

Simulation parfaite des chaînes de Markov

Ana Bušić

INRIA - ENS

`http://www.di.ens.fr/~busic/`

`ana.busic@inria.fr`

Master AMIS - UVSQ

Versailles, 21 octobre 2016

Rappels

Nous avons vu :

- ▶ Représentation des chaînes de Markov par des SED.
- ▶ Simulation MCMC.
- ▶ Simulation parfaite par couplage depuis le passé.
- ▶ Cas monotone (et anti-monotone).

Rappels

Nous avons vu :

- ▶ Représentation des chaînes de Markov par des SED.
- ▶ Simulation MCMC.
- ▶ Simulation parfaite par couplage depuis le passé.
- ▶ Cas monotone (et anti-monotone).

Quelques remarques/problèmes :

- ▶ Temps de couplage ?
- ▶ SED non-monotones ?
- ▶ Garder la suite aléatoire en mémoire ?

Temps de couplage

En général un problème très difficile.

Quelques résultats théoriques dans des files d'attente :

- ▶ Une file M/M/1/C. Pire cas : $\lambda = \mu$. $O(C^2)$.
- ▶ Réseaux de Jackson avec les capacités finies :

Pour un réseau acyclique de K files M/M/1/C, on peut montrer que $\mathbb{E}\tau^b \leq \alpha(\lambda, \mu)KC^2$ [Dopper, Gaujal, Vincent, 2006], alors que la taille de l'espace d'état est $N = C^K$.

Critère de Foster

Dans le cas d'un espace d'états \mathcal{X} dénombrable, une condition suffisante pour l'existence d'une distribution stationnaire.

Théorème de Foster. Soit $\{X_t\}_{t \geq 0}$ une chaîne de Markov avec la matrice de transition P irréductible. Si il existe un $\epsilon > 0$, un ensemble fini $F \subset \mathcal{X}$ et une fonction $h : \mathcal{X} \rightarrow \mathbb{R}^+$ telle que :

$$\sum_{j \in \mathcal{X}} p_{i,j} h(j) < \infty, \forall i \in F,$$

et

$$\sum_{j \in \mathcal{X}} p_{i,j} h(j) \leq h(i) - \epsilon, \forall i \notin F,$$

alors $\{X_t\}_{t \geq 0}$ est récurrente positive et possède une unique distribution stationnaire.

Preuve : Chapitre 5 de Pierre Brémaud. Markov Chains : Gibbs Fields, Monte Carlo Simulation, and Queues, Springer, 2008.

Lien avec le temps de couplage

Notation :

$$\tau_F = \min\{t \geq 1 : X_t \in F\}.$$

Lemme. Sous les mêmes hypothèses que dans le théorème de Foster, pour tout $i \notin F$,

$$\mathbb{E}[\tau_F \mid X_0 = i] \leq \frac{h(i)}{\epsilon}.$$

Dans le cas monotone, on peut obtenir une borne supérieure de temps de couplage en étudiant :

- ▶ $F = \{\perp\}$, et $i = \top$ ou
- ▶ $F = \{\top\}$, et $i = \perp$.

Exemple : file M/M/1/C avec $\lambda \neq \mu$ ($\top = C$, $\perp = 0$).

Le cas non-monotone

- ▶ [Kendall 1998] et [Häggeström and Nelander 1998] une idée similaire pour le cas anti-monotone :

$$x \preceq y \Rightarrow x \cdot a \succeq y \cdot a, \forall a \in A.$$

- ▶ [Kendall and Møller 2000] ont introduit une première idée des processus bornants qui enveloppent toutes les trajectoires.
- ▶ [Huber 2004] une idée similaire des processus pour détecter le couplage, mais sans introduire une notion d'ordre sur les états.

