INVESTIGATIONS PARTICULAIRES POUR L'INFÉRENCE STATISTIQUE ET L'OPTIMISATION DE PLAN D'EXPÉRIENCES

Éric PARENT [1], Billy AMZAL [2], Philippe GIRARD [3]

RÉSUMÉ

Les algorithmes particulaires sont des techniques de Monte-Carlo qui associent des étapes d'échantillonnage pondéré, de rééchantillonnage bootstrap, de régénérescence markovienne et de recuit simulé. Grâce à trois exemples de complexité croissante, nous décrivons leurs implémentations pour l'estimation du maximum de vraisemblance, l'évaluation de la distribution *a posteriori* pour un modèle à variables latentes et la recherche du plan d'expérience optimal. Les solutions de ces exemples pédagogiques illustrent les performances et les limites de ces algorithmes, promis à une place de choix dans la trousse à outils du statisticien.

Mots-clés : algorithmes particulaires, simulation de Monte-Carlo, plan d'expérience optimal, inférence bayésienne

ABSTRACT

Particle algorithms are Monte Carlo techniques that put together steps of importance sampling, bootstrap resampling, markovian rejuvenating and simulated annealing. We develop three examples of increasing complexity and explain how to implement such algorithms for maximum likelihood search, for inference of a model with latent variables and for optimal design. Since we believe that particle algorithms will soon become tools of choice for statistical practitioners, their results are compared with the known solutions of these rather common examples so as to test the algorithms' performances and to show their limits.

Keywords : Particle algorithms, Monte Carlo simulation, optimal experimental design, Bayesian inference

1. Introduction

Dans le paradigme bayésien (Robert, 2006), la formule de Bayes donne la clé de l'apprentissage statistique quant aux valeurs possibles du vecteur inconnu des paramètres θ d'un modèle paramétrique lorsqu'on dispose de données

1. Équipe Modélisation, Risques, Statistique, Environnement, de l'UMR 518 INRA/AgroParisTech, 19, Avenue du Maine, 75732 Paris Cedex 15, France, eric.parent@agroparistech.fr

2. Novartis Pharma AG, Postfach, CH-4002 Bâle, Suisse.

3. Nestlé Research Center, Vers-chez-les-Blanc, 1000 Lausanne 26, Suisse.

y. Le savoir *a posteriori* sur l'inconnue s'exprime sous la forme d'un pari probabiliste : la distribution *a posteriori* $[\theta\,|y]$ est le produit normalisé de la vraisemblance $[y\,|\theta]$ par la distribution *a priori* $[\theta]$. Hormis les cas de miracles mathématiques que constitue la conjugaison (il faut alors se restreindre aux seuls modèles de la famille exponentielle et choisir des lois *a priori ad hoc*), le calcul de la constante de normalisation $\int [y\,|\theta]\,[\theta]\,d\theta$ décourage les numériciens (et nombre de praticiens) dès que la dimension du modèle excède trois paramètres. Les algorithmes de simulation de Monte-Carlo (échantillonnage pondéré, algorithmes de Monte-Carlo par chaînes de Markov) n'ont pas besoin du calcul de cette constante (Robert et Casella, 2004) : ils remplacent astucieusement le calcul d'une densité de probabilité par la mise à disposition d'un échantillon ayant les mêmes propriétés vis-à-vis de la loi des grands nombres qu'un échantillon tiré selon la distribution $[\theta\,|y]$. Connus dès le milieu du 20ème siècle (Metropolis *et al.*, 1953), mais d'emploi généralisé seulement à partir des années 1990 avec l'avènement de micro-ordinateurs puissants, l'arrivée de la première vague de ces algorithmes (Gilks *et al.*, 1996) a donné lieu à une véritable révolution (Brooks, 2003) : libérés de bon nombre de difficultés de la phase d'inférence (Smith et Gelfand, 1992), les statisticiens bayésiens ont pu beaucoup investir de leur créativité dans l'étape de modélisation (Parent et Bernier, 2007). On peut objecter que cette exaltation *révolutionnaire* n'est pas allée sans déboire : un cadre formel séduisant peut inciter à ne pas prendre garde au risque de *sur-modéliser* et de confondre modèle et réalité, le prix à payer pour l'emploi des méthodes de simulation de Monte-Carlo est de vérifier précautionneusement les conditions de convergence des algorithmes (Mengersen *et al.*, 1999), etc. Mais l'apparition de ces algorithmes a également bénéficié aux approches de l'École Classique. Dans les modèles à effets aléatoires, ces mêmes algorithmes interviennent dans la phase *espérance* des méthodes EM qui nécessitent aussi de simuler des distributions de probabilité dont la constante de normalisation n'est généralement pas connue (Douc *et al.* ; Kuhn et Lavielle, 2004).

Depuis une dizaine d'années se développe une seconde vague de méthodes de simulation de Monte-Carlo : les algorithmes particulaires (Doucet *et al.*, 2001). Le principe général de ces algorithmes particulaires repose sur le fait de ne plus considérer rétrospectivement une chaîne simulée pour en extraire un échantillon, mais de construire l'échantillon globalement (Cappé *et al.*, 2004) et au fur et à mesure. A l'origine, ces algorithmes furent développés pour améliorer le filtrage séquentiel dans les domaines de l'automatique et du traitement du signal (Liu et Chen, 1998 ; Pitt et Shephard, 1999 ; Arulampalam *et al.*, 2002). Un cas d'application typique est le suivi radar où l'on désire suivre une cible pour laquelle des données bruitées de position, de vitesse et d'accélération sont disponibles à chaque instant. Le filtrage séquentiel permet de retrouver la position de la cible avec la précision maximale. Le filtrage particulaire fut alors posé en généralisation efficace du filtrage de Kalman aux cas non linéaires non gaussiens, profitant de la flexibilité des modèles bayésiens et des performances des algorithmes de simulation associés. Ils permettent ainsi d'intégrer en temps réel l'information disponible pour mettre à jour les prédictions. Très vite, ces algorithmes de Monte-Carlo particulaires

furent utilisés dans des cadres plus variés et plus généraux que le filtrage (Künsch, 2001 ; Doucet *et al.*, 2004). En effet, d'une manière plus générale, si l'on se donne une suite de densités de probabilité $p_1(\theta_1), p_2(\theta_2), ...$ quelconques, les méthodes de Monte-Carlo séquentielles permettent de faire évoluer un échantillon initial de manière à ce qu'il puisse être considéré comme tiré successivement de chacune des lois $p_1(\theta_1), p_2(\theta_2), ...$ Ainsi, ces algorithmes particulaires ont pu être utilisés pour toute estimation bayésienne, séquentielle ou non, de modèles statistiques (Chopin, 2002), de simulation de lois complexes difficiles à simuler que l'on approche par une suite de lois plus simples (Doucet *et al.*, 2001), ou encore pour des optimisations basées sur recuit simulé (Amzal *et al.*, 2006). En parallèle aux applications, le cadre d'étude théorique s'est développé (Del Moral, 2004, Del Moral *et al.*, 2001, Oujdane et Ruberthaler, 2005) et les théorèmes de convergence du type « limite centrale » démontrés (Künsch, 2005, Chopin, 2004, Del Moral et Guionnet, 1999).

Quel sera le devenir de ce type d'algorithme pour la communauté statistique ? Cet article présente une initiation à ces techniques et une réflexion prospective des utilisations, bayésiennes ou non, que pourront en faire les statisticiens.

Dans la partie 2, nous décrivons d'abord comment construire un algorithme particulaire et illustrons sa performance sur l'inférence bayésienne du modèle Normal. La partie 3 complète cet algorithme par une étape de recuit simulé, ce qui donne un moyen numérique d'atteindre le maximum de vraisemblance. Dans la partie 4, nous montrons comment mettre en œuvre cet algorithme sur un modèle plus complexe, de type binomial à variables latentes. Enfin la partie 5 décrit comment cet algorithme fournit une nouvelle piste de recherche des plans d'expériences optimaux (au sens de la théorie de l'utilité espérée). La conclusion discute des avantages et des limites de ces algorithmes.

2. Associer les méthodes de simulation de Monte-Carlo pour faire évoluer un essaim de particules

2.1. Un exemple de base dont la solution est connue

Considérons à titre d'exemple le modèle Normal. Le vecteur des paramètres dont on veut réaliser l'inférence sera ici formé de la moyenne et de la précision (inverse de la variance) : $\theta = (\mu, \sigma^{-2})$. La vraisemblance issue de l'observation d'un $n-$échantillon $y = (y_1, y_2, ...y_n)$ s'écrit :

$$\begin{aligned}[y\,|\theta] &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2 \\ &= \frac{(2\pi)^{-\frac{n}{2}}}{\sigma} \exp\left(-\frac{1}{2}\frac{n}{\sigma^2}\left(\bar{y}-\mu\right)^2\right)\left(\frac{1}{\sigma^2}\right)^{\frac{n+1}{2}-1} \exp -\frac{1}{2}\frac{S^2}{\sigma^2} \quad . \quad (1)\end{aligned}$$

On prendra par la suite les statistiques exhaustives $\bar{y} = n^{-1}\sum_{i=1}^{n} y_i$ et $S^2 = \sum_{i=1}^{n}(y_i - \bar{y})^2$. Pour traiter cet exemple élémentaire, on va ici choisir la loi *a priori* particulière impropre $[\theta] \propto 1$ de telle sorte que le max de

vraisemblance soit le mode de la loi *a posteriori*. Cette forme plate pour $[\theta]$ peut être obtenue comme limite d'une distribution *a priori* conjuguée. Pour ce cas d'école décrit dans tous les manuels, la constante de normalisation pour passer de $[y\,|\theta]\,[\theta]$ à $[\theta\,|y]$ se calcule aisément, en effet :

$$[\theta\,|y] = \left(\frac{\left(\frac{S^2}{2}\right)^{\frac{n+1}{2}} \sqrt{n}}{\Gamma(\frac{n+1}{2})\sqrt{2\pi}} (2\pi)^{\frac{n}{2}} \right) \times \frac{(2\pi)^{-\frac{n}{2}}}{\sigma} \exp\left(-\frac{1}{2}\frac{n}{\sigma^2}(\bar{y}-\mu)^2\right) \left(\frac{1}{\sigma^2}\right)^{\frac{n+1}{2}-1} \exp -\frac{1}{2}\frac{S^2}{\sigma^2} \quad . \quad (2)$$

Et obtenir des tirages selon la loi jointe *a posteriori* n'est pas difficile :

1. On tire σ^{-2} selon la loi $Gamma(\frac{n+1}{2}, \frac{S^2}{2})$,
2. On tire μ sachant σ^{-2} selon une loi Normale $\mathrm{N}(\bar{y}, \frac{\sigma^2}{n})$.

