

# Simulateur à événements discrets

Floriane Dardard, Pablo Rauzy

ENS

21 mai 2010

## Les postulats

- ▶ Il y a trop de sources d'information sur le net,
- ▶ avec beaucoup de redondance,
- ▶ des différences de qualité énorme dans le traitement d'un même sujet.

## La question

Quelle stratégie de sélection adopter pour qu'un surfeur trouve rapidement ce qui l'intéresse dans la multitude de sites web ?

Étant donné un corpus de sites web, dont on ne sait rien a priori, on utilise un algorithme qui suggère au surfeur un site web susceptible de l'intéresser.

À chaque proposition, le surfeur répond “approve” si le site l'intéresse, ou “next” dans le cas contraire, et l'algorithme suggère alors un nouveau site web.

Quand le surfeur “approve” un site web, le système adapte sa stratégie de proposition.

On prétend que cet algorithme nous permet de rapidement et sûrement converger vers un très bon rendement.

## Définition : session

On appelle *session* une utilisation du système : une session démarre lors de la première suggestion du système et se termine lorsque le surfeur “approve” un site web.

## Définition : rendement

Soit  $Y$  le nombre de “next” lors d’une session.

- ▶ Le rendement d’une session est  $\frac{1}{(Y+1)^2}$ .
- ▶ Le *rendement* du système est le rendement moyen d’une session.

# Le simulateur

## Simulation du comportement du surfeur

On attribue au surfeur un vecteur de probabilité correspondant à ses intérêts. On modélise les sites par un vecteur de même type correspondant à sa stratégie de publication.

Soit  $T$  le nombre de sujets pris en considération dans l'algorithme,  $S_{w,i}$  la probabilité que le site  $w$  parle du sujet  $i$ , et  $U_i$  la probabilité que le surfeur soit intéressé par le sujet  $i$ .

La distance entre le surfeur et le site  $w$  est :

$$d_U(w) = \sum_{i=0}^T |U_i - S_{w,i}|$$

On remarque que  $d_U(w)$  est compris entre 0 et 2, le surfeur “approve” le site web si cette distance est inférieure à un certain seuil + un peu d'aléatoire.

Toutes ces informations sont inconnues de l'algorithme

# Le simulateur

## Mise à jour de la stratégie de proposition

À la session  $k$  on note :

- ▶  $\vec{x}(k)$  la stratégie de proposition de l'algorithme
- ▶  $Y(k)$  le nombre de "next"
- ▶  $w_k$  le site accepté par le surfeur
- ▶  $\gamma(k-s)$  le facteur de gain qui traduit combien on modifie la stratégie à chaque session. Si  $s$  est la date de la dernière saute d'humeur, il "remet à zéro" le gain au temps  $s$  ( $\gamma$  est décroissante).

On prend  $\gamma(i) = \frac{1}{\log(i+3)}$ .

À la fin de la session, la stratégie est mise à jour selon la formule

$\vec{x}(k+1) = G(\vec{x}(k) + \gamma(k) \cdot \vec{g}(k))$  où  $g_w(k) = -\frac{Y(k) \cdot \left( \frac{1}{(Y(k)+2)^2} - \frac{1}{(Y(k)+1)^2} \right)}{x_w(k)}$  si  $w = w_k$  et 0 sinon.

Et  $G$  est une renormalisation du vecteur.

# Le simulateur

Entrées / sorties

## Paramètres du simulateur

- ▶ le nombre de topics
- ▶ le nombre de sites web du corpus
- ▶ une distance max entre les intérêts du surfeur et la stratégie de publication d'un site
- ▶ le nombre de sessions à simuler

## Sorties

Le simulateur nous donne l'évolution du rendement du système après chaque nouvelle session.

# Le simulateur

## Passage rapide sur l'implémentation

Notre simulateur est codé en C, de manière orienté objet ce qui a permis un code clair, simplement organisé et très court ( $< 500$  LoC, en comptant les lignes ne contenant qu'une accolade et les directives préprocesseur).

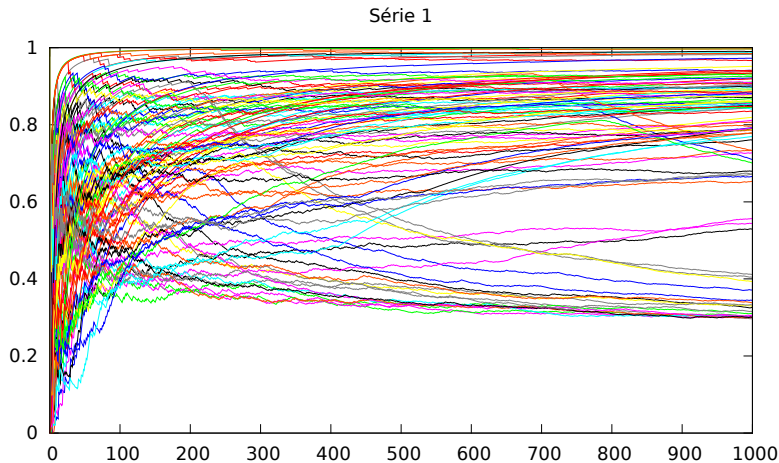
Les modules sont les suivants :

- ▶ **Surfeur** : stockage des informations relatives au surfeur.
- ▶ **Website** : représentation d'un site web.
- ▶ **Corpus** : conteneur de la liste des sites web.
- ▶ **Session** : suggestion des sites web au surfeur, gestion des actions "next" et "approve", de la mise à jour de la stratégie de proposition.
- ▶ **Simulation** : effectue les simulations.