Envelope Perfect Sampling Algorithm

[B., Gaujal, Vincent 2008]

- ▶ Hypothèse : (\mathcal{X}, \preceq) est un treillis.
- ▶ Pour $m, M \in \mathcal{X}$, $[m, M] \stackrel{\text{def}}{=} \{x \in \mathcal{X} : m \preceq x \preceq M\}$ un intervalle entre m et M .
- ▶ $\mathcal{I} = \{[m, M] : m, M \in \mathcal{X}, m \preceq M\}$ l'ensemble des intervalles non-vides.
- ▶ Nouvelle fonction de transition $\square : \mathcal{I} \times A \rightarrow \mathcal{I}$ appelée transition enveloppe : pour tout $[m, M] \in \mathcal{I}$ et $a \in A$

$$[m, M] \square a \stackrel{\text{def}}{=} \left[\inf_{m \preceq x \preceq M} \{x \cdot a\}, \sup_{m \preceq x \preceq M} \{x \cdot a\} \right].$$

Extention aux mots finis :

$$[m, M] \square u_{1 \rightarrow n} \stackrel{\text{def}}{=} [m, M] \square u_1 \square u_2 \square \cdots \square u_n.$$

Envelope Perfect Sampling Algorithm

Soient $\perp \stackrel{\text{def}}{=} \inf \mathcal{X}$ et $\top \stackrel{\text{def}}{=} \sup \mathcal{X}$. Le processus $([m_n, M_n])_{n \in \mathbb{N}}$

$$[m_n, M_n] \stackrel{\text{def}}{=} [\perp, \top] \square u_{1 \rightarrow n}$$

est une chaîne de Markov sur l'espace d'états $\mathcal{X} \times \mathcal{X}$, appelée **la chaîne enveloppe**.

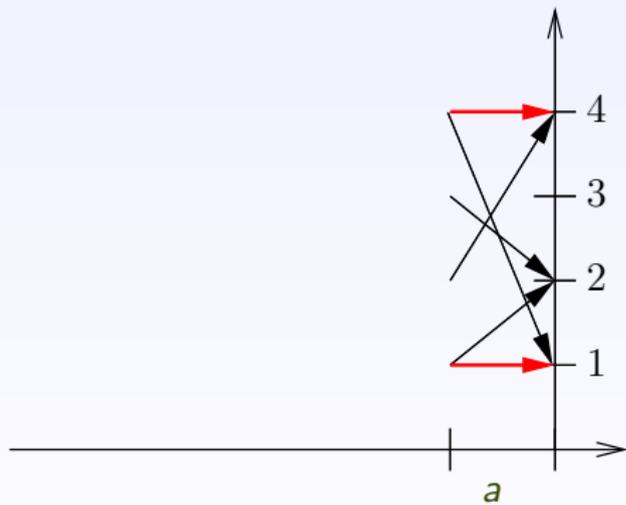
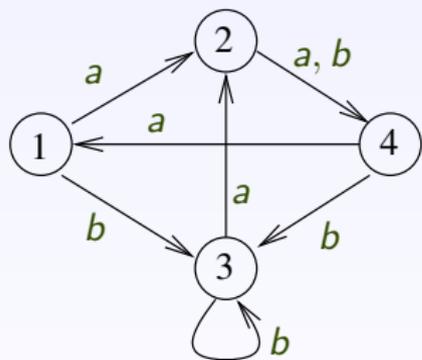
Théorème

Si $[m_n, M_n]$ appartient à $\mathcal{S} = \{[x, x] : x \in \mathcal{X}\}$ pour un n fini :

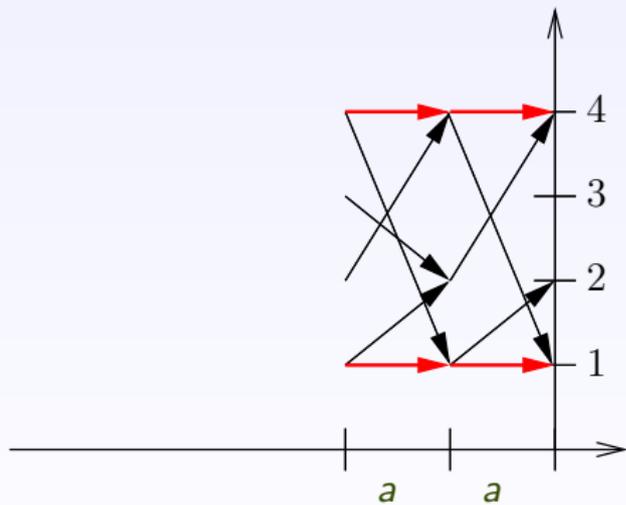
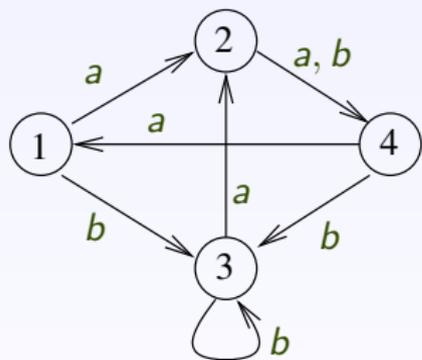
$$\tau_e \stackrel{\text{def}}{=} \min \left\{ n : [\perp, \top] \square u_{-n+1 \rightarrow 0} \in \mathcal{S} \right\},$$