2.2. Algorithme particulaire de simulation

Dans cette partie, nous mettons en œuvre un algorithme dit *particulaire* pour construire un échantillon de la distribution *a posteriori* de $\theta = (\mu, \sigma^{-2})$. À des fins d'illustration, nous avons pris $n = 10$ avec les observations suivantes :

$$y = (0.26, 1.53, 2.07, 3.55, 1.19, 1.27, 2.83, 2.09, 3.2, 2.01).$$

Les statistiques exhaustives valent $\bar{y} = 2$ et $S^2 = 8.982 = 10 \times (1.11334)^{-1}$. On notera : $f(\theta) = [\theta\,|y] \propto \prod_{i=1}^{n} [y_i\,|\theta]$ et on cherche à obtenir un échantillon de f sans faire appel à son expression analytique particulière (2).

Comme son nom l'indique, l'algorithme particulaire fait évoluer des *particules*, qui sont simplement ici des couples (μ, σ^{-2}). On les désigne par $\theta^{(g)}$ par la suite, avec l'indice g allant de 1 à G. Dans cet exemple on a pris $G = 10000$ particules. L'algorithme comprend trois étapes :

1. Échantillonnage pondéré, dit encore *échantillonnage préférentiel* (ou *importance sampling*), avec une loi d'importance auxiliaire $f_0(\theta)$: cette étape génère G particules selon la distribution auxiliaire $f_0(\theta)$ de simulation commode. Le choix de f_0 peut en théorie être quelconque pourvu que son support contienne celui de f. En pratique, il est souvent crucial pour l'efficacité de l'algorithme. La figure 1 montre le semis de 10000 particules indépendantes généré initialement avec μ en abscisses et σ^{-2} en ordonnées ; on a choisi ici $f_0(\theta)$ particulièrement simple à simuler : on a utilisé une loi produit à composantes indépendantes, Normale $N(2, 2^2)$ pour μ et $Gamma(1.375, 1.12275)$ pour σ^{-2}. Comme le rappelle l'annexe

A, le résultat de l'échantillonnage pondéré est d'associer à chaque particule un poids $w^{(g)} = \left(\frac{f(\theta^{(g)})}{f_0(\theta^{(g)})}\right) \Big/ \left(\sum_{g=1}^{G} \frac{f(\theta^{(g)})}{f_0(\theta^{(g)})}\right)$ de telle sorte que $\left(\theta^{(g)}, w^{(g)}\right)_{g=1,\ldots,G}$ puisse être considéré comme un échantillon pondéré de $f(\theta)$.

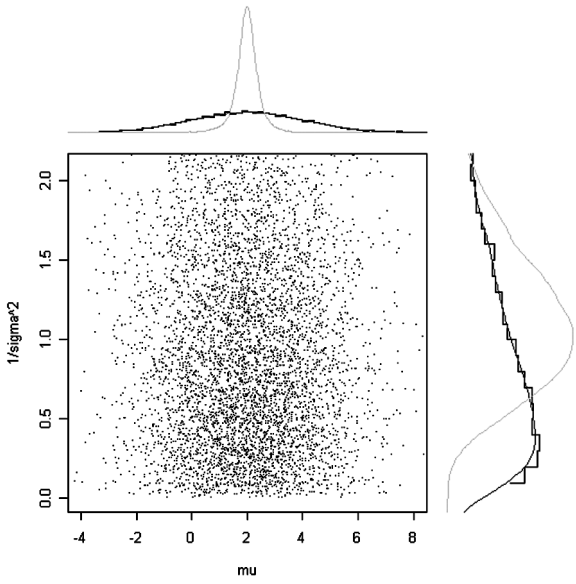

FIG 1. — Dispersion initiale de 10000 particules sur le modèle Normal, selon la densité produit $N(2, 2^2)$ pour μ et $Gamma(1.375, 1.12275)$ pour σ^{-2}.

2. Bootstrap multinomial : quoique toutes les caractéristiques statistiques de la distribution f puissent être obtenues avec l'échantillon pondéré $\left(\theta^{(g)}, w^{(g)}\right)_{g=1,\ldots,G}$, un échantillon ordinaire (équipondéré) de f est obtenu par un tirage de type bootstrap : on rééchantillonne toutes les particules précédemment obtenues selon leur poids (avec remise). On effectue donc un tirage multinomial dans G classes offrant chacune la valeur $\theta^{(g)}$ avec la probabilité $w^{(g)}$. On génère ainsi G' particules $\left(\theta'^{(g)}\right)_{g=1,\ldots,G'}$ ayant (asymptotiquement) les propriétés d'un échantillon de $f(\theta)$ (Rubin, 1988). On a pris ici $G' = G$ comme il est d'usage. La figure 2 montre le résultat de ce bootstrap multinomial à la même échelle que la figure 1 ; les caractéristiques statistiques de f sont bien reconstruites, les particules couvrent plus raisonnablement le champ de variation de f. Sur les marges de la figure 2 sont représentées les lois marginales *vraies* de f et une densité estimée par méthode de lissage à noyau qui semble suggérer une possible bimodalité pour la marginale de σ^{-2}. Sur l'histogramme de cette marginale, on voit également apparaître de grandes fluctuations de l'effectif des classes. Ces

instabilités témoignent du problème des *doublons* : lors du tirage bootstrap, les valeurs associées aux poids forts ont tendance à être répliquées tandis que celles associées à des faibles poids s'éteignent. Notons qu'en pratique, des variantes au bootstrap multinomial sont souvent préférables (comme les ré-échantillonnages systématique, stratifié ou résiduel) car ils réduisent la complexité des calculs informatiques et peuvent dans certains cas améliorer l'approximation des lois (Douc et Cappé, 2005).

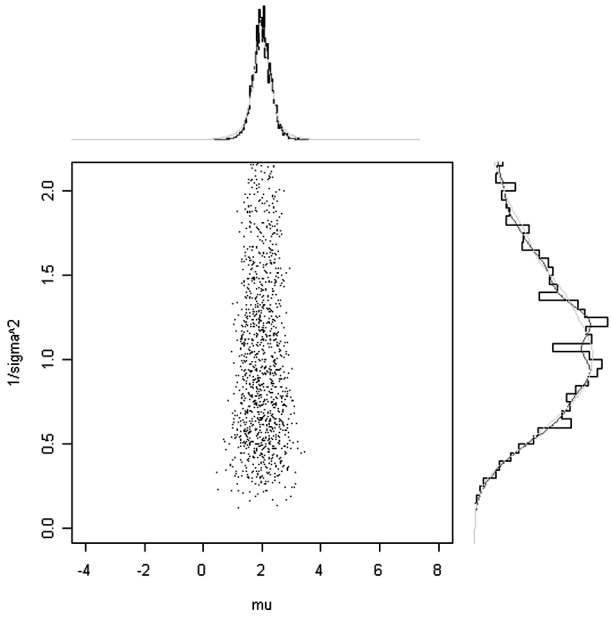

FIG 2. — Échantillonnage pondéré suivi d'un rééchantillonnage bootstrap binomial (10000 particules sur le modèle Normal).

3. Dispersion markovienne des éventuels doublons : pour lutter contre cette dégénérescence de l'échantillon, on redisperse les particules au moyen d'un noyau de transition markovien, ayant la propriété d'avoir f comme loi invariante. On obtient ainsi G particules images $\left(\theta^{''(g)}\right)_{g=1,\ldots,G}$, conservant la propriété d'être toujours issues de la loi f. En pratique, les algorithmes Metropolis-Hastings et de Gibbs sont les plus utilisés pour construire un tel noyau de transition (voir annexe B). Dans l'exemple, nous utilisons un noyau de Gibbs, qui facilite cette dispersion en donnant presque sûrement des images différentes $\theta^{''(g_1)} \neq \theta^{''(g_2)}$ à des valeurs initiales identiques $\theta^{'(g_1)} = \theta^{'(g_2)}$, et surtout facile à construire ici : la distribution de σ^{-2} connaissant y et μ est une loi *Gamma* tandis que la conditionnelle complète de μ est une loi Normale. La figure 3 illustre l'effet bénéfique de cette dispersion : les lois marginales sont reconstruites sans problème, et la loi

conjointe très bien évaluée. La figure 3 est à comparer à la figure 2 où les doublons, superposés, donnent une illusion de densité moindre : sur la figure 2, nous n'avons pas utilisé l'astuce graphique commune qui consiste à introduire de petites perturbations sur la position des points (« jittering » en anglais).