Nous allons maintenant voir et interpréter les résultats de simulations, avec des paramètres différents.

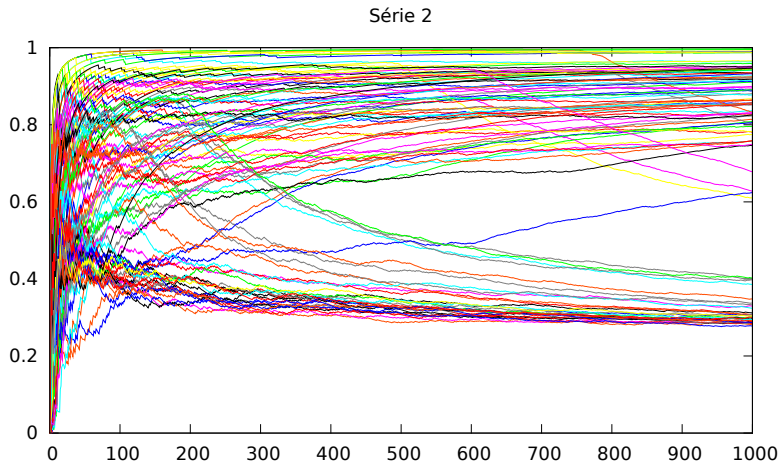
# Simulation : série 1

1000 sessions, 1000000 sites web, distance max à 0.42, 20 topics.



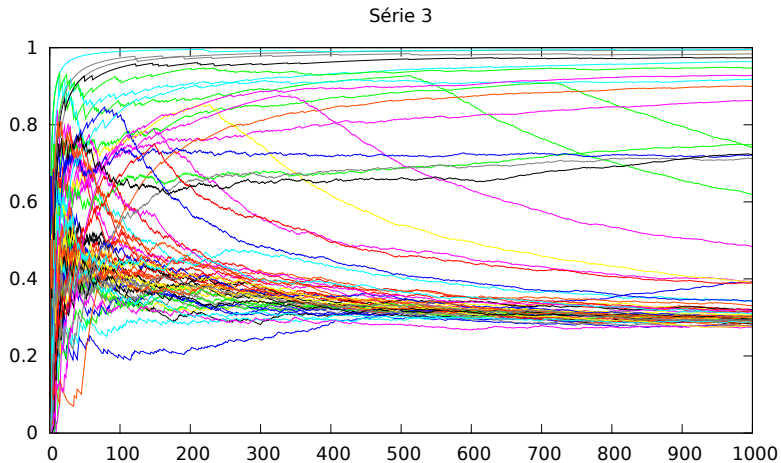
# Simulation : série 2

1000 sessions, **100000** sites web, distance max à 0.42, 20 topics.



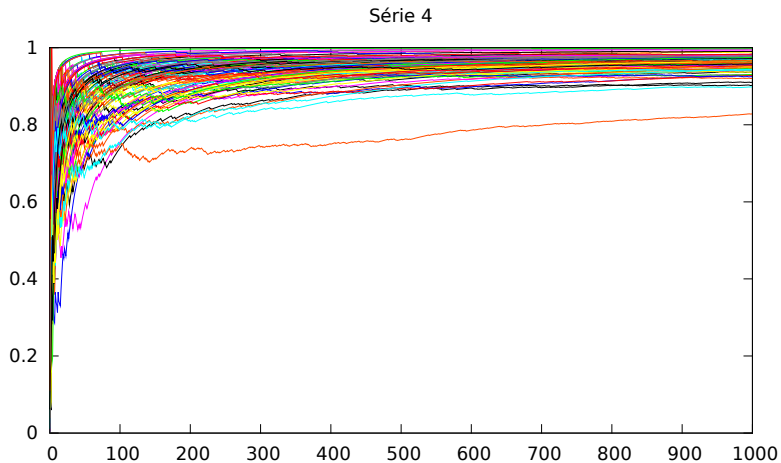
# Simulation : série 3

1000 sessions, 100000 sites web, distance max à **0.35**, 20 topics.



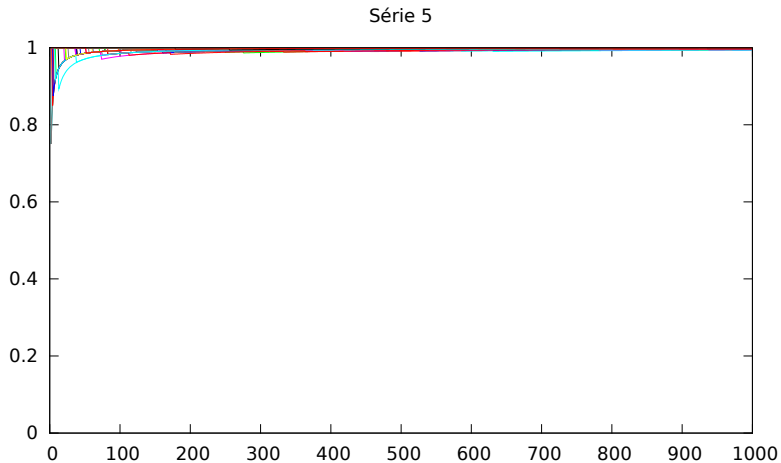
# Simulation : série 4

1000 sessions, 100000 sites web, distance max à **0.6**, 20 topics.



