The Canadian Journal of Statistics
Vol. xx, No. yy, 20??, Pages 1–25
La revue canadienne de statistique

On the empirical efficiency of local MCMC algorithms with pools of proposals

Mylène Bédard[1]* and Matei Mireuta[2]

[1]*Département de mathématiques et de statistique, Université de Montréal, Montréal, QC, Canada*
[2]*Lady Davis Institute, Montréal, QC, Canada*

Key words and phrases: Computational cost; correlated proposals; delayed rejection strategy ; multiple-try algorithm; regression.

MSC 2010: Primary 65C40; secondary 62F99

Abstract: In an attempt to improve on the Metropolis algorithm, various MCMC methods involving pools of proposals, such as the multiple-try Metropolis and delayed rejection strategies, have been proposed. These methods generate several candidates in a single iteration; accordingly they are computationally more intensive than the Metropolis algorithm. In this paper, we consider three samplers with pools of proposals - the multiple-try Metropolis algorithm, the multiple-try Metropolis hit-and-run algorithm, and the delayed rejection Metropolis algorithm with antithetic proposals - and investigate the net performance of these methods in various contexts. To allow for a fair comparison, the study is carried under optimal mixing conditions for each of these samplers. The algorithms are used in the contexts of Bayesian logistic regressions, inference for a linear regression model, high-dimensional hierarchical model, and bimodal distribution. *The Canadian Journal of Statistics* xx: 1–25; 20?? © 20?? Statistical Society of Canada

Résumé: Afin d'améliorer l'algorithme Metropolis, plusieurs méthodes MCMC impliquant des cohortes de candidats ont été proposées, telles que les stratégies à essais multiples et à rejet retardé. Ces méthodes génèrent plusieurs candidats par itération; leur implémentation est donc associée à un coût computationnel plus élevé que celui de l'algorithme Metropolis. Dans cet article, nous considérons trois approches avec cohortes de candidats - l'algorithme Metropolis à essais multiples, le Metropolis "hit-and-run" à essais multiples et le rejet retardé avec candidats antithétiques - et étudions la perfomance nette de ces méthodes dans différents contextes. Pour que la comparaison soit équitable, chaque échantillonneur est implémenté sous des conditions de mélange optimales. Les algorithmes sont utilisés dans des contextes de régression logistique bayésienne, régression linéaire, modèle hiérarchique en grandes dimensions et distribution bimodale. *La revue canadienne de statistique* xx: 1–25; 20?? © 20?? Société statistique du Canada

1. INTRODUCTION

Metropolis-Hastings algorithms are commonly used to produce samples from arbitrary distributions π that may be complex, high-dimensional, or both (Hastings [1970]). The idea is to build a Markov chain $\{\mathbf{X}[n], n \geq 0\}$ on a state space $\mathcal{X}$ by proposing candidates to be included in the path according to some acceptance probability. The resulting Markov chain is one that is reversible with respect to the target distribution π, and admits π as its unique invariant distribution. Hereafter, π shall also be used for denoting the target density on a state space $\mathcal{X}$ with respect to Lebesgue measure.

To generate candidates for a Markov chain currently at $\mathbf{X}[n] = \mathbf{x}$, a preferred proposal distribution is selected; $q(\mathbf{x}; \cdot)$ denotes the associated proposal density on $\mathcal{X}$ with respect to Lebesgue measure. A candidate $\mathbf{Y}[n+1]$ is then accepted as the next state of the Markov chain with prob-

* *Author to whom correspondence may be addressed.*
E-mail: bedard@dms.umontreal.ca

ability $\alpha(\mathbf{x}; \mathbf{Y}[n+1])$, where

$$\alpha(\mathbf{x}; \mathbf{y}) = 1 \wedge \frac{\pi(\mathbf{y})q(\mathbf{y}; \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x}; \mathbf{y})};$$

if the candidate is rejected (with probability $1 - \alpha(\mathbf{x}; \mathbf{Y}[n+1])$), then the Markov chain remains at state $\mathbf{x}$ for another time step.

In implementing these methods, many proposal distributions are available. Independence samplers are obtained by choosing proposal distributions that are independent of the current state $\mathbf{x}$, so candidates are always drawn according to the same density $q(\cdot)$. In random walk Metropolis (RWM) algorithms, candidates may be expressed as $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is a realization of the increment random variable $\mathbf{Z}$. If the proposal density is also symmetrical around the current state $\mathbf{x}$, i.e. $q(\mathbf{x}; \mathbf{y}) = q(|\mathbf{y} - \mathbf{x}|)$, then the acceptance probability reduces to $\alpha(\mathbf{x}; \mathbf{y}) = 1 \wedge \pi(\mathbf{y})/\pi(\mathbf{x})$.