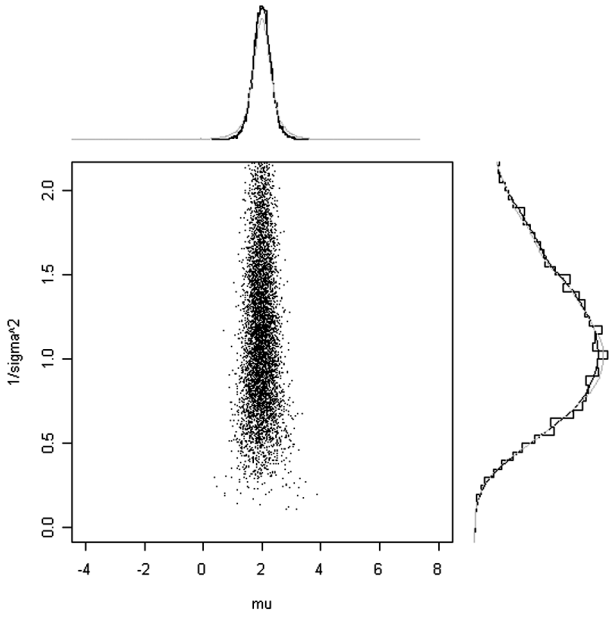

FIG 3. — Dispersion markovienne grâce à une étape de l'échantillonneur de Gibbs (10000 particules sur le modèle Normal).

3. Apprentissage séquentiel

3.1. Un processeur d'information

Remarquons qu'itérer les blocs de trois étapes de l'algorithme décrit précédemment permet l'apprentissage séquentiel. Oublions les statistiques exhaustives du modèle Normal et supposons que nous scindions les données en m blocs indépendants (sachant θ), $y = (y_1, y_2, ..y_m)$. Appelons $f_1(\theta) = [\theta\,|y_1\,]$, $f_2(\theta) = [\theta\,|y_1, y_2\,] \propto f_1(\theta) \times [y_2\,|\theta\,], ...f_m(\theta) = [\theta\,|y_1, y_2, ..y_m\,] \propto f_{m-1}(\theta) \times [y_m\,|\theta\,]$. On peut itérer m fois les 3 phases de l'algorithme. En effet, les particules issues de l'étape 3 de l'itération 1 forment un échantillon approximativement distribué selon la loi f_1. Si on les réutilise en entrée de la première étape de l'itération 2, il faut simplement leur associer un poids $w(\theta)$ proportionnel à $\frac{f_2(\theta)}{f_1(\theta)} = [y_2\,|\theta\,]$ pour les considérer comme un échantillon pondéré de f_2, puis

enchaîner les deux étapes suivantes pour sortir un ensemble équidistribué de particules approximativement distribuées selon loi f_2. En itérant, on passe successivement de f_1 à f_2, puis de f_2 à f_3, etc. jusqu'à f_m. Typiquement, on rencontre cette situation lorsque les données arrivent par paquets, et que leur analyse est requise entre chaque paquet. Cette possibilité d'*apprentissage séquentiel* procure un avantage notable sur les méthodes MCMC qui ne permettent pas de réutiliser les simulations déjà effectuées pour réajuster la distribution *a posteriori* quand s'accumulent les données. On comprend l'intérêt de tels algorithmes pour les modèles sans résumés exhaustifs, les applications « temps réel » comme le suivi radar ou la digestion de fichiers de données trop volumineux qu'il faut saucissonner (Chopin, 2002) ! De plus, la récursivité de ces 3 phases peut permettre d'adapter les lois d'importance au fil des itérations de manière efficace, sous certaines conditions (voir Douc *et al.*, 2007).

3.2. Recuit simulé pour la maximisation de vraisemblance

Tout enseignant a déjà dissuadé des étudiants trop enthousiastes de réutiliser comme loi *a priori* de la formule de Bayes la distribution *a posteriori* qu'ils venaient de calculer sur le même jeu de données. Que se passe-t-il si on effectue néanmoins cette opération illicite T fois ? Formellement, c'est comme si l'on analysait un jeu de données abusivement *iid* :

$$Y_T = (\underbrace{y, y, ...y}_{T \text{ fois}}).$$

Pour ce jeu répété, la densité *a posteriori* est $f_T(\theta) = \dfrac{[\theta\,|y]^T\,[\theta]}{\displaystyle\int [\theta\,|y]^T\,[\theta] d\theta} \propto [y\,|\theta]^T$ (car on a pris $[\theta] = 1$). Pour l'exemple Normal, on trouve :

$$f_T(\theta) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{nT}{2}} \exp\left(-\frac{1}{2}\frac{Tn}{\sigma^2}\left(\bar{y}-\mu\right)^2\right) \exp -\frac{1}{2}\frac{TS^2}{\sigma^2} \quad . \tag{3}$$

On constate que, quand $T \to \infty$, la loi $f_T(\theta)$ tend vers une distribution de Dirac au point de maximum de vraisemblance ($\hat{\mu} = \bar{y}$ et $\hat{\sigma}^{-2} = \left(\frac{n}{S^2}\right)$) : la vraisemblance subit un effet dit de *recuit simulé* (Van Laarhoven et Aarts, 1987). Ce comportement général n'est pas lié à l'exemple normal choisi associé à une loi *a priori* plate en μ : d'après le théorème central limite, la convergence en loi a lieu à la vitesse $\sqrt{T}$ pour toute vraisemblance (sous réserve qu'elle soit associée à une loi *a priori* n'excluant aucune valeur possible du paramètre) quand on considère, abusivement comme les étudiants ci-dessus, que les T pseudo-réplications identiques de l'échantillon forment nT observations indépendantes.

La figure 4 montre ce comportement de regroupement vers le maximum de vraisemblance ($\hat{\mu} = \bar{y}$ et $\hat{\sigma}^2 = \frac{S^2}{n}$) avec les données de la section précédente.

On a pris 1000 particules et $T = 36$, car si l'objectif est trouver le mode, on peut se satisfaire de moins de particules que pour reconstruire toute la distribution *a posteriori*.

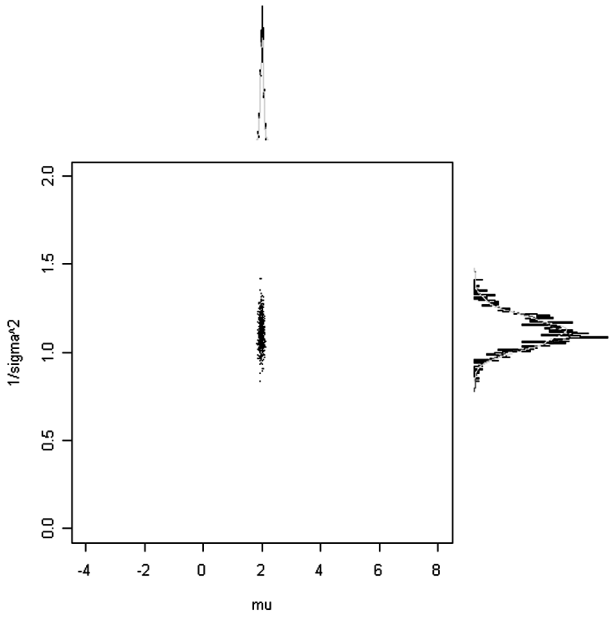

FIG 4. — Recuit simulé sur le modèle Normal (1000 particules avec 36 itérations).

Enfin, 100 particules itérées 10000 fois à travers l'algorithme précédent permettent d'évaluer très précisément $\hat{\mu}$ et $\hat{\sigma}^2$. À titre de comparaison pour le tableau 1, les valeurs exactes sont 2 et 1.113338.

TABLEAU 1. — Répartition des 100 particules après 10000 itérations.

quantile	25%	50%	75%
μ	1.9980	2.00001	2.0020
σ^{-2}	1.10996	1.11329	1.11673

Cela illustre que, au moins formellement, ces méthodes peuvent être employées pour évaluer le maximum de vraisemblance pour un modèle plus général que le modèle Normal (voir par exemple Brooks et Morgan, 1995 ou Andrieu et Doucet, 2000) : posant une loi *a priori* uniforme $[\theta] = 1$, le mode de la loi *a posteriori* $[\theta\,|y]$ est alors évalué à partir d'un échantillon de cette loi. Berger utilise d'ailleurs cette analogie pour établir une approximation asymptotique Normale de la loi *a posteriori*, centrée sur l'estimateur du maximum de vraisemblance dans le cas de modèles avec observations *iid* (Berger, 1985).

4. Prise en compte de variables latentes

4.1 Un exemple classique

Utiliser un algorithme particulaire pour conduire l'inférence d'un modèle Normal, c'est comme prendre un marteau pour écraser une mouche. Cet outil est bien sûr promis à de meilleures utilisations. Dans la suite, nous prenons comme second exemple le cas plus difficile décrit à la section 9.3.3 de Mc Cullagh et Nelder, 1989. Deux variables aléatoires binomiales $X_1 \sim Bin(n_1, \theta_1)$ et $X_2 \sim Bin(n_2, \theta_2)$ ne sont observées qu'au travers de leur somme $Y = X_1 + X_2$. Mc Cullagh et Nelder considèrent trois répétitions de cette expérience, mesurées dans le tableau 2.