alors τ_e est le temps de couplage depuis le passé de la chaîne enveloppe. L'état défini par $[\perp, \top] \square u_{-\tau_e+1 \rightarrow 0}$ est distribué selon π .

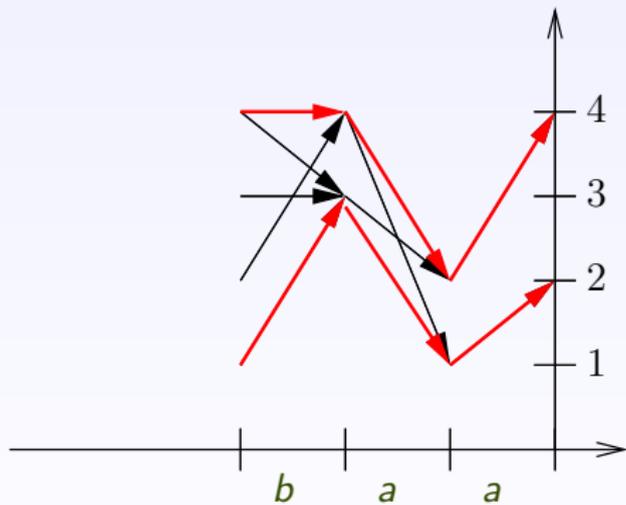
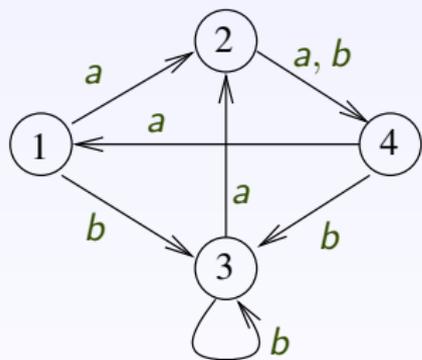
Exemple



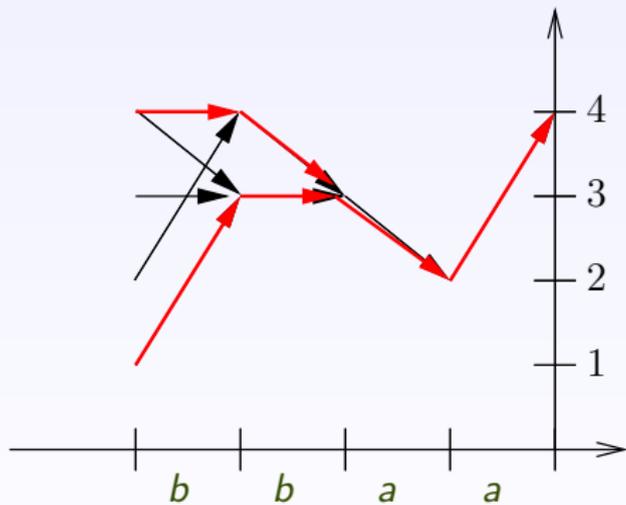
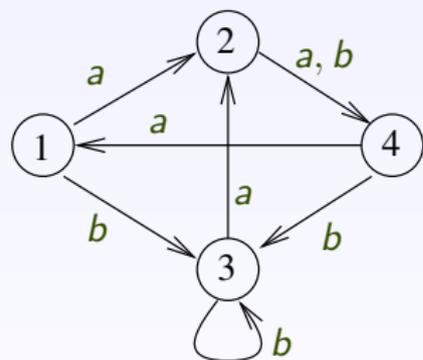
Exemple