# Simulation : série 5

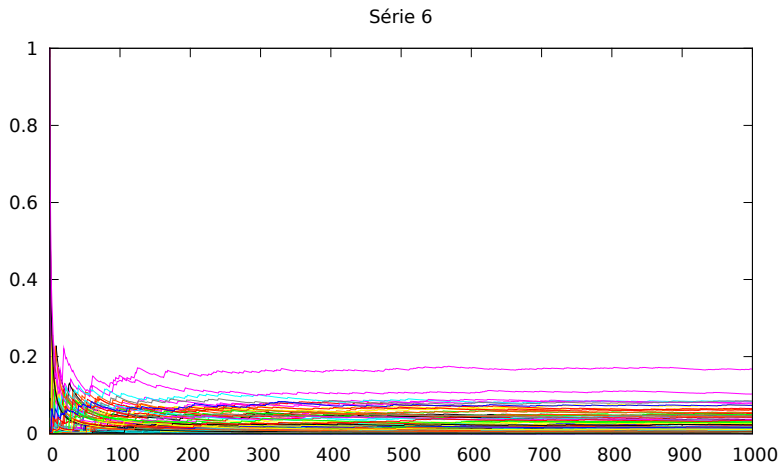
1000 sessions, 100000 sites web, distance max à **1.0**, 20 topics.



# Simulation : série 6

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

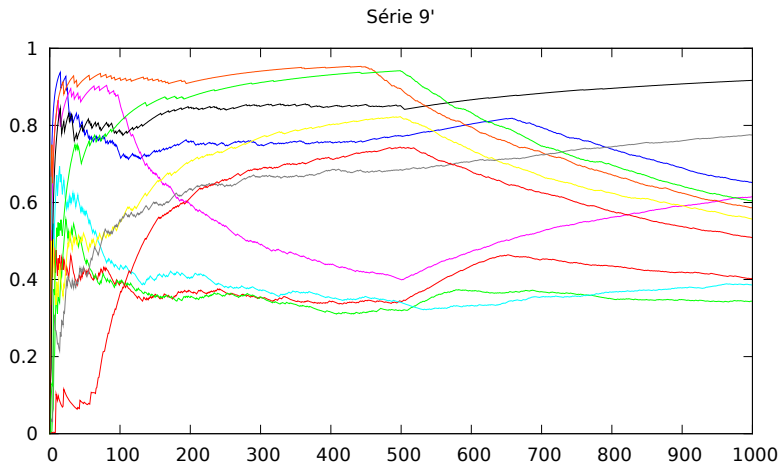
Pas de mise à jour de la stratégie de propositions des sites web.



# Simulation : série 9'

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

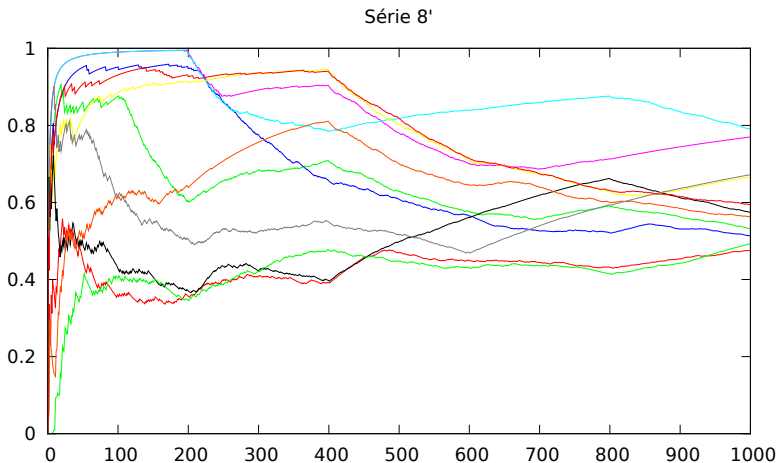
Ajout d'une saute d'humeur à la 500ème session.



# Simulation : série 8'

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

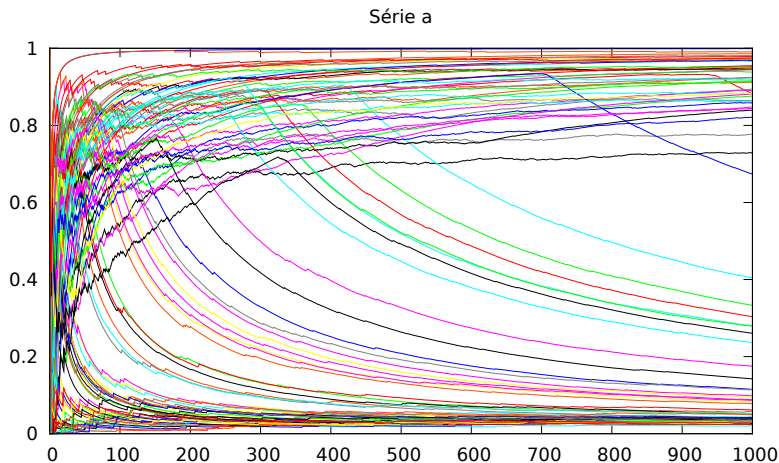
Ajout d'une saute d'humeur toutes les 200 sessions.