TABLEAU 2. — Observations et variables latentes pour l'exemple de Mc Cullagh et Nelder.

Répétitions	$i = 1$	$i = 2$	$i = 3$
$n_{i,1}$	5	6	4
$n_{i,2}$	5	4	6
y_i	7	5	6

On peut écrire la vraisemblance $[y_1, y_2, y_3 \,|\theta_1, \theta_2\,]$ sous la forme :

$$\prod_{i=1}^{3} \sum_{j_i} \binom{n_{i,1}}{j_i} \binom{n_{i,2}}{y_j - j_i} (\theta_1)^{j_i} (1-\theta_1)^{n_{i,1}-j_i} (\theta_2)^{y_i-j_i} (1-\theta_2)^{n_{i,2}-y_i+j_i} \quad (4)$$

avec $\max(0, y_i - n_{i,2}) \leqslant j_i \leqslant \min(n_{i,1}, y_i)$. Cette surface est représentée à la figure 5.

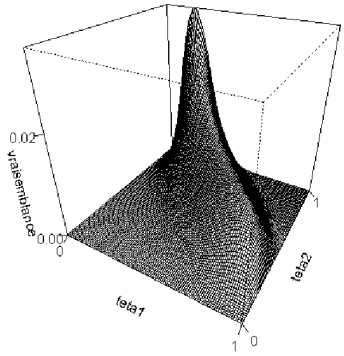

FIG 5. — Surface de la vraisemblance pour le modèle de sommes binomiales avec les données de Mc Cullagh et Nelder.

Dans l'approche bayésienne, on peut écrire la loi *a posteriori* des paramètres. Pour cela, il est plus commode et plus naturel de faire apparaître les variables cachées X_1 et X_2 et d'introduire séquentiellement les données. Appelons H l'historique de l'information cumulée sur le système à un instant donné, on notera par exemple $H_0 = \emptyset$ pour la distribution *a priori*, puis successivement $H_1 = \{y_1\}$, $H_2 = \{y_1, y_2\}$, $H_3 = \{y_1, y_2, y_3\}$ etc. Supposons que l'on soit dans un état de connaissance H et qu'arrive une nouvelle donnée $Y = y$, alors :

$$\begin{aligned}
[\theta_1, \theta_2 \,|y, H\,] &\propto [y, \theta_1, \theta_2 \,|H\,] \\
&\propto [\theta_1, \theta_2 \,|H\,] \times \sum_{x_1, x_2} [y, x_1, x_2 \,|\theta_1, \theta_2, H\,] \\
&\propto [\theta_1, \theta_2 \,|H\,] \\
&\times \sum_{x_1, x_2} 1_{x_1+x_2=y} \binom{n_1}{x_1} \theta_1^{x_1} (1-\theta_1)^{n_1-x_1} \binom{n_2}{x_2} \theta_2^{x_2} (1-\theta_2)^{n_2-x_2}.
\end{aligned}$$

Dans la suite, nous supposerons de plus que $[\theta_1, \theta_2 \,|H_0\,] = 1$, afin de rendre proportionnelles vraisemblance et distribution *a posteriori*. Ce choix correspond à une loi *a priori* vague obtenue en pariant de façon indépendante *a priori* sur des valeurs de θ_1 et θ_2 selon une loi *beta*(1,1), dite aussi loi uniforme sur $(0, 1)$. Ainsi, formellement le problème de recherche du mode de la loi *a posteriori* devient ici identique à celui de la maximisation de vraisemblance. Pour cet exemple, on pourrait certes mettre en place un algorithme EM (Tanner, 1992) afin d'exécuter ce travail (avec moins d'effort), à la présence possible d'extrema locaux près. Cependant, compte-tenu de l'objectif de cet article essentiellement pédagogique d'illustrer l'emploi des méthodes particulaires, nous empruntons la piste bayésienne.

4.2. Apprentissage particulaire séquentiel

Nous proposons de traiter le problème d'inférence introduit en utilisant un algorithme de type particulaire. Cette section décrit l'algorithme d'exploration de la loi *a posteriori* en utilisant des simulations d'ensembles de particules, dans un premier temps sans le recuit simulé.

On appelle particule de la séquence i, un $(2+i) - uplet$ $\Psi = (\theta_1, \theta_2, z_1, ..z_i)$. Chaque séquence de trois étapes de l'algorithme particulaire *assimilera* un y_i, et les particules grossiront d'un vecteur latent $z_i = x_{i,1}$. On note :

1. $f_0\,(\theta_1, \theta_2) = [\theta_1, \theta_2]$,
2. $f_1\,(\theta_1, \theta_2, z_1) = [\theta_1, \theta_2, X_{1,1} = z_1, X_{1,2} = y_1 - z_1 \,|y_1\,]$
3. $f_2(\theta_1, \theta_2, z_1, z_2)$
   $= [\theta_1, \theta_2, X_{1,1} = z_1, X_{1,2} = y_1 - z_1, X_{2,1} = z_2, X_{2,2} = y_2 - z_2 \,|y_1, y_2\,]$
4. $f_3((\theta_1, \theta_2, z_1, z_2, z_3)$
   $= [\theta_1, \theta_2, X_{1,1} = z_1, X_{1,2} = y_1 - z_1, X_{2,1} = z_2, X_{2,2} = y_2 - z_2, X_{3,1} = z_3,$
   $X_{3,2} = y_3 - z_3 \,|y_1, y_2, y_3\,]$.

Une séquence de l'algorithme se déroule de la façon suivante, pour $i = 1, 2,$ ou 3 :

1. *Échantillonnage pondéré.* Au départ, on suppose disposer d'un échantillon de G particules de la distribution f_{i-1}. En particulier, les couples $\left(\theta_1^{(g)}, \theta_2^{(g)}\right)$ sont approximativement distribués selon une loi $[\theta_1, \theta_2 \,|H_{i-1}\,]$. On tire $Z_i = z_i$ selon une loi binomiale $Bin(n_1, \theta_1)$ tronquée entre $\max(0, y_i - n_2)$ et $\min(n_1, y_i)$, que l'on juxtapose à la particule correspondante issue de f_{i-1}. Le poids associé à la particule $\Psi^{(g)} = \left(\theta_1^{(g)}, \theta_2^{(g)}, ..z_i^{(g)}\right)$ est :
$$w(\Psi^{(g)}) = \begin{pmatrix} n_2 \\ z_i^{(g)} \end{pmatrix} \left(\theta_2^{(g)}\right)^{y - z_i^{(g)}} \left(1 - \theta_2^{(g)}\right)^{n_2 - y + z_i^{(g)}} \quad .$$

2. *Rééchantillonnage.* On effectue un rééchantillonnage de type bootstrap binomial avec les poids précédents. On se retrouve alors avec un jeu de particules approximativement distribuées selon la loi $f_i(\theta_1, \theta_2, z_1, ..z_i)$ mais comprenant de possibles doublons dus à d'éventuels déséquilibres entre les pondérations lors du rééchantillonnage.
3. *Dispersion markovienne.* Pour la dispersion markovienne, on va profiter de la conjugaison conditionnelle pour construire un noyau de Gibbs. À $z_1, ..z_i$ fixés, les lois conditionnelles complètes de θ_1 et θ_2 sont respectivement des lois *beta* indépendantes de paramètres $\left(\left(\sum_{j=1}^{i} z_j\right), \left(\sum_{j=1}^{i} (n_{j1} - z_j)\right)\right)$ et $\left(\left(\sum_{j=1}^{i} y_j - z_j\right), \left(\sum_{j=1}^{i} (n_{j1} + z_j - y_j)\right)\right)$. À θ_1 et θ_2 fixés, les lois conditionnelles complètes des $z_1, ..z_i$ sont indépendantes : il s'agit de distributions très facilement simulables, car discrètes entre $\max(0, y_i - n_{i,2})$ et $\min(n_{i,1}, y_i)$ avec pour loi de probabilité
$$[Z_j = z \,|\theta_1, \theta_2, Y_j = y\,] = \frac{\begin{pmatrix} n_1 \\ z \end{pmatrix} \begin{pmatrix} n_2 \\ y - z \end{pmatrix} \left(\frac{\theta_1}{1 - \theta_1}\right)^z \left(\frac{1 - \theta_2}{\theta_2}\right)^z}{\sum_{\max(0, y - n_2)}^{\min(n_1, y)} \begin{pmatrix} n_1 \\ z \end{pmatrix} \begin{pmatrix} n_2 \\ y - z \end{pmatrix} \left(\frac{\theta_1}{1 - \theta_1}\right)^z \left(\frac{1 - \theta_2}{\theta_2}\right)^z} \quad .$$

   Notons qu'il est ici préférable d'échantillonner d'abord les z_j, puis les (θ_1, θ_2) pour améliorer la dispersion sur l'espace des (θ_1, θ_2) qui nous intéresse. Par ailleurs, il n'est plus nécessaire de garder les z_j en mémoire à la fin de cette étape.

La figure 6 montre le nuage en (θ_1, θ_2) après trois passages par les trois étapes de cet algorithme, afin d'assimiler le jeu de données de Mc Cullagh et Nelder (c'est-à-dire la densité *a posteriori* $[\theta_1, \theta_2 \,|H_3\,]$, ici proportionnelle à la vraisemblance).