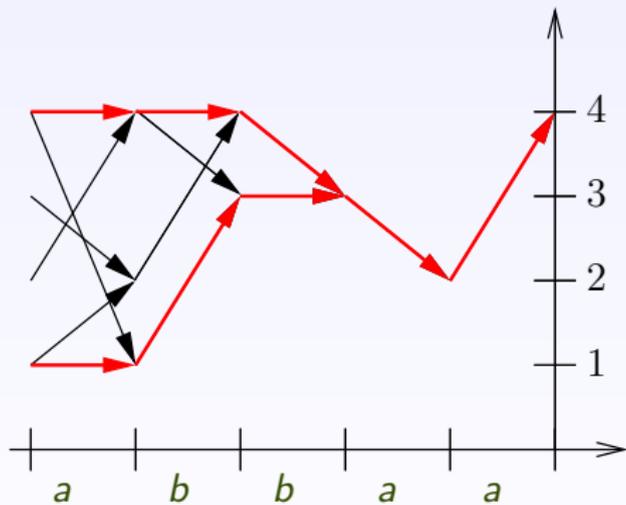
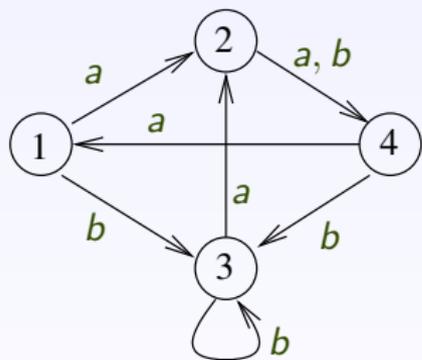
Exemple



Exemple



Exemple



Envelope Perfect Sampling Algorithm (EPSA)

Data: suite i.i.d. $(u_{-n})_{n \in \mathbb{N}} \in A^{\mathbb{N}}$; l'opération enveloppe \square

Result: Un état de \mathcal{X} distribué selon π .

begin

$n = 1$; $m := \perp$; $M := \top$;

repeat

for $i = n - 1$ **downto** 0 **do**

$[m, M] := [m, M] \square u_{-i}$;

$n := 2n$;

until $m = M$;

$x^* := m$;

return x^* ;

Complexité : $O(C_e \times \tau_e)$ (à comparer avec $O(C \times |\mathcal{X}| \times \tau^b)$).

Commentaires

1. L'opération \square peut être remplacée par une sur-approximation \odot telle que pour tout intervalle $[m, M]$,
 $[m, M] \square a \subseteq [m, M] \odot a$, sans modifier la validité de l'algorithme.

Commentaires

1. L'opération \boxplus peut être remplacée par une sur-approximation \odot telle que pour tout intervalle $[m, M]$,
 $[m, M] \boxplus a \subseteq [m, M] \odot a$, sans modifier la validité de l'algorithme.
2. La définition de l'opération enveloppe dépend de la description SED de la chaîne de Markov.

Commentaires

1. L'opération \boxtimes peut être remplacée par une sur-approximation \odot telle que pour tout intervalle $[m, M]$,
 $[m, M] \boxtimes a \subseteq [m, M] \odot a$, sans modifier la validité de l'algorithme.
2. La définition de l'opération enveloppe dépend de la description SED de la chaîne de Markov.
3. Si un événement $a \in A$ est monotone, alors pour tout $m \preceq M$,
 $[m, M] \boxtimes a = [m \cdot a, M \cdot a]$.

Si tous les événements sont monotones (ou anti-monotones), alors EPSA coïncide avec la simulation parfaite classique - cas monotone.

Problèmes

- ▶ Il est possible que les enveloppes ne couplent pas même si les vraies trajectoires couplent.

Problèmes

- ▶ Il est possible que les enveloppes ne couplent pas même si les vraies trajectoires couplent.
- ▶ Quand les enveloppes couplent, leur temps de couplage peut être plus long que celui des trajectoires.

Problèmes

- ▶ Il est possible que les enveloppes ne couplent pas même si les vraies trajectoires couplent.
- ▶ Quand les enveloppes couplent, leur temps de couplage peut être plus long que celui des trajectoires.
- ▶ La complexité du calcul des enveloppes peut être trop grande.
Complexité de EPSA : $O(\mathcal{C}_e \times \tau_e)$.
 \mathcal{C}_e ne peut pas être dans $\Omega(N)$ - sinon on ne gagne rien !