In order to improve on available samplers, a number of authors have attempted to optimize the usual Metropolis-Hastings proposal scheme by generating a pool of candidates at every iteration (see Tierney and Mira [1999], Liu et al. [2000], Mira [2001a], Mira and Sargent [2003], Craiu and Lemieux [2007]). The multiple-try Metropolis (MTM) and delayed rejection (DR) strategies have been sucessfully applied to challenging problems from various fields of applications, and have recently been studied from a theoretical point of view (see Bédard et al. [2010], Bédard et al. [2012], and the references therein).

Bédard et al. [2010] and Bédard et al. [2012] respectively study the DR and MTM algorithms; these reports focus on candidates generated from a $\mathcal{N}(\mathbf{x}, \sigma^2 I_d)$, where d is the dimension of the target distribution and I_d the d-dimensional identity matrix. The theoretical developments in these papers provide ways to quantify efficiency gains in using pools of candidates, and to improve the samplers considered. They also lead to the determination of the proposal variances and acceptance rates producing optimally mixing chains. The results are derived for high-dimensional target densities with independent and identically distributed (i.i.d.) components. Although these assumptions cannot be deemed realistic from a practical point of view, the associated results are believed to be applicable in greater generality.

Among the samplers considered in these papers, the MTM hit-and-run algorithm (MTM-HR) and the DR algorithm with antithetic proposals (DR-A) seem the most promising (both with Gaussian proposals). According to the asymptotic results derived, these samplers are twice as efficient theoretically as the RWM algorithm with Gaussian proposal (i.e., the speed at which they explore their state space under stationarity is doubled), and can be implemented at a marginal additional cost. Although appealing from a theoretical point of view, these methods have only been applied to a few problems. The aim of this paper is to study the performance of these samplers in various practical settings. Of particular interest are the robustness of the asymptotically optimal acceptance rates derived in Bédard et al. [2010] and Bédard et al. [2012], as well as the extra computational effort required to implement samplers involving pools of proposals.

There exist, in MCMC theory, different notions of efficiency. The term efficiency is used here as a measure of how rapidly the Markov chain explores its state space once stationarity has been reached. For finite-dimensional chains this might be measured by the average quadratic variation (AQV) in (2). In an infinite-dimensional setting, the theoretical efficiency is measured through the speed function of the limiting Langevin diffusion (briefly discussed in Section 3.1). Finally, net efficiency refers to the AQV, corrected to take into account the associated computational effort (Section 4).

In the next section, a motivating example is exposed. The MTM, MTM-HR, and DR-A algorithms are then briefly described, and related optimality results are stated. Sections 4 to 7 address, in order, a Bayesian logistic regression, a linear regression model, a high-dimensional

hierarchical density, and a bimodal distribution.

2. LOGISTIC REGRESSION MODEL WITH LUPUS DATA

In Bédard et al. [2010], a logistic regression model was used to compare the RWM and DR-A algorithms. The aim of the experiment was to predict the occurrence of latent membranous lupus in patients with the help of two clinical covariates. The dataset, containing measurements on 55 patients, can be found in Craiu and Lemieux [2007].

The model $\text{logit}\mathbb{P}(Y_i = 1) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$ was considered, where $X_i' = (1, X_{i1}, X_{i2})$ is the vector of covariates for patient i. To perform a Bayesian analysis, Tan [2006] suggests a $\mathcal{N}(\mathbf{0}, 100^2 I_3)$ prior for $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)'$; this leads to the posterior density

$$\pi(\boldsymbol{\beta}|\mathbf{x}, \mathbf{y}) \propto \exp\Big(-\frac{0.5}{100^2}\boldsymbol{\beta}'\boldsymbol{\beta}\Big) \prod_{i=1}^{55} \frac{[\exp(X_i'\boldsymbol{\beta})]^{y_i}}{1 + \exp(X_i'\boldsymbol{\beta})}.$$

Samples from π were obtained with the RWM and DR-A algorithms; in both cases, a $\mathcal{N}(\boldsymbol{\beta}, \sigma^2 I_3)$ proposal was used to generate candidates, where σ^2 was adjusted so that each sampler explores the state space as rapidly as possible. These tunings relied on the theoretical results introduced in Roberts et al. [1997] and Bédard et al. [2010], and shall be discussed in Section 3.

As in Craiu and Lemieux [2007], the goal was to estimate β_1 and $p_{25} = \mathbf{1}_{\{\beta_1 > 25\}}$. The RWM algorithm was initialized at $\boldsymbol{\beta}[0] = \mathbf{0}$, and 3,064,800 iterations were performed. The first 5,000 values were discarded as burn-in; the remaining values were divided into $\tau = 300$ batches of size $\eta = 10,000$. To reduce correlation, 200 values were discarded between each batch. The estimates, along with their Monte Carlo mean squared error (MC-MSE), were obtained based on the 3,000,000 remaining values. The same steps were then repeated with the DR-A algorithm.