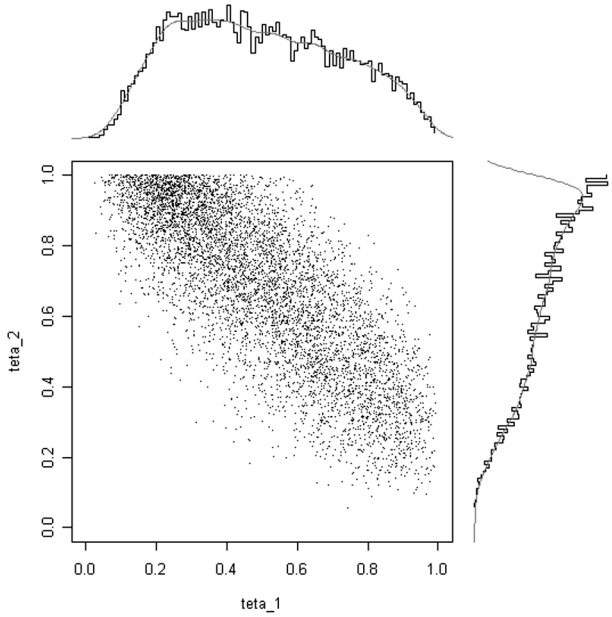

FIG 6. — Simulation particulaire selon la loi *a posteriori* du modèle de somme de deux binomiales avec les données de Mc Cullagh et Nelder.

On retrouve les traits caractéristiques de l'inférence de taux de succès de binomiales observées par leur somme, notamment la forte dispersion des marginales (seulement trois données rendent la situation peu informative), une corrélation négative entre θ_1 et θ_2 qui va de pair avec une légère bimodalité (une compensation possible entre les rôles respectifs des paramètres). Le tableau 3 montre pour les premières statistiques caractéristiques de la loi conjointe *a posteriori* $[\theta_1, \theta_2 \,|H_3\,]$, l'excellente correspondance entre les résultats de l'algorithme et le calcul direct sur l'équation 4 (par discrétisation du carré unitaire en un million de tuiles).

TABLEAU 3. — Comparaison entre valeurs théoriques et valeurs estimées des moments de la loi *a posteriori* avec $G = 10000$ particules.

Moment	valeur théorique	valeur estimée
$\mathbb{E}(\theta_1)$	0,5014	0,5030
$\mathbb{E}(\theta_2)$	0,6751	0,6732
$\mathbb{V}ar(\theta_1)$	0,0502	0,0506
$\mathbb{V}ar(\theta_2)$	0,0502	0,0487
corrélation	-0,7885	-0,7834

Comme dans la première partie, et à la manière de Doucet *et al.*, 2002, on peut itérer l'algorithme pour rechercher le mode de la loi a *posteriori* (c'est-à-dire, avec une distribution a *priori* plate, le maximum de vraisemblance), au prix de l'accroissement linéaire de la taille des particules à chaque itération. Les résultats du recuit simulé particulaire ainsi effectué sont reportés dans le tableau 4.

TABLEAU 4. — Comparaison entre valeurs théoriques et valeurs estimées du mode bivarié de la loi a *posteriori* avec $G = 100$ particules et $T = 50$, avec son intervalle de confiance à 90% .

	valeur théorique	Q-5%	valeur estimée	Q-95%
Mode coord1	0,2	0,166	0,202	0,230
Mode coord2	0,999	0,98037	0,9959	0,99984

Remarquons que, dans les données du tableau 2, si on attribue à y_3 la valeur 5 (au lieu de 6 dans les données originelles), on symétrise le problème, et donc sa solution. Un algorithme d'optimisation brutal pour la recherche du maximum de vraisemblance ferait fi de cette situation particulière. Au contraire, le résultat quasi symétrique de la figure 7 donnant $[\theta_1, \theta_2 \,|H_3]^T$ avec $T = 6$ détecte bien ce comportement typique symétrique.

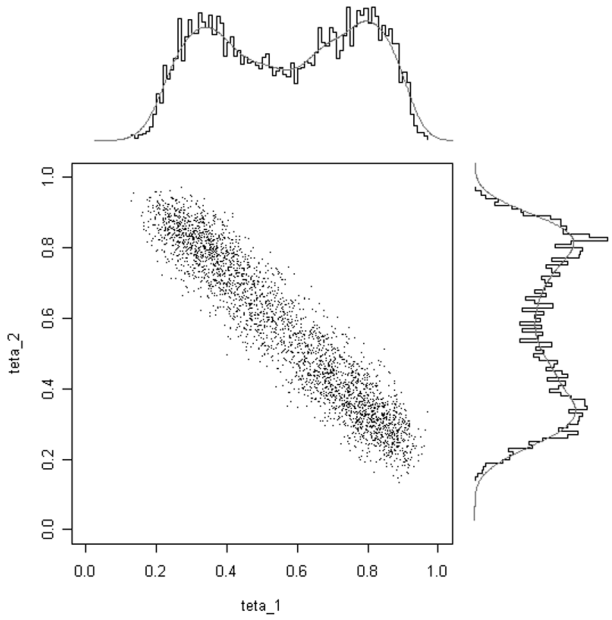

FIG 7. — Recherche particulaire avec recuit simulé ($T = 6$) du maximum de vraisemblance avec une version symétrisée des données de Mc Cullagh et Nelder.

5. Particules à la recherche du plan d'expérience optimal

Nous proposons maintenant d'appliquer l'approche particulaire à l'optimisation de plans d'expériences, en restant dans un cadre très général. En effet, aucune hypothèse forte ne sera faite ni sur la structure du modèle (non nécessairement Gaussien ou linéaire par exemple), ni sur les lois *a priori*, ni même sur le critère de l'optimisation (non nécessairement quadratique par exemple).

5.1 Maximisation particulaire de l'utilité espérée

La théorie de l'utilité espérée s'appuie sur la statistique bayésienne pour recommander une décision Δ en situation d'incertitudes sur un état de la nature ϕ. Avec un état de connaissances $H(\Delta)$ conditionnant le pari $[\phi\,|H(\Delta)\,]$ sur les valeurs possibles du vecteur inconnu ϕ, on introduit une fonction d'utilité $w(\Delta,\phi) \geqslant 0$ mesurant les conséquences d'adopter la décision Δ si l'inconnue prend la valeur ϕ. La notion d'utilité est assez commune en économétrie (Von Neumann et Morgenstern, 1944) ou en théorie de la décision (Barnett, 1973; Carlin *et al.*, 1998, Tagaras, 1986). Elle mesure la préférence des décideurs et en cela reste spécifique au problème de décision ou de planification étudié. À ce titre, il est avantageux de proposer une méthode d'optimisation ne nécessitant pas de forme particulière d'utilité. Ici, les seules hypothèses seront que w soit bornée et positive (quitte à la transformer un peu).

En situation risquée, un décideur *rationnel* au sens des axiomes de la théorie de l'utilité espérée (Munier et Parent, 1998; Bernier *et al.*, 2000) prendra la décision Δ^* qui maximise l'utilité espérée $W(\Delta)$.

$$\Delta^* = \arg\max_{\Delta}(W(\Delta)) \tag{5}$$

$$W(\Delta) = \int w(\Delta,\phi)\,[\phi\,|H(\Delta)\,]\,d\phi \tag{6}$$

Müller, 1999, a proposé d'interpréter la quantité $w(\Delta,\phi)\,[\phi\,|H\,]$ comme une fonction proportionnelle à la densité de probabilité f_1 en (Δ,ϕ). On voit alors que le problème (5) est équivalent à trouver le mode de la marginale de f_1 en Δ (cette marginale est proportionnelle à $W(\Delta)$). D'une manière formellement similaire aux cas décrits précédemment, nous sommes ramenés à un problème de recherche du mode d'une densité, et l'approche par simulation de particules peut de nouveau être utilisée à cette fin. L'approche par simulation permet donc de traiter des problèmes d'optimisation très généraux du type (5) qui ne peuvent en général pas être résolus analytiquement.

De nouveau, on obtient un effet de recuit simulé en introduisant T réplicats de ϕ tirés de façon *iid* selon $[\phi\,|H\,]$ et en remarquant que :

$$(W(\Delta))^T = \int ... \int w(\Delta,\phi_1)\,[\phi_1\,|H(\Delta)\,]\,d\phi_1 ... w(\Delta,\phi_T)\,[\phi_T\,|H(\Delta)\,]\,d\phi_T \quad .$$

On cherche cette fois le mode de la distribution $(T+1)$-variée f_T en $(\Delta, \phi_1, .. \phi_T)$ proportionnelle à $\prod_{j=1}^{T} w(\Delta, \phi_j)\,[\phi_j\,|H(\Delta)\,]$. Comme dans l'exemple précédent, les particules de la séquence j de l'algorithme seront du type $(\Delta^{(g)}, \phi_1^{(g)}, ..\phi_j^{(g)})$ et réaliseront un échantillonnage de f_j.