Simulation de files d'attentes non monotones

Dans le cas des files d'attentes, les actions sont "homogènes en espace" par partie (i.e. $x \cdot a = x + v_R$ pour x dans la région R) et on a souvent $\mathcal{C} \sim \mathcal{C}_e$. [B., Gaujal, Vincent, 2008].

La différence entre la méthode classique et celle des enveloppes se joue donc sur la comparaison entre $|\mathcal{X}| \times \tau$ et τ_e .

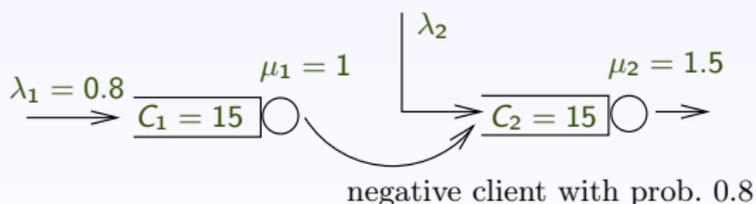


Figure : Un réseaux de files avec clients négatifs.

Exemple

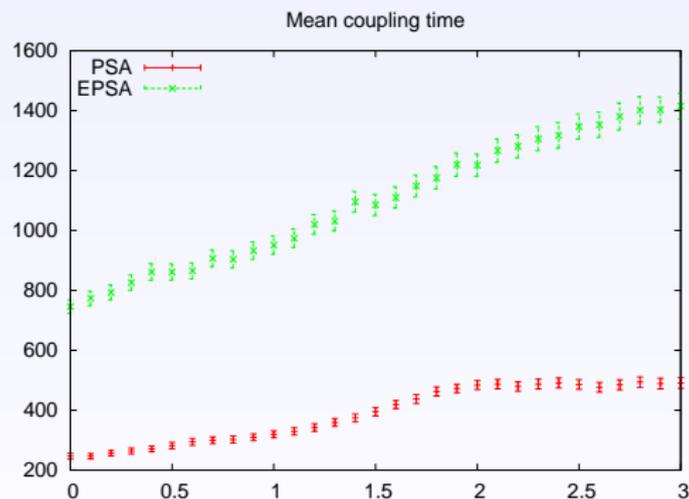


Figure : Temps moyen de couplage en fonction de λ_2 .

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

- ▶ Les événements 'lot' sont non-monotones.
(Exemple : l'arrivée par lot de taille 2 dans une file de capacité C : $C - 2 < C - 1$ mais $C > C - 1$.)

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

- ▶ Les événements 'lot' sont non-monotones.
(Exemple : l'arrivée par lot de taille 2 dans une file de capacité C : $C - 2 < C - 1$ mais $C > C - 1$.)
- ▶ Proposition : EPSA couple si et seulement si les lots de taille 1 peuvent survenir avec une probabilité positive dans chaque file.

Files d'attente avec arrivée par lot (batch)

Les clients arrivent et partent par lot. un nouveau lot est accepté si il y a suffisamment de place dans la file. Sinon l'intégralité du lot est rejeté.

- ▶ Les événements 'lot' sont non-monotones.
(Exemple : l'arrivée par lot de taille 2 dans une file de capacité C : $C - 2 < C - 1$ mais $C > C - 1$.)
- ▶ Proposition : EPSA couple si et seulement si les lots de taille 1 peuvent survenir avec une probabilité positive dans chaque file.
- ▶ Les enveloppes peuvent être calculées en temps constant :

