

## Simulation parfaite dans un réseau fermé de files d'attente

Anne Bouillard, Ana Bušić, Christelle Rovetta \*

Inria - Département d'Informatique de l'ENS  
23 avenue d'Italie, 75013 Paris, France  
christelle.rovetta@inria.fr

### 1. Modèle

Nous présentons une méthode par chaîne bornante pour la simulation parfaite efficace de réseaux fermés de files d'attente. On considère un réseau fermé de  $K$  files d'attente  $M/1/C$  à  $M$  clients. Chaque file d'attente  $k \in \{1, \dots, K\} := F$  a une capacité  $C_k$  et un taux de service  $\nu_k$ . On note  $p_{i,j}$  la probabilité pour un client venant d'être servi dans la file  $i$  d'être dirigé dans la file  $j$ . Une transition peut s'effectuer si la file  $i$  n'est pas vide et la file  $j$  n'est pas pleine. Le réseau est modélisé par un graphe orienté  $G = (F, R)$  avec  $R = \{(i, j) \mid p_{i,j} > 0\}$ . On le supposera fortement connexe.

On note  $\mathcal{S}$  l'espace des états du système. Ainsi  $\mathcal{S}$  est l'ensemble des  $x = (x_1, x_2, \dots, x_K) \in \mathbb{N}^K$  vérifiant  $\sum_{k=1}^K x_k = M$  et  $\forall k \ 0 \leq x_k \leq C_k$ . Une borne supérieure pour  $|\mathcal{S}|$  est donnée par  $\binom{K+M-1}{K-1}$ . Soit  $(i, j) \in R$ , on définit  $t_{i,j} : \mathcal{S} \rightarrow \mathcal{S}$  la fonction qui décrit le service d'un client de la file  $i$  vers la file  $j$  :

$$t_{i,j}(x) = \begin{cases} x - e_i + e_j & \text{si } x_i > 0 \text{ et } x_j < C_j, \\ x & \text{sinon.} \end{cases}$$

L'évolution de ce système peut être décrite par une chaîne de Markov et être échantillonnée grâce à l'algorithme 1 de simulation parfaite (voir [1]).

La taille exponentielle de l'espace des états et la non-monotonie de la chaîne rendent l'algorithme 1 inutilisable en pratique. On propose dans ce résumé une chaîne bornante sous forme de diagramme afin de pouvoir appliquer la simulation parfaite pour  $K$  et  $M$  grands. On exploite la contrainte forte du nombre de clients fixé à  $M$ , les états sont représentés dans le diagramme par des chemins partant de  $(0, 0)$  et allant à  $(K, M)$ .

### 2. Représentation par un diagramme

#### 2.1. Diagramme

On appelle *diagramme complet* un graphe orienté  $\mathcal{D} = (N, A)$ . Les nœuds sont placés dans une grille

\*. Auteure correspondante.

---

#### Algorithm 1: Simulation parfaite avec $\mathcal{S}$

---

**Data:**  $(U_{-n} = (i_{-n}, j_{-n}))_{n \in \mathbb{N}}$  suite i.i.d.

**Result:**  $s \in \mathcal{S}$  suivant la distribution stationnaire

```

1 begin
2   n ← 1;
3   t ← tU-1;
4   while |t(S)| ≠ 1 do
5     n ← 2n;
6     t ← tU-n ∘ ⋯ ∘ tU-1;
7   end
8   return s0 = {t(S)}
9 end

```

---

à  $(K + 1)$  colonnes et  $(M + 1)$  lignes.

•  $N = \{(c, \ell), \mid c \in \{1, \dots, K - 1\}, \mid \ell \in \{0, \dots, M\}\} \cup \{(0, 0)\} \cup \{(K, M)\}$ .

•  $A = \{(c - 1, \ell), (c, \ell') \mid 0 \leq \ell' - \ell \leq C_c\}$ .