Denoting the j-th replicate of β_1 within the i-th sample by b_{ij}, the MC-MSE of β_1 is

$$\text{MC-MSE}(\beta_1) = (\bar{b}_{\cdot\cdot} - \mathbb{E}[\beta_1|\mathbf{x}, \mathbf{y}])^2 + \frac{1}{\tau - 1}\sum_{i=1}^{\tau}(\bar{b}_{i\cdot} - \bar{b}_{\cdot\cdot})^2, \tag{1}$$

where $\bar{b}_{i\cdot} = \sum_{j=1}^{\eta} b_{ij}/\eta$ $(i = 1, \ldots, \tau)$ and $\bar{b}_{\cdot\cdot} = \sum_{i=1}^{\tau}\sum_{j=1}^{\eta} b_{ij}/(\tau\eta)$. A similar equation may be obtained for p_{25}. The MC-MSEs rely on the approximations $\mathbb{E}[\beta_1|\mathbf{x}, \mathbf{y}] \approx 13.57$ and $\mathbb{E}[p_{25}|\mathbf{x}, \mathbf{y}] \approx 0.073$, obtained by Tan [2006] through numerical integration.

The ratio $\text{MC-MSE}_{\text{DR-A}}(\beta_1)/\text{MC-MSE}_{\text{RWM}}(\beta_1)$ is 0.65. Although the bias of $\bar{b}_{\cdot\cdot}$ (global estimator of β_1) obtained with the RWM sampler is found to be small, the variability between batch estimators $(\bar{b}_{i\cdot}, i = 1, \ldots, 300)$ is important, suggesting a slow-mixing Markov chain. The DR-A algorithm also provides a global estimator that is close to its expected value; futhermore, it explores the state space more efficiently, resulting in a lower variability between batch estimators. This ratio, combined to the fact that η is large, indicates that the RWM method suffers from a very slow mixing. The ratio $\text{MC-MSE}_{\text{DR-A}}(p_{25})/\text{MC-MSE}_{\text{RWM}}(p_{25}) = 0.56$ corroborates the previous assessment.

The average quadratic variation (AQV) is another measure of efficiency, which has the advantage of being independent of specific estimates that we might be interested in computing from the Markov chain. In the present context, it is obtained as

$$\text{AQV} = \frac{1}{N}\sum_{i=0}^{d-1}\sum_{j=1}^{N}(\tilde{b}_j^{(i)} - \tilde{b}_{j-1}^{(i)})^2, \tag{2}$$

where $N = 3,064,800$ is the number of iterations, $d = 3$ refers to the chain dimensions, and $\tilde{b}_j^{(i)}$ is the j-th replicate of β_i. Interestingly, optimizing the AQV is equivalent to minimizing first-order autocorrelations of the chain (see Pasarica and Gelman [2010]).

The ratio of AQVs obtained with the DR-A and RWM methods is 1.8. This reveals that the DR-A sampler makes, on average, larger jumps than the RWM algorithm. Based on these numbers, the DR-A algorithm appears to be, in certain situations, an appealing alternative to the usual RWM sampler. Whether this affirmation is true in general, and whether other MCMC strategies involving pools of proposals may offer a similar improvement, will be discussed in the rest of the paper.

3. LOCAL ALGORITHMS

We briefly describe the local optimization samplers considered, i.e. the MTM, MTM-HR, and DR-A algorithms. In implementing these methods, a pool of candidates is generated in a given iteration. In multiple-try versions, candidates are generated simultaneoulsy while in delayed rejection strategies, they are generated successively.

3.1. Multiple-try Metropolis with independent candidates

The MTM algorithm was introduced by Liu et al. [2000], although the idea of generating multiple candidates per iteration has first appeared in the context of Monte Carlo simulations for molecular dynamics (see Frenkel et al. [1992] and Frenkel and Smit [1996]). In the original MTM method, the pool of candidates generated in an iteration is formed of independent values. The global transition kernel thus consists in the product of the marginal kernels: $q(\mathbf{x}; \mathbf{y}_{1:K}) = \prod_{i=1}^{K} q_i(\mathbf{x}; \mathbf{y}_i)$, where $\mathbf{x}$ is the current value of the chain and $\mathbf{y}_{1:K} = (\mathbf{y}_1, \ldots, \mathbf{y}_K)$ is the pool of K candidates.