# Simulation : série a

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

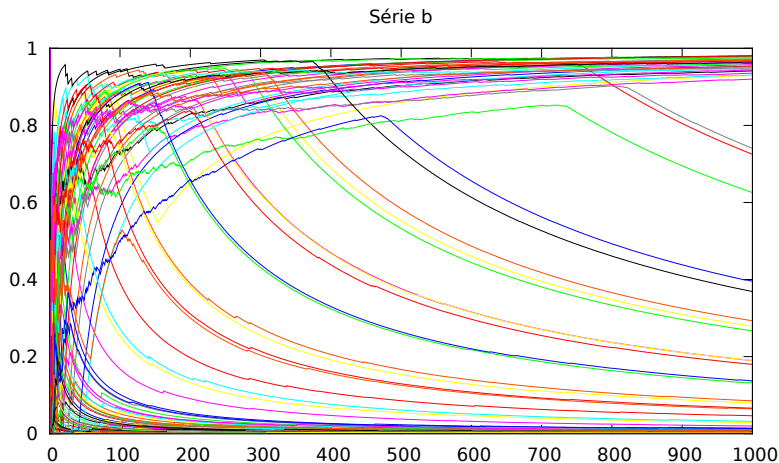
Facteur de gain constant égal à 0.01.



# Simulation : série b

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

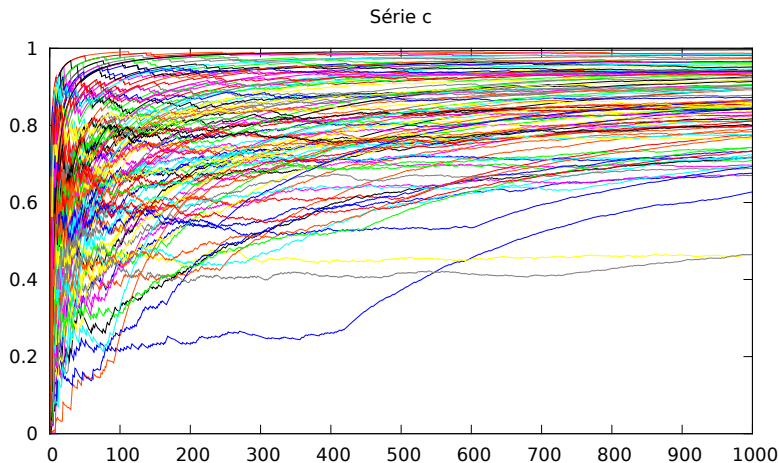
Facteur de gain constant égal à 0.001.



# Simulation : série c

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

Facteur de gain constant égal à 100.



# Évaluation de la confiance

Moyenne, écart-type et fonction de répartition

On peut considérer qu'à chaque fois qu'un utilisateur se connecte, l'algorithme réussit ou ne réussit pas à avoir un bon rendement (disons  $>0,6$ , parce que les rendements sont éparpillés en deux groupes).  $n$  utilisateurs qui se connectent constituent donc une loi binomiale. On veut savoir quel est son paramètre  $p$  de réussir.

On a fait beaucoup de simulations, donc on peut modéliser l'expérience par une loi normale. On va calculer l'intervalle de confiance de la proportion  $p$  d'utilisateurs pour qui l'algorithme a effectivement marché au bout de 1000 sessions.

On a trouvé expérimentalement  $p' = 1 - F(0, 6)$ . L'intervalle de confiance à 95% est donné par la formule  $[p' \pm 1,96 \times \sqrt{p' \times (1 - p')/1000}]$

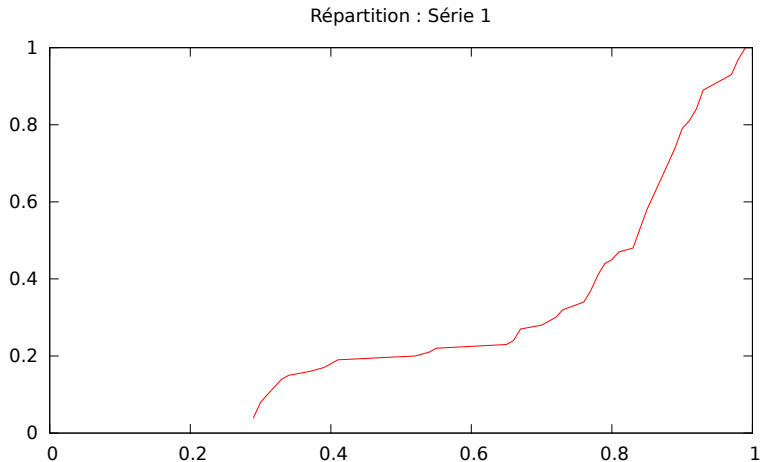
# Évaluation de la confiance

## Convergence

série	moyenne	écart-type	intervalle de confiance
1	0,75	0,22	$0,77 \pm 0,082$
2	0,7	0,27	$0,69 \pm 0,09$
3	0,45	0,19	$0,53 \pm 0,10$
4	0,96	0,02	1
5	0,99	0,01	1
6	0,02	0,03	0
8	0,62	0,09	nd
9	0,6	0,15	nd
a	0,42	0,41	$0,4 \pm 0,09$
b	0,31	0,42	$0,29 \pm 0,09$
c	0,85	0,1	$0,97 \pm 0,03$