5.2 Planification expérimentale du contrôle de la qualité

Prenons pour exemple un cas de contrôle de la qualité par attribut comme dans Parent *et al.*, 1995. Un lot de taille N est soumis au contrôle, et il faut prendre la décision d'accepter ou de rejeter tout le lot fabriqué. On tire un échantillon de taille n aléatoirement dans ce lot, Y objets ne satisfont pas le contrôle et la règle de décision choisie est : « *Si le nombre d'objets défectueux y est inférieur à un seuil s, accepter le lot, sinon le rejeter* ». L'objectif est de déterminer la taille de l'échantillon n et le niveau de sévérité s. Le vecteur des grandeurs de décision de ce problème est donc $\Delta = (n, s)$. On considère généralement que la probabilité de Y est binomiale ($N \gg n$). Appelons θ la proportion inconnue d'objets défectueux dans tout le lot. En contexte bayésien, on choisit de prendre ici une distribution *a priori* conjuguée *beta* d'hyperparamètres a and b pour décrire la connaissance *a priori* sur θ. Pour ce problème $\phi = (\theta, y)$ et, par abus de notations, on écrira $H = (a, b, n)$. Les préférences du fabricant sont traduites par une fonction d'utilité $u(\Delta, \theta, Y)$ qui décrit les bénéfices associés à la décision Δ et à l'obtention de Y défectueux quand le paramètre du modèle vaut θ. Pour le cas traité, on peut défendre le réalisme du choix de la fonction d'utilité suivante :

$$-u((n,s)\,,(y,\theta)) = (kN)n + (C\theta N) \times 1_{y<s} + N \times 1_{y\geqslant s} \quad .$$

où k est le coût d'échantillonnage ramené à l'unité fabriquée. Si on décide de rejeter, on met au rebut tout le lot et la perte subie est le coût de fabrication (ce coût de fabrication sera pris comme unité monétaire). Si on accepte tout le lot, on laisse partir sur le marché θN objets de qualité non conforme. On imagine que cela engendrera des coûts liés à l'insatisfaction des clients ou à la perte d'image (supposés linéaires) $C \times (\theta N)$. Bien sûr, C et k sont tels que :

$$C > 1 > k \quad .$$

La décision est ici formée du couple $\Delta = (n, s)$ et l'utilité espérée maximale est à rechercher après intégration de $u((n,s)\,,(y,\theta))$ contre les termes aléatoires (y, θ) . La loi jointe de (y, θ) est une loi dite de Polya (classiquement associée au modèle bayésien avec loi *a priori beta* et vraisemblance binomiale, cf. Parent *et al.*, 1996)

$$[(y,\theta)\,|H(\Delta)\,] = \frac{\Gamma(n+1)}{\Gamma(y+1)\Gamma\,(n-y+1)}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta^{a+y-1}\,(1-\theta)^{b+n-y-1} \quad . \quad (7)$$

5.3 Mise en place de l'algorithme et résultats

Il existe une méthode quasi explicite d'optimisation pour ce problème particulier dont le principe est donné dans Parent *et al.*, 1995 et Parent *et al.*, 1996. Ainsi pour $a = 2.18$, $b = 38.18$, $C = 17$, et $k = 1/3000$, on peut montrer que la solution optimale est le plan d'expérience $n^* = 90$ et $s^* = 5$. L'algorithme particulaire peut-il retrouver cette solution ? Pour l'exemple, le domaine de recherche des n possibles a été limité à $n_{\max} = 150$. La fonction d'utilité s'écrit de nouveau $\left(U(n,s) = \int u(\Delta,\phi)\,[\phi\,|H(\Delta)]\,d\phi\right)$ et on note, pour tout n fixé, $s(n) = \arg\max_s(U(n,s))$.

On rencontre deux difficultés spécifiques au cas étudié :

- Les fonctions d'utilité définies ne sont pas positives. On pourrait se ramener à l'étude d'une utilité globale positive en remplaçant U par $-U + Max(U)$, mais pour les simulations des lois jointes, il est nécessaire de définir une utilité sur (Δ, ϕ). On ajoute une constante à u, et l'on travaillera avec $w(\Delta,\phi) = kn_{\max} + (C+1) - u(\Delta,\phi)/N$.
- Avec les valeurs numériques choisies précédemment, cette constante vaut 18.05. Il va falloir un fort effet de recuit simulé pour localiser le mode de la loi marginale en Δ : $W(\Delta) = 18.05 - U(\Delta)$ parce que l'amplitude des variations possibles de $U(\Delta)$ n'est que de l'ordre de 0.25 soit guère plus que 1% de cette constante ajoutée. Ceci est visible sur la forme de $U(n, s(n))$ représentée sur la figure 8 où l'on voit l'allure relativement écrasée et *en dentelles* de la fonction (pourtant dilatante) $n \longmapsto \exp 30(-U(n,s(n)) + Max(U))$. De plus, la forme de l'utilité présente de nombreux modes locaux, ce qui rend classiquement difficile la recherche de l'optimum global.

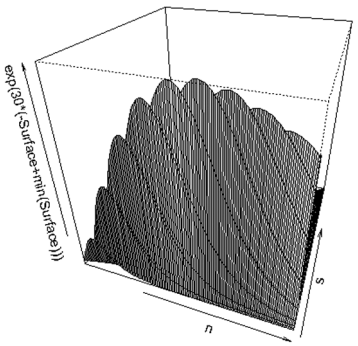

FIG 8. — La fonction d'utilité $U(n, s(n))$ comporte de nombreux modes locaux.

Décrivons l'algorithme particulaire pour passer de f_{i-1} à f_i :

1. Exploration : pour $g = 1, ..., G$, générer $\phi_j^{(g)}$ selon $\left[\phi \left| H(\Delta^{(g)}) \right.\right]$ (c'est-à-dire selon le modèle *beta* binomial) et l'accoler à la particule correspondante issue de la séquence précédente. Pour que le nouvel ensemble particulaire soit un échantillon de f_i il faut corriger par la pondération $w(\Delta^{(g)}, \phi_j^{(g)})$.
2. On effectue un bootstrap multinomial de ces particules avec leurs poids respectifs.
3. Dispersion avec un pas de Metropolis-Hasting : la fonction d'exploration auxiliaire est telle qu'elle propose des candidats $\Delta^\bullet$ décalant n et s symétriquement de ± 1, (les candidats $\phi_l^\bullet$, $l = 1, ..., j$ sont tirés selon le modèle $[\phi \,|H(\Delta^\bullet)]$ et le candidat est accepté si le rapport de Metropolis-Hastings

$$\prod_{l=1}^{j} \frac{w(\Delta^\bullet, \phi_l^\bullet)\,[\phi_l^\bullet \,|H(\Delta^\bullet)]}{w(\Delta, \phi_l)\,[\phi_l \,|H(\Delta)]}$$

est plus grand qu'un tirage annexe uniforme).

La figure 9 montre la distribution de 1000 particules après utilisation de l'algorithme avec $T = 200$. Les niveaux de gris et les lignes de niveaux ont été tracés après lissage par noyaux de l'essaim particulaire final. Les particules ne sont plus autant concentrées que pour les exemples des parties précédentes, mais elles localisent néanmoins les pics principaux (indiqués par les deux flèches) et fournissent une proposition de plan d'expériences proche de l'optimum, exceptionnellement connu ici : $n = 90, s = 5$.

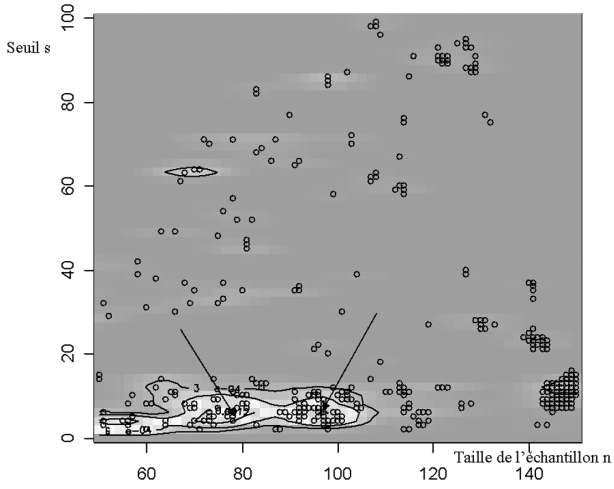

FIG 9. — Deux pics principaux proches de l'optimum théorique de l'utilité espérée sont localisés après une simulation particulaire.

6. Conclusion

Le développement des méthodes de Monte-Carlo a pris un nouvel essor avec les algorithmes particulaires. Tant pour l'implémentation de méthodes bayésiennes que de maximisation de vraisemblance, le statisticien a intérêt à s'approprier ces nouveaux outils. Leur portée dépasse les exemples élémentaires utilisés dans cet article où nous avons montré qu'ils sont efficaces dans des cadres très généraux. Ils sont utiles en inférence bayésienne pour simuler la distribution *a posteriori* et en inférence classique pour rechercher le maximum de vraisemblance dans les modèles à variables latentes. La difficulté d'implémentation informatique n'augmente guère avec la complexité du modèle utilisé. Dans le domaine de la théorie de la décision statistique, ils fournissent un moyen d'atteindre l'optimum d'une intégrale non explicite. Avoir les moyens de surmonter cette partie technique de l'aide à la décision sous incertitudes, offre l'espoir de valoriser les efforts d'explicitation des coûts associés à une décision et de quantification des connaissances issues d'expertise.