We focus here on the case where $q_i(\mathbf{x}; \mathbf{y}_i) = \sigma^{-d}\phi((\mathbf{y}_i - \mathbf{x})/\sigma)$ for $i = 1, \ldots, K$, with $\phi(\cdot)$ the d-dimensional standard Gaussian density and $\sigma > 0$; i.e., candidates are independent Gaussian variables. Intuitively, the proposal variance σ^2 should be a decreasing function of the target dimensions d. Indeed, a constant σ^2 would result in candidates that are rejected more and more frequently as d grows. If, $d = 10$ and σ^2 is fixed (say), then one must generate 10 independent, scalar components to form a candidate; it is thus 10 times more likely to generate an unsuitable component that will lead to the rejection of the candidate than if $d = 1$. It was shown in Bédard et al. [2012] that the proposal variance should be expressed as $\sigma^2 = \ell^2/d$, where ℓ is a positive constant. This allows obtaining weak convergence results about the behavior of the Markov chain (as $d \to \infty$), any other proposal scaling leading to asymptotic processes that are degenerate.

Selecting the value to be proposed among the candidates in the pool can be achieved in different ways; we consider the case where each candidate $\mathbf{y}_i$ in the pool is assigned a probability proportional to its target density, $\pi(\mathbf{y}_i)$. This approach is also equivalent to the orientational-biased Monte Carlo described by Frenkel and Smit [1996]. It has been argued in Liu et al. [2000] that the MTM algorithm does not seem overly sensitive to the choice of the weight function. Bédard et al. [2012] outline the fact that this simple version results in a faster exploration of the state space than, for instance, importance weights, under the framework of high-dimensional target distributions formed of i.i.d. components.

Using the symmetrical proposal kernel described above, the multiple-try RWM sampler can be implemented as follows.

Algorithm 1. (Multiple-try RWM algorithm)

1) Given the time-n state $\mathbf{X}[n] = \mathbf{x}$ of the Markov chain, generate K i.i.d. trial candidates $\mathbf{Y}_{1:K}[n+1]$ according to $\mathbf{Y}_i[n+1] = \mathbf{x} + d^{-1/2}\ell\mathbf{Z}_i[n+1]$, where $\ell > 0$ and $\mathbf{Z}_i[n+1] \sim \mathcal{N}(0, I_d)$ independent for $i = 1, \ldots, K$.

2) *Obtain the selected candidate* $\mathbf{Y}_{k^*}[n+1]$ *by drawing an index* $K^*[n+1] = k^*$ *from a multinomial distribution with parameters proportional to* $\pi\left(\mathbf{Y}_1[n+1]\right), \ldots, \pi\left(\mathbf{Y}_K[n+1]\right)$.

3) *Given* $K^*[n+1] = k^*$, *generate shadow variables* $\mathbf{X}^*_{1:K}[n+1]$ *according to* $\mathbf{X}^*_i[n+1] = \mathbf{Y}_{k^*}[n+1] + d^{-1/2}\ell\mathbf{Z}^*_i[n+1]$, *where* $\mathbf{Z}^*_i[n+1] \sim \mathcal{N}(0, I_d)$ *independent for* $i = 1, \ldots, K-1$, *and* $\mathbf{X}^*_K[n+1] = \mathbf{x}$.

4) *Given* $K^*[n+1] = k^*$, *accept the candidate* $\mathbf{Y}_{k^*}[n+1] = \mathbf{y}_{k^*}$ *with probability* $\alpha^{(K^*[n+1])}\left(\mathbf{X}^*_{1:K}[n+1]; \mathbf{Y}_{1:K}[n+1]\right)$, *where*

$$\alpha^{(k^*)}(\mathbf{x}^*_{1:K}; \mathbf{y}_{1:K}) = \min\left\{1, \sum_{i=1}^{K}\pi(\mathbf{y}_i) / \sum_{i=1}^{K}\pi(\mathbf{x}^*_i)\right\}. \tag{3}$$

The shadow sample $\mathbf{x}^*_{1:K}$ ensures that the reversibility of the Markov chain with respect to π is preserved, and combined to the usual irreducibility and aperiodicity conditions, that the chain converges to the target distribution in total variation distance. Note that the pool of candidates is generated according to the kernel $q(\mathbf{x}; \mathbf{y}_{1:K}) = \prod_{i=1}^{K}\sigma^{-d}\phi((\mathbf{y}_i - \mathbf{x})/\sigma)$ while the shadow sample is generated according to the kernel $q(\mathbf{y}_{k*}; \mathbf{x}^*_{1:(K-1)}) = \prod_{i=1}^{K-1}\sigma^{-d}\phi((\mathbf{x}^*_i - \mathbf{y}_{k*})/\sigma)$.

