2	4.9900	1.2000	2.8900	1.6700	1.6539	7.3216e-008
0.25	3.9900	1.5000	1.8000	1.6400	47.1720	8.3804e-005
≥ 0.26	—	—	—	—	—	

TAB. 5.4 – Tableau comparatif des erreurs. Cas d'application de l'algorithme 2 et du simulateur pour $N = 50$.

D'après le tableau, nous pouvons constater que la condition de stabilité n'est pas vérifiée pour $\rho = \lambda/\mu \geq .26$ pour les deux cas $N = 5$ et $N = 50$. En plus, les résultats obtenus par simulateur sont inférieurs à ceux obtenus par l'algorithme 2. Comme nous pouvons remarquer sur les deux tableaux que l'erreur augmente avec l'augmentation de la valeur de ρ . De même, on remarque que l'augmentation de la capacité N du système, n'apporte pas une grande amélioration sur la borne d'approximation. Ceci s'explique par le fait que la déviation du noyau de transition est indépendant du paramètre N . De plus, on remarque que les résultats obtenus par l'algorithme 1 sont meilleurs que ceux obtenus par l'algorithme 2. Cela explique que la méthode d'augmentation de la première colonne est meilleure que celle d'augmentation de la dernière colonne.

Application de l'algorithme 3

L'implémentation de l'algorithme 3 et du simulateur pour la même fonction V et le même ordre n a permis d'obtenir les résultats représentés dans le tableau suivant :

$\rho = \lambda/\mu$	n	m	ε	$\ \pi - {}_{(n)}\pi_n\ _V$	Erreur Simulée
0.1	7	4	0.5	0.4306	1.0833e-006
0.2	11	7	0.5	0.4433	1.2242e-007
0.3	17	11	0.5	0.4365	5.6723e-009
0.4	26	19	0.5	0.4795	5.2518e-010
0.5	43	33	0.5	0.4703	2.9741e-011
0.6	76	62	0.5	0.4802	5.9735e-015
0.7	153	131	0.5	0.4983	6.5757e-018
0.8	366	362	0.5	0.4953	1.2443e-019
0.9	721	486	0.5	0.4942	3.2575e-020

TAB. 5.5 – Tableau comparatif des erreurs. Cas d’application de l’algorithme 3 et du simulateur.

D’après le tableau, nous constatons que pour chaque précision ε , nous pouvons trouver un certain n et m tel que l’erreur d’approximation soit inférieure à cet ε . On remarque aussi qu’à la différence de la méthode de stabilité forte, l’erreur d’approximation est calculable quel que soit la valeur de $\rho < 1$, ce qui montre l’avantage de cette approche.

8.5 Conclusion

Nous avons implémenté des algorithmes et des simulateurs nous permettant de déterminer le domaine d’approximation des caractéristiques stationnaires des systèmes relatifs aux bornes obtenues. Les erreurs d’approximation ont été comparées à celles obtenues par simulation, ce qui nous a permis de valider l’applicabilité de la méthode de stabilité forte ainsi de l’approche sur les chaînes stochastiquement monotones.

Références

1. K. Abbas. Approximation dans les systèmes de files d’attente à serveur non fiable. *Thèse de Doctorat, Université de Bejaia, 2010.*
2. D. Aïssani and N. V. Kartashov. Ergodicity and stability of Markov chains with respect to operator topology in the space of transition kernels. *Compte Rendu Academy of Sciences U.S.S.R, ser.A, (11) : 3-5, 1983*
3. R. L. Tweedie. Truncation approximations of invariant measures for Markov chains. *J. Appl. Probab, (35) : 517-536, 1998*