L'interprétation intuitive du fonctionnement de ces nouveaux outils se rapproche de celle des algorithmes génétiques (Reeves et Wright, 1996) : une population s'adapte (phase d'échantillonage pondéré), les individus-particules les mieux adaptés ont le plus de chance de survie (bootstrap multinomial) puis surviennent des mutations (phase de régénérescence markovienne). Éventuellement les conditions extérieures deviennent plus dures (phase de recuit simulé) et le cycle recommence. Mais alors que les algorithmes génétiques comportaient une grosse part d'heuristique, le bon comportement quasi systématique des algorithmes particulaires (Del Moral *et al.*, 2001) repose sur des propriétés établies de convergence (voir par exemple Amzal *et al.*, 2006). De plus, les algorithmes génétiques n'offrent pas le niveau de généralité que permet de traiter l'approche particulaire, notamment parce que cette dernière ne nécessite pas une forme particulière du critère d'optimisation et autorise la prise en compte des incertitudes sur les paramètres sans calcul d'intégrales.

Leur comportement algorithmique combine la plupart des avantages (et défauts) des méthodes MCMC et de l'échantillonnage pondéré : héritières des méthodes MCMC, les particules sont *autoportées* vers les modes cibles, mais générées en grand nombre lors de la phase d'échantillonnage pondéré, elles explorent mieux l'espace (à condition que la loi auxiliaire d'importance ne soit pas trop éloignée de la loi que l'on recherche). Enfin un échantillon de la loi cible est disponible à tout moment, sans avoir à attendre la convergence d'une chaîne MCMC et peut être réutilisé pour nourrir une phase de filtrage en séquence.

Néanmoins, le réglage de nombreuses *manettes* peut rendre la pratique des algorithmes particulaires délicate : choix du nombre de particules, paramètrage du modèle (le paramètrage en n et s/n du dernier exemple fonctionne mieux que le paramètrage en n et s), température de recuit simulé et vitesse de progression de cet effet, construction des noyaux de transition markovienne qui conservent la loi cible, etc.

Il y a clairement un champ considérable de problèmes statistiques à explorer grâce à ces nouveaux outils de simulation de Monte-Carlo (Andrieu *et al.*, 2004 ; Cappé *et al.*, 2005). Pouvoir simuler les distributions cibles (et les visualiser au fur et à mesure que les données précisent l'état de la nature) rend l'inférence formelle plus accessible aux praticiens de la statistique. L'essentiel n'est plus une affaire de technique. Seules la pratique et la créativité des lecteurs séparent les exemples pédagogiques traités dans cet article d'applications plus complexes dans des champs très variés : optimisation de prix d'options en finance, réseau de capteurs optimal en hydrologie, plans d'expériences multi-niveaux en pharmacologie, etc.

Annexes

Annexe A : Échantillonnage pondéré

Le chapitre 3 de Robert et Casella, 2004, présente en détail l'échantillonnage pondéré (ou *importance sampling*). C'est une technique de Monte-Carlo pour obtenir un échantillon d'une loi de probabilité cible $p(\psi)$ à partir d'une loi de probabilité (dite loi d'importance) $g(\psi)$ qu'il est facile de simuler.

Algorithme d'échantillonnage pondéré

L'algorithme procède en deux étapes :

1. Pour les itérations $i = 1, ..., G$, générer un échantillon $\psi^i \sim g(\cdot)$ selon la loi d'importance.
2. On définit le poids brut associé au tirage i par $\dfrac{p(\psi^i)}{g(\psi^i)}$. On notera qu'après renormalisation, les poids d'importance $w_i = \dfrac{\dfrac{p(\psi^i)}{g(\psi^i)}}{\displaystyle\sum_{i=1}^{G} \dfrac{p(\psi^i)}{g(\psi^i)}}$ ne font plus intervenir la constante de normalisation de $p(\psi)$ (ni celle de $g(\psi)$).

Propriétés

Il faut bien sûr que $g(\psi) = 0 \Longrightarrow p(\psi) = 0$, c'est-à-dire que le support de g englobe celui de p. On exige aussi que la variance des poids d'importance soit finie :

$$\int \left(\left(h^2(\psi) + 1\right) \frac{p^2(\psi)}{g(\psi)} \right) d\psi < \infty \, .$$

La séquence $\{\psi^i,\ w_i, i = 1, ..., G\}$ représente un G-échantillon pondéré de $p(\psi)$ et les w_i peuvent être interprétés comme la probabilité de tirer ψ^i. Asymptotiquement, on obtient pour toute fonction h un estimateur sans biais

de $\int h(\psi)p(\psi)d\psi$ par

$$E(h(\psi^i)w_i) \approx \int h(\psi)p(\psi)d\psi \approx \sum_{i=1}^{G} h(\psi^i)w_i \ .$$

Sous des conditions techniques de régularité, on peut en plus obtenir un théorème central limite de convergence de $\left(\int h(\psi)p(\psi)d\psi - \sum_{i=1}^{G} h(\psi^i)w_i\right)$ vers une $N(0, \frac{\sigma^2}{G})$.

Annexe B : Méthodes de Monte-Carlo par chaînes de Markov

Les méthodes MCMC sont une famille générique de méthodes pour échantillonner selon une loi cible $p(\psi)$ connue à une constante près, ce qui est notablement le cas rencontré en inférence bayésienne. Ces méthodes ont donné lieu à un important courant de recherche. On trouve dans Kass *et al.*, 1998, et dans Brooks, 1998, un état de l'art et une discussion sur les perpectives d'emploi de ces méthodes. Gilks *et al.*, 1996, en présentent de nombreuses applications.

Algorithme de Metropolis-Hasting

La plupart du temps c'est l'algorithme de Metropolis (Metropolis *et al.*, 1953) qui est mis en œuvre. Il est remarquablement simple et de portée générale :

1. À la $i^{ème}$ itération, on tire un candidat ψ^* selon une loi d'exploration (typiquement multinormale) centrée sur la valeur précédente. Pour cette marche aléatoire $\psi^* \sim g(\cdot \,|\psi^{i-1})$, ψ^{i-1} est la valeur du paramètre obtenu à l'itération précédente. On prend couramment une fonction d'exploration symétrique $g(\psi^* \,|\psi^{i-1}) = g(\psi^{i-1} \,|\psi^*)$. Dans la plupart des études cette loi d'exploration est la multinormale $N(\psi^{i-1}, \lambda\Sigma)$ où Σ est la matrice de variance covariance de ψ ou une approximation et λ est un facteur d'échelle à adapter pour obtenir un taux d'acceptation convenable (voir étape 3).
2. Évaluer le ratio $\frac{p(\psi^*)}{p(\psi)}$ (à noter que la contrainte de normalisation de $p(\psi)$ n'intervient pas dans ce calcul).
3. On effectue un tirage uniforme annexe, $Z \sim U(0,1)$. Si $Z < \frac{p(\psi^*)}{p(\psi)}$, le candidat ψ^* est accepté et on pose $\psi^i \leftarrow \psi^*$. Sinon, si $Z \geqslant \frac{p(\psi^*)}{p(\psi)}$, on reste dans l'état courant que l'on enregistre à nouveau, $\psi^i \leftarrow \psi^{i-1}$. Ce pas d'acceptation/rejet est la clé de l'algorithme. Si l'algorithme ne bougeait que lorsqu'apparaissent des candidats de crédibilité relative plus forte que celles déjà enregistrées, on ne visiterait jamais les régions peu plausibles de $p(\psi)$.

4. Incrémenter i et aller à l'étape 1 jusqu'au nombre G d'itérations souhaitées.

De fait cet algorithme simule une chaîne de Markov qui réalise la séquence d'états $(\psi^1, \psi^2, ...\psi^G)$. Grâce à ses propriétés d'ergodicité, on peut montrer qu'il converge vers une distribution stationnaire qui est justement la loi cible $p(\psi)$ (Parent et Bernier, 2007).

Algorithme de Gibbs

L'échantillonneur de Gibbs (Geman et Geman, 1984) est une méthode MCMC très prisée des praticiens car elle évite le choix d'une fonction d'exploration comme dans l'algorithme de Metropolis. Supposons que le paramètre d'intérêt se décompose par blocs $\psi = (\psi_1, \psi_2 ... \psi_j, ...)$ et que l'on sache simuler le tirage d'une composante ψ_j dans la distribution conditionnelle complète $p(\psi_j \,|\psi_1, \psi_2\,, \ ...\psi_{j-1}, \psi_{j+1}, ...)$. En itérant ces tirages dans les distributions conditionnelles complètes, on crée une chaîne ergodique de distribution stationnaire $p(\psi)$. Bien sûr, pour les modèles complexes, il est rare que l'on dispose de toutes ces conditionnelles. En pratique on combinera des étapes de Gibbs et de Métropolis.

Propriétés

Sous des conditions techniques de comportement en décroissance géométrique de la mémoire de la chaîne markovienne, on peut en plus obtenir un théorème central limite de convergence, quand le nombre d'itérations G tend vers l'infini, de $\sqrt{G}\left(\int h(\psi)p(\psi)d\psi - \sum_{i=1}^{G} h(\psi^i)w_i\right)$ vers une loi $N(0, \sigma^2)$.