As with other types of RWM algorithms, it is important to pay a particular attention to the tuning of the proposal variance σ^2 (or ℓ) in the MTM method. To this effect, asymptotic results related to the efficiency of the MTM sampler for high-dimensional target distributions formed of i.i.d. components have recently been obtained.

(**A1**) Consider the d-dimensional target density $\pi(\mathbf{x}) = \prod_{i=1}^{d} f(x_i)$; the one-dimensional density f is assumed to be a positive, twice continuously differentiable function, $[\ln f]''$ is bounded Lipschitz, and $\int f(x)\left|[\ln f]'(x)\right|^4 dx < \infty$.

In the sequel, an alternative formulation for the proposal variance shall reveal useful:

$$\sigma^2 = \frac{\ell^2}{d} \triangleq \frac{\tilde{\ell}^2}{\mathcal{I}d}, \tag{4}$$

where $\mathcal{I} = \int f(x)\left|[\ln f]'(x)\right|^2 dx$ is a measure of roughness of the density f. The first formulation in (4) is useful to describe algorithms, while the second allows stating asymptotically optimal scaling results that are independent of the function f in (A1).

By studying the scaling limit of the Markov chain as $d \to \infty$, it is usually possible to show that RWM samplers involving pools of proposals weakly converge (in the Skorokhod topology) to Langevin diffusion processes. These limiting diffusion processes usually differ through the form of their speed measure. Optimizing the speed measure (the only term depending on $\tilde{\ell}$ in (4)) leads to the asymptotically optimal proposal variance and acceptance rate. The speed measure is the only available efficiency measure in an asymptotic context; all possible finite-dimensional efficiency measures converge to this quantity as $d \to \infty$. The present work is by no means an extensive account on the theory behind optimal scaling results; for more detail about this issue, we refer the reader to Roberts et al. [1997], Bédard et al. [2010], Bédard et al. [2012] for the algorithms considered in this paper, and more generally to Roberts and Rosenthal [2001], Bédard and Rosenthal [2008], Mattingly et al. [2012], and the references therein.

The asymptotically optimal scaling results in Table 1 have been reproduced from Bédard et al. [2012] for K ranging from 1 to 5. It was established from these values that the greatest gain in theoretical efficiency is obtained when going from $K = 1$ (relative efficiency proportional to 1.32) to $K = 2$ (relative efficiency proportional to 2.24). This represents an improvement of

TABLE 1: Asymptotically optimal scaling constants ($\tilde{\ell}^\star$), relative efficiency ($\lambda^\star$), and optimal acceptance rates ($a^\star$) for the MTM algorithm with independent candidates.

K	1	2	3	4	5
$\tilde{\ell}^\star$	2.38	2.64	2.82	2.99	3.12
$\lambda^\star$	1.32	2.24	2.94	3.51	4.00
$a^\star$	0.23	0.32	0.37	0.39	0.41

70% when compared to the RWM method. The marginal efficiency gain brought by additional candidates was shown to decrease with the number of candidates.

3.2. Multiple-try Metropolis hit-and-run algorithm

In extended versions of the MTM method, correlation is included among the candidates constituting the proposal pool. Craiu and Lemieux [2007] have shown how to modify the MTM strategy so as to allow dependent candidates in a given iteration, leading to the multiple correlated-try Metropolis algorithm. Bédard et al. [2012] have considered a more extreme form of dependence, in which all the candidates in the pool are drawn using a common random variable, yielding the multiple-try Metropolis sampler with common proposal (MTM-C algorithm). The acceptance ratio of the latter algorithm has the advantage of not requiring the generation of shadow variables, which are usually necessary to guarantee the reversibility of the Markov chain.

We now consider a specific variant of the MTM-C strategy that appears to be sucessful, namely the multiple-try Metropolis hit-and-run (MTM-HR) algorithm. In the MTM-HR method, candidates in a given iteration are proposed along a common search axis. A Gaussian random vector first determines the search axis, and candidates at various preset distances from the current value are then proposed. There are various ways of selecting the distances; our preferred method is to use regularly spaced step sizes $(\gamma^i)_{i=1}^K$ in $[-\ell, \ell]$ with $\ell > 0$. To this end, it suffices to divide the interval into $K-1$ subintervals of equal lengths. For $K = 2$, $(\gamma^1, \gamma^2) = (-\ell, \ell)$; for $K = 3$, $(\gamma^i)_{i=1}^3 = (-\ell, 0, \ell)$; for $K = 4$, $(\gamma^i)_{i=1}^4 = (-\ell, -\ell/3, \ell/3, \ell)$, and so on. A version of this sampler in which the distances are chosen randomly have been proposed in Liu et al. [2000]; however, this extra randomization does not seem useful, and requires an shadow sample to guarantee reversibility.