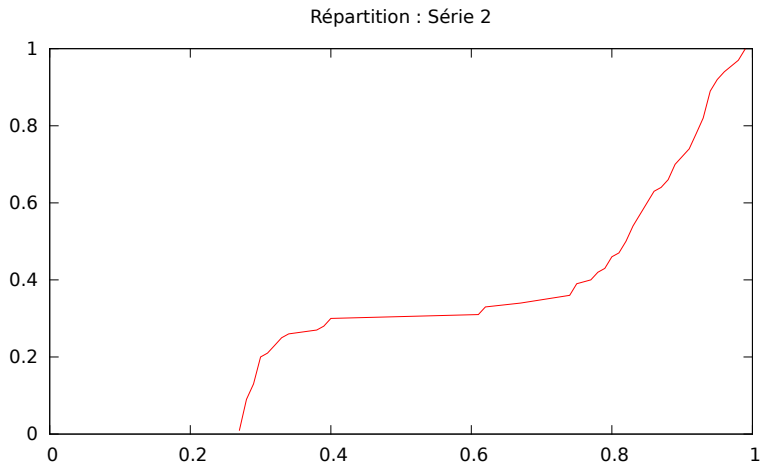
# Fonction de répartition : série 1

1000 sessions, 1000000 sites web, distance max à 0.42, 20 topics.



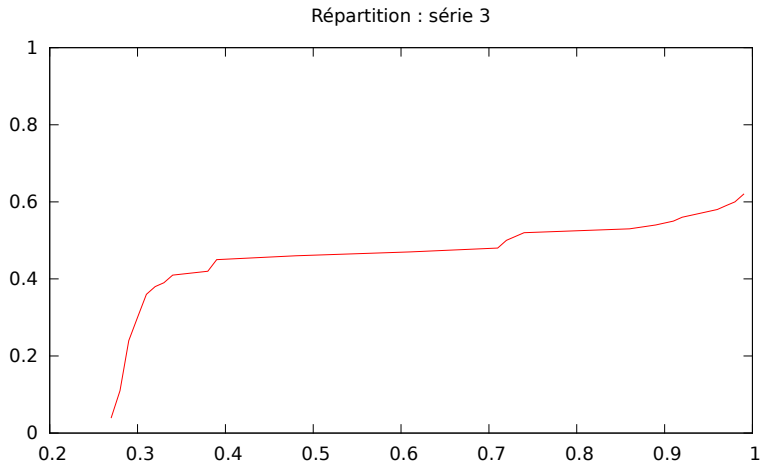
# Fonction de répartition : série 2

1000 sessions, **100000** sites web, distance max à 0.42, 20 topics.



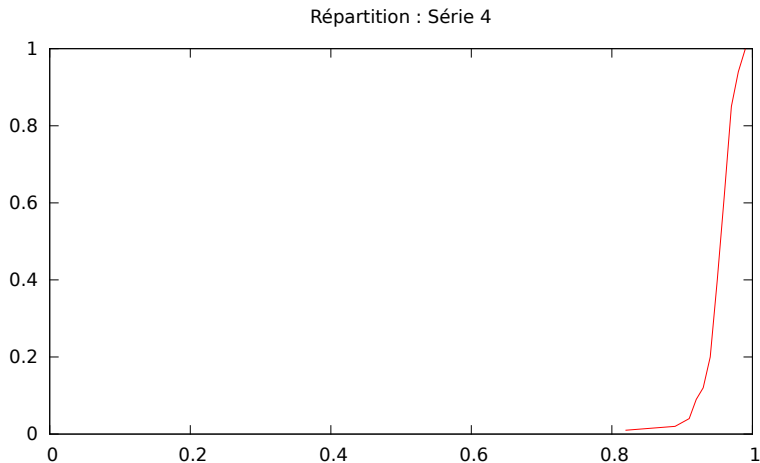
# Fonction de répartition : série 3

1000 sessions, 100000 sites web, distance max à **0.35**, 20 topics.



# Fonction de répartition : série 4

1000 sessions, 100000 sites web, distance max à **0.6**, 20 topics.



# Fonction de répartition : série 5

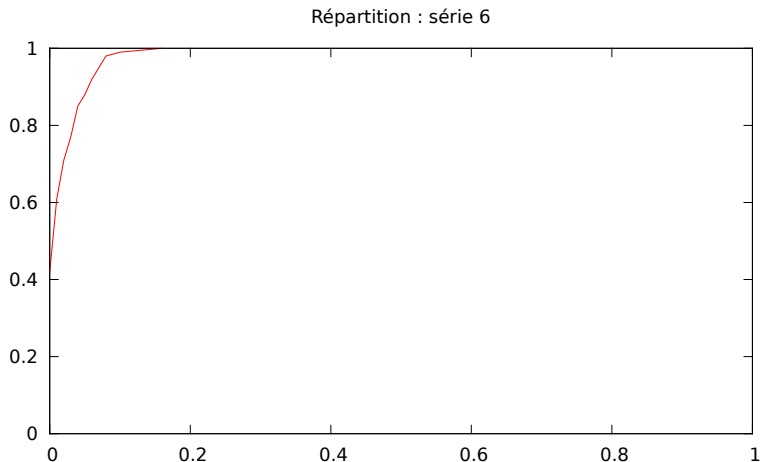
1000 sessions, 100000 sites web, distance max à **1.0**, 20 topics.