On note  $\Pi(\mathcal{D})$  l'ensemble de tous les chemins de  $(0, 0)$  à  $(K, M)$ . Ce sont des chemins de longueur  $K$ . On définit la fonction  $f : \mathcal{S} \rightarrow \Pi(\mathcal{D})$  qui à chaque élément  $x = (x_1, \dots, x_K) \in \mathcal{S}$  associe un chemin  $f(x) = ((0, 0), (1, x_1), \dots, (c, \sum_{i=1}^c x_i), \dots, (K, M))$ . La fonction  $f$  est une bijection entre  $\mathcal{S}$  et  $\Pi(\mathcal{D})$ .

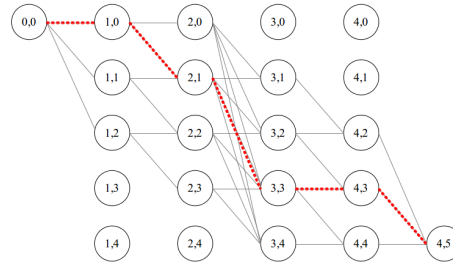


FIGURE 1 – Chemins dans un diagramme complet à 5 files, 4 clients avec  $C_1 = C_5 = 2$ ,  $C_2 = C_4 = 1$  et  $C_3 = \infty$ . Le chemin en pointillés représente l'état  $x = (0, 1, 2, 0, 1)$ .

On dit que  $D = (N, A)$  ou  $A \subseteq \mathcal{A}$  est un *diagramme* si  $(0, 0)$  est le seul nœud ayant un degré entrant égal à 0 et  $(K, M)$  est le seul nœud ayant un degré sortant égal à 0. Un diagramme peut être décrit par l'ensemble  $\Pi$  de ses chemins, on le note alors  $D = (N, A_\Pi)$ . À partir d'un diagramme on peut définir un ensemble d'états et inversement :

$$\phi(\mathcal{S}) = (N, A_{f(\mathcal{S})}) \quad \text{et} \quad \psi(\mathcal{D}) = f^{-1}(\Pi(\mathcal{D})).$$

Si  $S = \psi(\mathcal{D})$ , on dit que  $\mathcal{D}$  est un *représentant* de  $S$ . Si  $S \subseteq \psi(\mathcal{D})$ , on dit que  $\mathcal{D}$  est un *sur-représentant* de  $S$ . Le diagramme complet  $\mathcal{D}$  est un représentant de  $\mathcal{S}$ .

## 2.2. Transitions

Soit  $(i, j) \in R$ . On définit la fonction  $T_{i,j}$  par :

$$T_{i,j}(D) = \phi \circ t_{i,j} \circ \psi(D) \text{ pour } D \subset \mathcal{D}.$$

**Proposition 1** (i) Si  $D$  est un sur-représentant de  $S$  alors  $T_{i,j}(D)$  est un sur-représentant de  $t_{i,j}(S)$ .  
(ii) Si  $|\psi(D)| = 1$  alors  $|\psi \circ T_{i,j}(D)| = 1$ .

Pour effectuer une transition  $T_{i,j}$  sur un diagramme, on sépare ses arêtes en trois sous-ensembles :

- $\mathcal{V}ide = \{a \in A \mid a \in w \in \Pi(D) \text{ et } f^{-1}(w)_i = 0\}$ ,
- $\mathcal{P}lein = \{a \in A \mid a \in w \in \Pi(D) \text{ et } f^{-1}(w)_j = C_j\}$ ,
- $\mathcal{A}ctif = \{a \in A \mid a \in w \in \Pi(D), f^{-1}(w)_i > 0 \text{ et } f(w)_j < C_j\}$ .