The multiple-try RWM hit-and-run method with Gaussian proposal distribution can be implemented as follows.

Algorithm 2. (Multiple-try RWM hit-and-run algorithm)

1) Given $\mathbf{X}[n] = \mathbf{x}$, generate a Gaussian random vector $\mathbf{Z}[n+1] \sim \mathcal{N}(0, I_d)$ and let the K candidates satisfy $\mathbf{Y}_i[n+1] = \mathbf{x} + d^{-1/2}\gamma^i \mathbf{Z}[n+1]$, $i = 1, \ldots, K$.

2) Obtain the selected candidate $\mathbf{Y}_{k^*}[n+1]$ by drawing an index $K^*[n+1] = k^*$ from a multinomial distribution with parameters proportional to $\pi(\mathbf{Y}_1[n+1]), \ldots, \pi(\mathbf{Y}_K[n+1])$.

3) Given $K^*[n+1] = k^*$, accept the candidate $\mathbf{Y}_{k^*}[n+1] = \mathbf{y}_{k^*}$ with probability $\alpha^{(K^*[n+1])}(\mathbf{X}^*_{1:K}[n+1]; \mathbf{Y}_{1:K}[n+1])$, where $\alpha^{(k^*)}$ is given in (3) and

$$\begin{aligned}\mathbf{X}^*_i[n+1] &= \mathbf{x} + d^{-1/2}(\gamma^{k^*} - \gamma^i)\mathbf{Z}[n+1] \\ &= \mathbf{Y}_{k^*}[n+1] - d^{-1/2}\gamma^i \mathbf{Z}[n+1]\end{aligned}$$

for $i = 1, \ldots, K-1$ and $\mathbf{X}^*_K[n+1] = \mathbf{x}$.

The marginal kernels q_i are multivariate Gaussian with mean $\mathbf{x}$ and covariance $(\gamma^i)^2 I_d/d$; when $K = 2$ for instance, the covariance of both marginal kernels satisfies $\ell^2 I_d/d$, where $\ell^2 = \tilde{\ell}^2/\mathcal{I}$, see (4). This corresponds to the case $\Psi^i(\mathbf{x}; \mathbf{v}) = \mathbf{x} + d^{-1/2}\gamma^i \Phi^{-1}(\mathbf{v})$ and $\Psi^{k^*,i}(\mathbf{x}; \mathbf{y}) = \mathbf{x} + (\gamma^i/\gamma^{k^*})(\mathbf{y} - \mathbf{x})$ in the MTM-C algorithm of Bédard et al. [2012]; here, Φ denotes the standard Gaussian cumulative distribution function. Hence, candidates may be expressed as a function of $\mathbf{Y}_{K^*[n+1]}[n+1]$ through $\Psi^{k^*;i}(\mathbf{x}, \mathbf{y})$, and variables acting as the shadow sample may also be obtained directly from $\mathbf{Y}_{K^*[n+1]}[n+1]$ through the relationship specified in Step (3) of Algorithm 3.2.

Based on some numerical explorations, this sampler seems quite promising in practice; the theoretical results derived in Bédard et al. [2012] corroborate these impressions. Asymptotic results similar to those outlined in Section 3.1 have been obtained for the MTM-C method, and in particular for the MTM-HR sampler. The asymptotically optimal scaling results of Table 2 have been reproduced from Bédard et al. [2012] for various values of K.

TABLE 2: Asymptotically optimal scaling constants ($\tilde{\ell}^\star$), relative efficiency ($\lambda^\star$), and optimal acceptance rates ($a^\star$) for the multiple-try RWM hit-and-run algorithm.

K	1	2	4	6	8
$\tilde{\ell}^\star$	2.38	2.37	7.11	11.85	16.75
$\lambda^\star$	1.32	2.64	2.65	2.65	2.65
$a^\star$	0.23	0.46	0.46	0.46	0.46

It was established that the theoretical efficiency of the MTM-HR algorithm doubles under optimal conditions when going from $K = 1$ to $K = 2$, but then stagnates as the number of trials continues to grow. The acceptance rate is in fact dominated by the candidates that are closest to the current value $\mathbf{x}$. To optimize the mixing of the chain as K grows, it is thus necessary to expand the search interval $[-\ell, \ell]$ such that the values closest to $\mathbf{x}$ remain the same as when $K = 2$. Suppose $K = 4$; from Table 2, the asymptotically optimal search region is $[\mathbf{x} - 7.11/(\mathcal{I}d)^{1/2}, \mathbf{x} + 7.11/(\mathcal{I}d)^{1/2}]$. In a given iteration, the sampler thus considers the four candidates $\mathbf{x} \pm 7.11/(\mathcal{I}d)^{1/2}$, $\mathbf{x} \pm 2.37/(\mathcal{I}d)^{1/2}$. The candidates closest to $\mathbf{x}$ coincide with the candidates when $K = 2$.