# Fonction de répartition : série 6

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

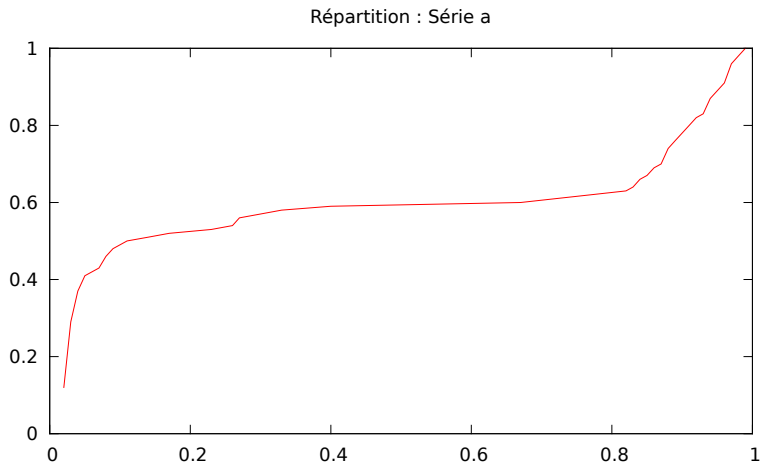
Pas de mise à jour de la stratégie de propositions des sites web.



# Fonction de répartition : série a

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

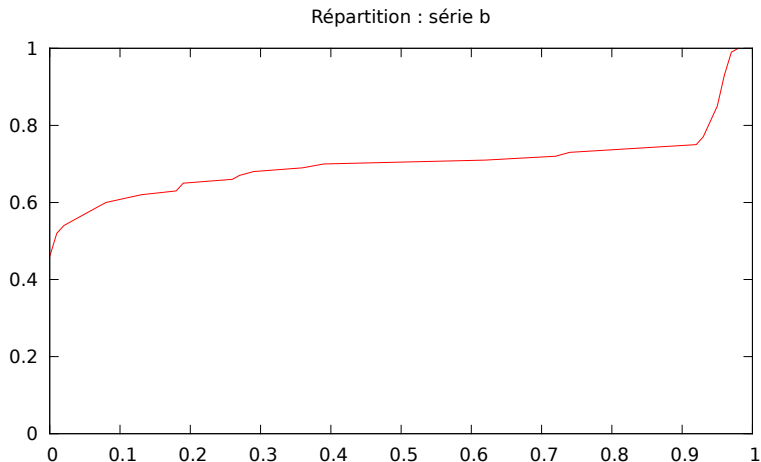
Facteur de gain constant égal à 0.01.



# Fonction de répartition : série b

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

Facteur de gain constant égal à 0.001.

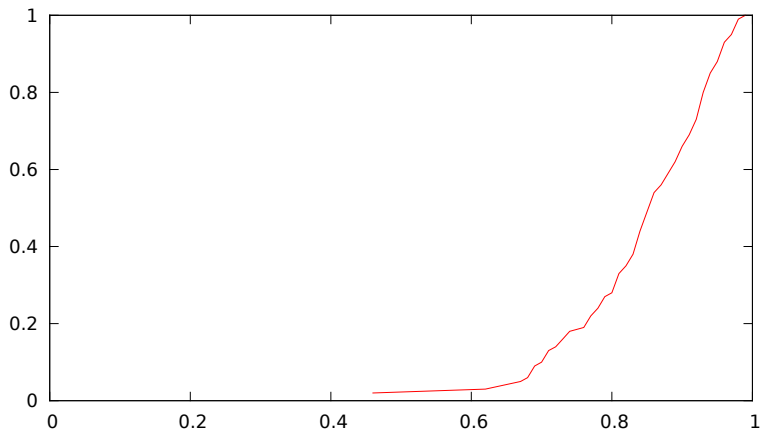


# Fonction de répartition : série c

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

Facteur de gain constant égal à 100.

Répartition : Série c



# Évaluation de la confiance

## Rapidité de la convergence

On ne considère plus que les utilisateurs pour lesquels l'algorithme a réussi. On calcule le temps moyen qu'il faut à l'algorithme pour atteindre la valeur finale de son rendement, à 5% près.

On considère l'évènement  $\{T < \text{temps moyen}\}$ . On va estimer la probabilité qu'a cet évènement. Si on a la probabilité approchée avec nos simulations  $p$ , la proba réelle est avec 95% de chances dans l'intervalle  $[p \pm 1,96 \times \sqrt{(p \times (1 - p)/k)}]$  où  $k$  est le nombre de simulations qui ont réussi.