Les ensembles  $\mathcal{V}ide$  et  $\mathcal{P}lein$  contiennent les arêtes ne pouvant pas subir la transition  $T_{i,j}$ . L'ensemble  $\mathcal{A}ctif$  a contrario, contient les arêtes pouvant subir la transition  $T_{i,j}$ . On note  $\mathcal{A}ctif'$ , l'ensemble des arêtes du diagramme  $T_{i,j}(D)$  ayant subi la transition  $T_{i,j}$ . Il est obtenu à partir de  $\mathcal{A}ctif$  en abaissant ou en remontant les arêtes des colonnes comprises entre  $i - 1$  et  $j - 1$ . On obtient ensuite :

$$T_{i,j}(D) = (N, A') \text{ avec } A' = \mathcal{V}ide \cup \mathcal{P}lein \cup \mathcal{A}ctif'.$$

Une transition  $T_{i,j}(D)$  s'effectue avec une complexité en temps en  $O(KM^2)$ .

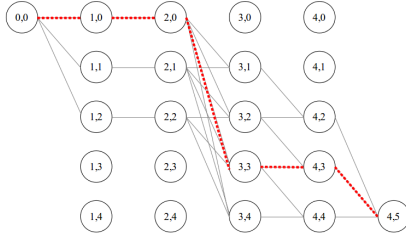


FIGURE 2 –  $T_{2,3}(D)$

## 3. Simulation parfaite & diagramme

**Théorème 1** Il existe une suite finie de transitions  $T = T_{i_p, j_p} \circ \dots \circ T_{i_1, j_1}$  telles que  $|\Pi(T(\mathcal{D}))| = 1$ .

Le théorème 1 prouve que l'algorithme 2 se termine. La proposition 1 assure qu'avec une même suite  $(U_{-n})$  si  $|\Pi(T(\mathcal{D}))| = 1$  alors  $w_0 = \{\Pi(T(\mathcal{D}))\} = \{t(S)\}$  ce qui implique que  $w_0$  est distribué selon la loi stationnaire.

On appelle *temps de couplage* le nombre  $n$  permettant de terminer l'algorithme de simulation parfaite. La figure 3 donne le temps de couplage moyen pour 200 simulations des deux algorithmes pour un même réseau. L'expérimentation avec l'algorithme 1 ne peut pas en temps raisonnable (moins de douze heures) dépasser  $M = 23$ .

---

## Algorithm 2: Simulation parfaite avec $\mathcal{D}$

---

**Data:**  $(U_{-n} = (i_{-n}, j_{-n}))_{n \in \mathbb{N}}$  suite i.i.d

**Result:**  $w \in \Pi(\mathcal{D})$

---

```

1 begin
2   n ← 1;
3   T ← TU-1;
4   while |Π(T(D))| ≠ 1 do
5     n ← 2n;
6     T ← TU-n ∘ ⋯ ∘ TU-1;
7   end
8   return w0 = {Π(T(D))}
9 end

```

---

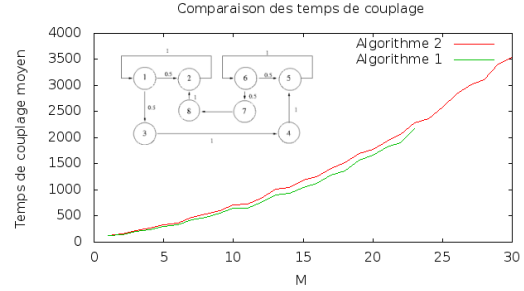


FIGURE 3 –  $K = 8$  et  $C = \frac{M}{2}$

## 4. Conclusion

La principale contribution est la description d'une nouvelle technique de simulation parfaite pour les réseaux fermés de files d'attente. Notre approche permet d'échantillonner la distribution stationnaire pour des réseaux ayant des files à capacité finie ou infinie, à serveurs uniques ou multiples. La démonstration de bonnes propriétés du diagramme impliquera que les transitions  $T_{i,j}(D)$  peuvent s'effectuer en  $O(KM)$ . D'autre part, les simulations montrent que les temps de couplage sont assez proches. Une avancée majeure serait de trouver une borne théorique donnant le temps de couplage de l'algorithme 2 en fonction de celui de l'algorithme 1.

## Bibliographie

1. Propp, J.G. and Wilson, D.B. – Exact sampling with coupled Markov chains & applications to statistical mechanics. – Random Structures & Algorithms 9 (1996), no. 1-2, 223-252.