MTM-HR algorithms with $K = 2$ should then generally be favored over algorithms with $K > 2$, which suffer from a pathological behavior. The case $K = 2$ also is interesting from a computational viewpoint: the symmetry between $\mathbf{Y}_1$ and $\mathbf{Y}_2$ around the current $\mathbf{x}$ often leads to cheaper evaluations of the target density at these points.

3.3. Delayed rejection algorithm with antithetic proposals

The DR strategy has been introduced in Tierney and Mira [1999], then further studied in Mira [2001a] and Green and Mira [2001]. This strategy aims at improving the Metropolis algorithm by reducing the proportion of iterations in which no candidate is accepted. In these methods, the candidates in a given iteration are generated successively, and the idea is to learn from past rejections in order to refine the quality of the generated candidates. The delayed rejection scheme is quite general; it may be applied, for instance, by combining a first-level independence proposal with a second-level random-walk proposal, or by designing second-level proposals based on the target density of the rejected candidates, etc.

As mentioned in Mira [2001a], a particularly interesting and easy way to implement the delayed rejection scheme is to use symmetrical, random walk increments. Given $\mathbf{X}[n] = \mathbf{x}$, generate a Gaussian random vector $\mathbf{Z}_1[n+1] \sim \mathcal{N}(0, I_d)$ and let the first candidate satisfy $\mathbf{Y}_1[n+1] = \mathbf{x} + d^{-1/2}\ell_1 \mathbf{Z}_1[n+1]$. Accept this candidate with probability $\alpha_1(\mathbf{x}; \mathbf{Y}_1[n+1])$,

where $\alpha_1(\mathbf{x},\mathbf{y}) = 1 \wedge \pi(\mathbf{y})/\pi(\mathbf{x})$. Upon the rejection of $\mathbf{Y}_1[n+1]$, generate a second candidate $\mathbf{Y}_2[n+1] = \mathbf{x} + d^{-1/2}\ell_2\mathbf{Z}_2[n+1]$ with $\mathbf{Z}_2[n+1] \sim \mathcal{N}(0, I_d)$ (independent of $\mathbf{Z}_1[n+1]$); accept it with probability $\alpha_2(\mathbf{x}, \mathbf{Y}_1[n+1]; \mathbf{Y}_2[n+1])$, where

$$\alpha_2(\mathbf{x}, \mathbf{y}_1; \mathbf{y}_2) = 1 \wedge \frac{\pi(\mathbf{y}_2)q_1(\mathbf{y}_2; \mathbf{y}_1)[1 - \pi(\mathbf{y}_1)/\pi(\mathbf{y}_2)]_+}{\pi(\mathbf{x})q_1(\mathbf{x}; \mathbf{y}_1)[1 - \pi(\mathbf{y}_1)/\pi(\mathbf{x})]_+}, \tag{5}$$

$a_+ = \max(a, 0)$ and $q_1(\mathbf{x}; \cdot)$ is a $\mathcal{N}(\mathbf{x}, \ell_1^2 I_d/d)$ density. Then, move on to the next iteration.

The second-level acceptance probability α_2 has been determined so as to preserve the reversibility of the chain with respect to π; this is achieved by requiring the forward and reverse paths to go through the same intermediate value. Suppose, for instance, that the chain currently at $\mathbf{x}$ rejects a first candidate $\mathbf{y}_1$, and then accepts a second candidate $\mathbf{y}_2$. Equation (5) ensures that the reverse path (denominator) goes from $\mathbf{y}_2$ to $\mathbf{x}$ via that same intermediate candidate $\mathbf{y}_1$. It is also possible to extend the proposal scheme to $K > 2$ candidates per iteration, but these extensions shall not be discussed further.

The sampler described above is a DR algorithm in which both random walk increments in an iteration are generated independently from each other, hereafter referred to as DR-I algorithm. In this case, intuition tells us that upon the rejection of a first candidate, one should try a second, more conservative candidate (i.e. $\ell_2 < \ell_1$). It was recently proved in Bédard et al. [2010] that this combination of proposal distributions leads to a pathological algorithm for high-dimensional target densities satisfying Assumption (A1). Indeed, this specific DR strategy is shown to be asymptotically equal to the RWM algorithm, while demanding a higher computational cost. In high dimensions, it generates second-level candidates that are extremely close to the current $\mathbf{x}$; in the limit, proposing a second-level candidate in a given iteration thus becomes equivalent to simply rejecting the candidate and remaining at the same state for another time interval.