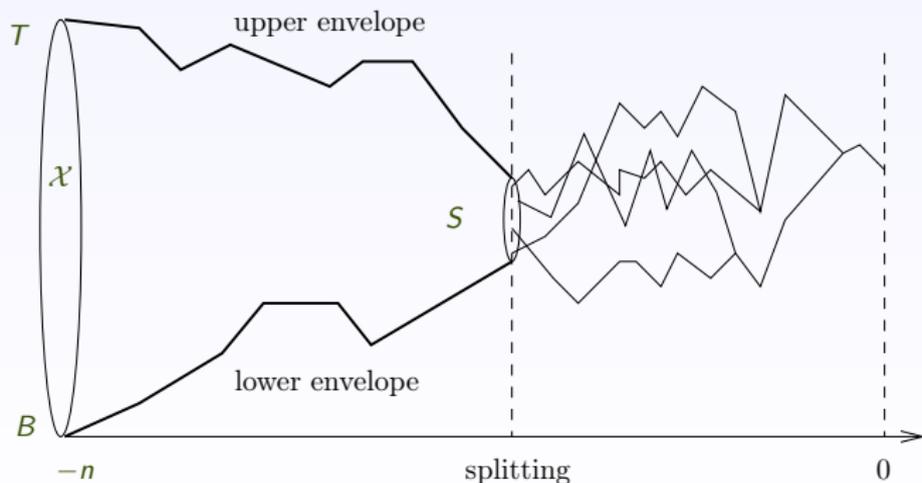
$$m' = \begin{cases} m + k, & M \leq C - k \\ (m + k) \wedge ((C + k - 1) \vee m), & M > C - k. \end{cases}$$

$$M' = \begin{cases} (M + k) \wedge C, & m \leq C - k \\ M, & m > C - k \end{cases}$$

Au delà des enveloppes

Dans le cas où les enveloppes couplent beaucoup plus lentement que les trajectoires elles-mêmes (ou si elles ne couplent pas) :

- ▶ bornes
- ▶ splitting - on peut combiner les avantages des deux méthodes en faisant un éclatement.



Exemple

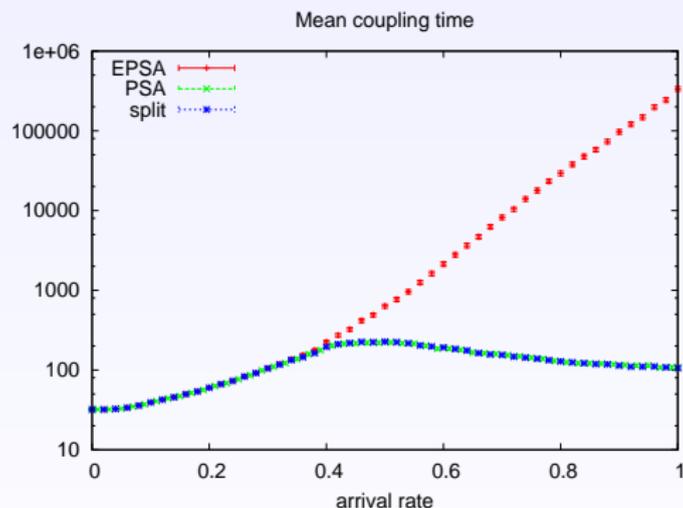
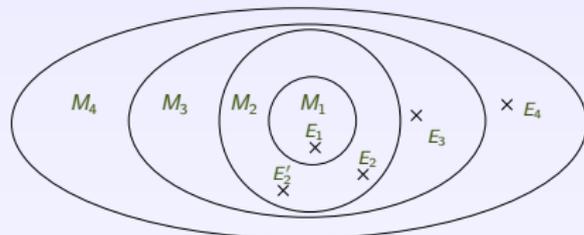


Figure : Temps moyen de couplage de PSA, EPSA et EPSA avec éclatement pour une file $(+2, +3, -1)$.

Classes



Exemples :

Classes :

- ▶ M_1 - monotone MC.
- ▶ M_2 - non-monotone MC, où la simulation parfaite des enveloppes peut être utilisée efficacement.
- ▶ M_3 - enveloppes qui couplent, mais en prenant plus de temps.
- ▶ M_4 - enveloppes qui ne couplent pas (bornes, splitting).

- ▶ E_1 - un réseau de files d'attentes finies avec un routage monotone.
- ▶ E_2 - un réseau comme E_1 avec des clients négatifs.
 E_2' - un réseau comme E_1 avec des noeuds fork/join
- ▶ E_3 - un réseau avec des clients arrivant individuellement et par lot
- ▶ E_4 - un réseau de files d'attentes avec des arrivées par lot de taille supérieure à 2.