# Évaluation de la confiance

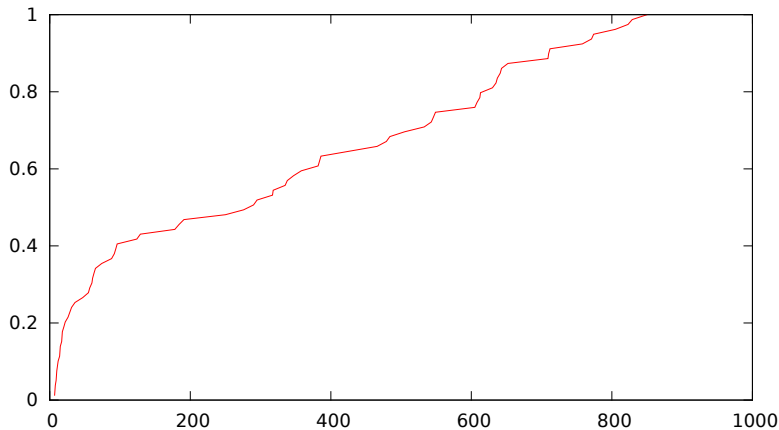
## Rapidité de la convergence

série	temps moyen	écart-type	intervalle de confiance
1	314,2	286,42	$0,25 \pm 0,085$
2	273,67	250,35	$0,39 \pm 0,09$
3	235,18	213,9	$0,43 \pm 0,10$
4	130,28	119,14	$0,26 \pm 0,08$
5	0,4	0,4	$0,98 \pm 0,03$
a	298,97	273,79	$0,12 \pm 0,06$
b	244,45	221,06	$0,13 \pm 0,06$
c	343,07	312,12	$0,25 \pm 0,08$

# Fonction d'efficacité : série 1

1000 sessions, 1000000 sites web, distance max à 0.42, 20 topics.

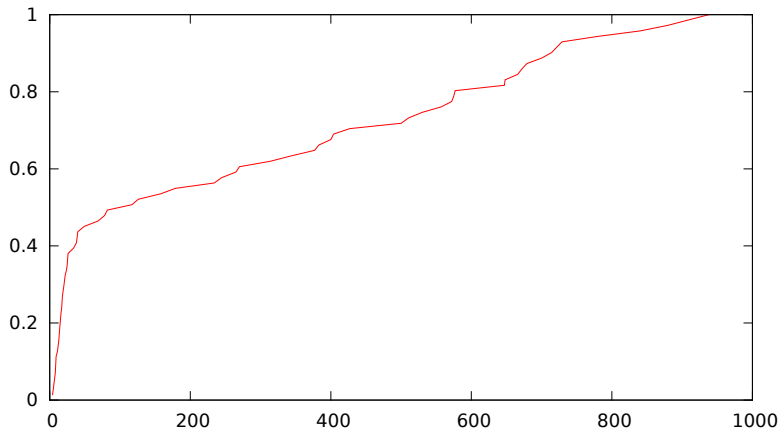
Répartition temps moyen : série 1



# Fonction d'efficacité : série 2

1000 sessions, **100000** sites web, distance max à 0.42, 20 topics.

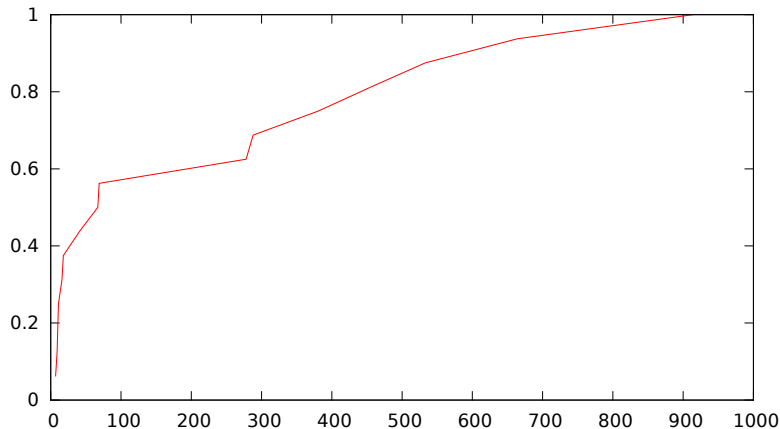
Répartition temps moyen : série 2



# Fonction d'efficacité : série 3

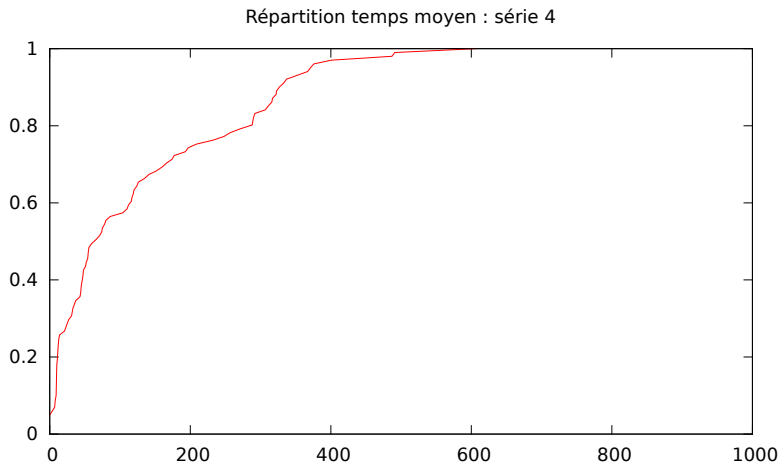
1000 sessions, 100000 sites web, distance max à **0.35**, 20 topics.