Implémentation

Deux problèmes d'implémentation se posent quant à la convergence de la chaîne vers $p(\psi)$. (a) Combien de temps faut-il laisser tourner l'algorithme pour réaliser une approximation correcte de la distribution stationnaire ? On peut lancer en parallèle K séquences à partir de différents points. À la convergence, toutes les séquences doivent provenir de la même loi limite $p(\psi)$ ce qui peut être testé par divers tests paramétriques et non paramétriques. Gelman et Rubin, 1992, proposent une statistique R qui compare les dispersions inter et intra des paramètres générés lors des K séquences. (b) Les résultats théoriques de l'algorithme sont prouvés pour une chaîne homogène, mais en pratique on règle progressivement les caractéristiques de la chaîne (en particulier la variance d'exploration) pour augmenter la vitesse de convergence. Il est en effet tentant d'évaluer la dispersion des valeurs générées et d'ajuster progressivement la variance de la fonction d'exploration. Ce réglage gouverne le taux d'acceptation de l'algorithme pour explorer de façon adéquate le voisinage des ψ's successifs et des règles empiriques de réglage sont proposées dans Gelman *et al.*, 1995).

Tous ces algorithmes supposent bien sûr que la distribution $p(\psi)$ existe ! En particulier si $p(\psi)$ est une distribution a *posteriori* impropre résultant d'une

combinaison non intégrable d'une distribution *a priori* et d'une vraisemblance, les algorithmes MCMC peuvent fort bien générer des échantillons de belle allure d'un objet mathématiquement inexistant !

Bibliographie

Amzal B., Bois F., Parent E., Robert C.P. 2006. Bayesian Optimal Design Via Interacting Particle Systems. *Journal of the American Statistical Association*, **101**(474), 773–785.

Andrieu C., Doucet A. 2000. Simulated Annealing for Maximum a posteriori Parameter Estimation of Hidden Markov Models. *IEEE Trans. Information Theory*, **46**(3), 994–1004.

Andrieu C., Doucet A., Singh S., Tadic V. 2004. Particle Methods for Change Detection, System Identification, and Control. *Proceedings of the IEEE*, **92**(3), 423–438.

Arulampalam M. S., Maskell S., Gordon N., Clapp T. 2002. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transaction on Signal Processing*, **50**(2), 174–188.

Barnett V. 1973. Bayesian and Decision Theoric Methods Applied to Industrial Problems. *The statistician*, **22**(3), 199–226.

Berger J. O. 1985. *Statistical Decision Theory and Bayesian Analysis*. Springer Verlag, New York.

Bernier J., Parent E., Boreux J-J. 2000. *Statistique de l'Environnement. Traitement Bayésien de l'Incertitude*. Paris : Lavoisier.

Brooks S P. 1998. Markov Chain Monte Carlo Method and its Application. *The Statistician*, **47**, 69–100.

Brooks S.P. 2003. Bayesian Computation : A Statistical Revolution. *Trans. Roy. Statist. Soc., series A*, **15**, 2681–2697.

Brooks S.P., Morgan B.J.T. 1995. Optimization Using Simulated Annealing. *The Statistician*, **44**, 241–257.

Cappé O., Guilin A., Marin J.M., Robert C.P. 2004. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, **13**(4), 907–929.

Cappé O., Moulines E., Rydèn T. 2005. *Inference in Hidden Markov Models*. Springer.

Carlin B.P., Kadane J.B., Gelfand A.E. 1998. Approaches for Optimal Sequential Decision Analysis in Clinical Trials. *Biometrics*, **54**(3), 964–975.

Chopin N. 2002. A Sequential Particle Filter Method for Static Models. *Biometrika*, **89**, 539–552.

Chopin N. 2004. Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference. *Ann. Stat.*, **32**(6), 2385–2411.

Del Moral P. 2004. *Feynman-Kac Formulae : Genealogical and Interacting Particle Systems with Applications*. Springer-Verlag.

Del Moral P., Guionnet A. 1999. Central Limit Theorem for Nonlinear Filtering and Interacting Particule Systems. *Ann. Appl. Prob.*, 155–194.

Del Moral P., Kallel L., Rowe J. 2001. *Natural Computing Series : Theoretical Aspects of Evolutionary Computing*. Springer-Verlag, Berlin. Chap. Modelling Genetic Algorithms with Interacting Particle Systems, pages 10–67.

Douc R., Cappé O. 2005. Comparison of Resampling Schemes for Particle Filtering. *Pages 64–69 of : Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*.

Douc R., Cappé O., Moulines E., Robert C.P. 2002. On the Convergence of the Monte Carlo Maximum Likelihood Method for Latent Variable Models. *Scandinavian Journal of Statistics*, **29**(4), 615–635.

Douc R., Guilin A., Marin J.M., Robert C.P. 2007. Convergence of Adaptive Mixtures of Importance Sampling Schemes. *Ann. Stat.*, **35**(1), 420–448.

Doucet A., de Freitas N., Gordon N. 2001. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Doucet A., Godsill J.M., Robert C.P. 2002. Population Marginal maximum a posteriori Estimation Using Markov Chain Monte Carlo. *Statistics and Computing*, **12**, 77–84.

Doucet A., Del Moral P., Peters G.W. 2004. *Sequential Monte Carlo Samplers*. Tech. rept. Cambridge University.

Gelman A., Rubin D. B. 1992. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 457–472.

Gelman A., Carlin J. B., Stern H. S., Rubin D. B. 1995. *Bayesian Data Analysis*. London : Chapman and Hall.

Geman S., Geman D. 1984. Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(6), 721–741.

Gilks W.R., Richardson S., Spiegelhalter D. 1996. *Markov Chain Monte Carlo in Practice*. London : Chapman and Hall.

Kass R. E., Carlin B. P., Gelman A., Neal R. M. 1998. Markov Chain Monte Carlo in Practice : A Roundtable Discussion. *The American Statistician*, **52**(2), 93–100.

Kuhn E., Lavielle M. 2004. Coupling a Stochastic Approximation Version of EM with an MCMC Procedure. *ESAIM, Probab. Stat.*, **8**, 115–131.

Künsch H. 2001. State Space and Hidden Markov Models. *Pages 109–173 of :* Barndor-Nielsen, O. E., Cox, D. R., Klüppelberg, C. (eds), *Complex Stochastic Systems*. Chapman and Hall.

Künsch H. 2005. Recursive Monte Carlo Filters. *Ann. Stat.*, 1983–2021.

Liu J.S., Chen R. 1998. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of American Statistical Association*, **93**(443), 113–119.

Mc Cullagh P., Nelder J.A. 1989. *Generalized Linear Models*. second edn. Chapman and Hall/CRC.

Mengersen K.L., Robert C.P., Guihennec-Jouyaux C. 1999. MCMC Convergence Diagnostics : A Reviewww (with Discussion). *Pages 415–440 of :* Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (eds), *Bayesian Stat.*, vol. 6. Oxford University Press.

Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H, Teller E. 1953. Equations of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, **21**, 1087–1091.

MÜLLER P. 1999. Simulation-Based Optimal Design. Pages 459–474 of : Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (eds), *Bayesian Stat.*, vol. 6. Oxford University Press.

MUNIER B., PARENT E. 1998. Le Développement Récent des Sciences de la Décision : Un Regard Critique sur la Statistique Décisionnelle Bayésienne. *In :* Parent, E., Hubert, P., Bobée, B., Miquel, J. (eds), *Bayesian Methods in Hydrol. Sci.* UNESCO Publishing.

OUJDANE N., RUBERTHALER S. 2005. Stability and Uniform Particle Approximation of Nonlinear Filters in case of non Ergodic Signals. *Stochastic Analysis and Applications*, **23**, 421–448.

PARENT E., BERNIER J. 2007. *Le Raisonnement Bayésien : Modélisation et Inférence*. Paris : Springer Verlag France.

PARENT, E., CHAOUCHE A., GIRARD P. 1995. Sur l'Apport des Statistiques Bayésiennes au Contrôle de la Qualité par Attribut, partie 1 : Contrôle Simple. *Rev. Statistique Appliquée*, **XLIII**(4), 5–18.

PARENT E., LANG G., GIRARD P. 1996. Sur l'Apport des Statistiques Bayésiennes au Contrôle de la Qualité par Attribut, partie 2 : Contrôle Séquentiel Tronqué. *Rev. Statistique Appliquée*, **XLIV**(1), 37–54.

PITT M., SHEPHARD N. 1999. Filtering Via Simulation : Auxiliary Particle Filters. *Ecological Applications*, **94**(446), 590–599.

REEVES C.R., WRIGHT C.C. 1996. Genetic Algorithms and the Design of Experiments. *In : Proc. IMA Fall Workshop on Evolutionary Algorithms.*

ROBERT C. P. 2006. *Le Choix Bayésien : Principes et Pratique*. Paris : Springer Verlag France.

ROBERT C.P., CASELLA G. 2004. *Monte-Carlo Statistical Methods*. Springer-Verlag.

RUBIN D.B. 1988. Using the SIR Algorithm to Simulate Posterior Distributions. *Pages 395–402 of : Bayesian Statistics 3*, vol. 3. Oxford University Press.

SMITH A.F.M., GELFAND A.E. 1992. Bayesian Statistics without Tears : a Sampling-Resampling Perspective. *The American Statistician*, **46**, 84–88.

TAGARAS G. 1986. *Economic Design of Acceptance Sampling and Process Control Procedures for Quality Assurance in Complex Production Systems*. Ph.D. thesis, Stanford University.

TANNER M. H. 1992. *Tools for Statistical Inference : Observed Data and Data Augmentation Methods*. New York : Springer-Verlag.

VAN LAARHOVEN P.J.M., AARTS E.H.L. 1987. *Simulated Annealing : Theory and Applications*. Reider Pub. and Kluwer Dordrecht, Holland.

VON NEUMANN J., MORGENSTERN O. 1944. *Theory of Games and Economic Behavior*. Princeton Univ. Press, Princeton NJ.