Répartition temps moyen : série 3



# Fonction d'efficacité : série 4

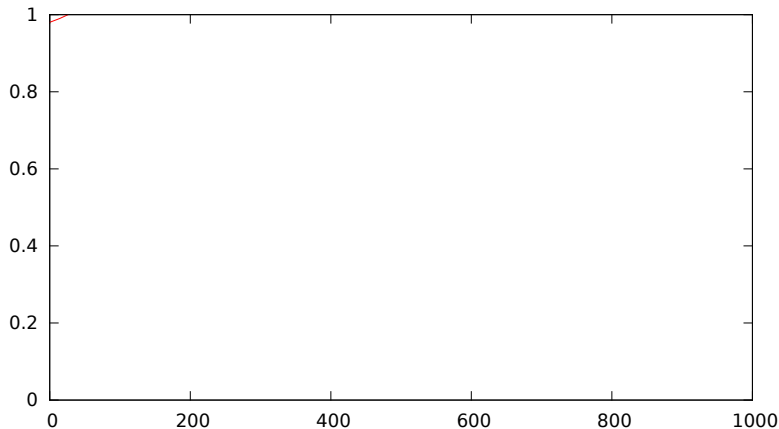
1000 sessions, 100000 sites web, distance max à **0.6**, 20 topics.



# Fonction d'efficacité : série 5

1000 sessions, 100000 sites web, distance max à **1.0**, 20 topics.

Répartition temps moyen : série 5

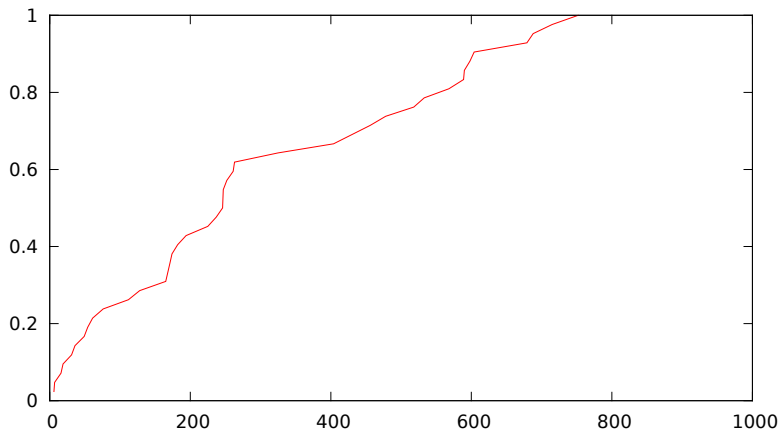


# Fonction d'efficacité : série a

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

Facteur de gain constant égal à 0.01.

Répartition temps moyen : série a

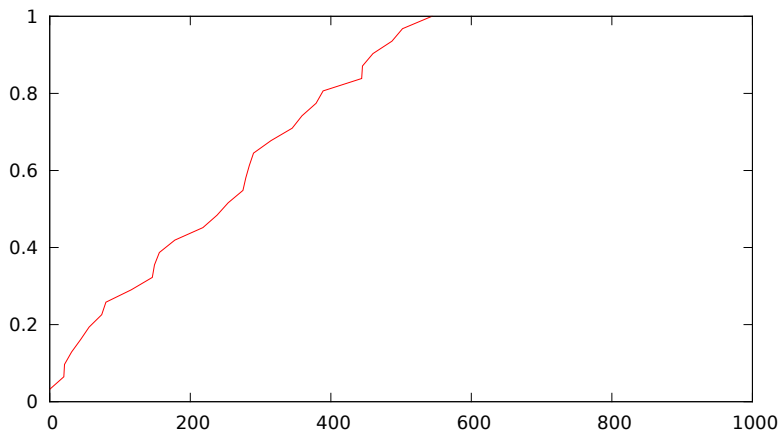


# Fonction d'efficacité : série b

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

Facteur de gain constant égal à 0.001.

Répartition temps moyen : série b

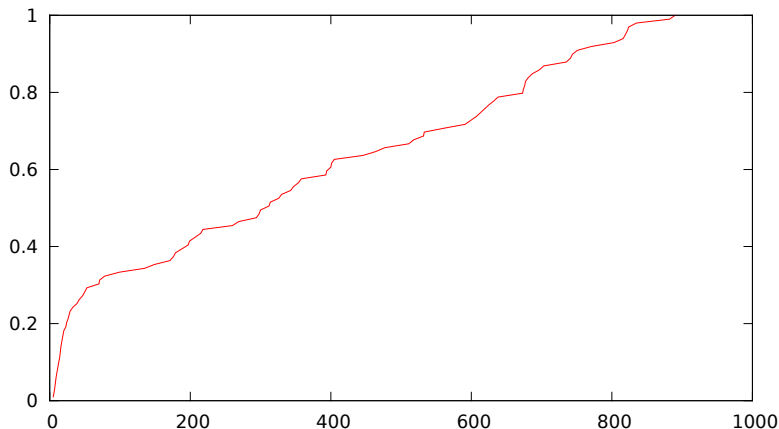


# Fonction d'efficacité : série c

1000 sessions, 100000 sites web, distance max à 0.42, 20 topics.

Facteur de gain constant égal à 100.

Répartition temps moyen : série c



**Questions ?**