As mentioned in Green and Mira [2001], preserving reversibility by requiring the intermediate candidate $\mathbf{y}_1$ to be the same in each direction does not allow an optimal use of the DR strategy. In that paper, the authors introduce a generalization of the DR strategy for trans-dimensional target distributions; they achieve this goal by deriving an expression for α_2 that preserves reversibility without forcing the return path from $\mathbf{y}_2$ to $\mathbf{x}$ to go through $\mathbf{y}_1$. It thus becomes necessary to replace the intermediate value $\mathbf{y}_1$ in the reverse path by an alternative auxiliary variable. This approach can also be enforced in a fixed dimensional setting, possibly preventing the kind of second-level degeneracy faced with the DR-I sampler. There exist potentially many ways of making use of this approach; in the remainder of this section, we present one of them.

An issue of the DR-I is its inability to make use of the information available, i.e. the rejection of a $\mathbf{Y}_1[n+1]$, to propose a second candidate. To fill this gap, Bédard et al. [2010] proposed a version of the DR method in which candidates are correlated. In particular, if $\mathbf{Y}_1[n+1]$ is rejected, then it makes sense to believe that a second, more conservative candidate should be proposed *in the same direction* $\mathbf{Z}_1[n+1]$, or even *in the opposite direction* $-\mathbf{Z}_1[n+1]$.

In such a case, the first candidate is set to $\mathbf{Y}_1[n+1] = \mathbf{x} + d^{-1/2}\ell_1\mathbf{Z}[n+1]$, where $\mathbf{Z}[n+1]$ is a symmetrical random vector and ℓ_1 is a scale parameter. To propose a second candidate along the same search axis, let $\mathbf{Y}_2[n+1] = \mathbf{x} + d^{-1/2}\ell_2\mathbf{Z}[n+1] = \Psi(\mathbf{x}; \mathbf{Y}_1[n+1])$, where

$$\Psi(\mathbf{x}; \mathbf{y}) = \mathbf{x} + \ell_2\ell_1^{-1}(\mathbf{y} - \mathbf{x}). \tag{6}$$

Just like $\mathbf{Y}_2[n+1]$ is obtained deterministically from $\mathbf{x}$ and $\mathbf{Y}_1[n+1]$, knowledge of $\mathbf{x}$ and $\mathbf{Y}_2[n+1]$ brings us back to $\mathbf{Y}_1[n+1]$: $\mathbf{Y}_1[n+1] = \bar{\Psi}(\mathbf{x}; \mathbf{Y}_2[n+1])$, where

$$\bar{\Psi}(\mathbf{x}; \mathbf{y}) = \mathbf{x} + \ell_1\ell_2^{-1}(\mathbf{y} - \mathbf{x}). \tag{7}$$

Under this scheme, $\ell_2 > 0$ consists in proposing a second candidate heading in the same direction as $\mathbf{Y}_1[n+1]$, while $\ell_2 < 0$ corresponds to a move in the opposite direction.

The delayed rejection Metropolis algorithm with common Gaussian random walk increment (DR-A) is implemented as follows.

Algorithm 3. DR-A algorithm

1) Given $\mathbf{X}[n] = \mathbf{x}$*, generate a Gaussian random vector* $\mathbf{Z}[n+1] \sim \mathcal{N}(0, I_d)$ *and let the first candidate satisfy* $\mathbf{Y}_1[n+1] = \mathbf{x} + d^{-1/2}\ell_1\mathbf{Z}[n+1]$*; accept this candidate with probability* $\alpha_1(\mathbf{x}; \mathbf{Y}_1[n+1])$*, where* $\alpha_1(\mathbf{x}; \mathbf{y}) = 1 \wedge \pi(\mathbf{y})/\pi(\mathbf{x})$.

2) Upon the rejection of $\mathbf{Y}_1[n+1]$*, generate a second candidate*

$$\mathbf{Y}_2[n+1] = \mathbf{x} + d^{-1/2}\ell_2\mathbf{Z}[n+1] = \Psi(\mathbf{x}; \mathbf{Y}_1[n+1]),$$

with Ψ *as in (6); accept this candidate with probability* $\alpha_2(\mathbf{x}; \mathbf{Y}_2[n+1])$*, where*

$$\begin{aligned}\alpha_2(\mathbf{x}; \mathbf{y}) &= 1 \wedge \frac{\pi(\mathbf{y})[1 - \pi(\bar{\Psi}(\mathbf{y}; \mathbf{x}))/\pi(\mathbf{y})]_+}{\pi(\mathbf{x})[1 - \pi(\bar{\Psi}(\mathbf{x}; \mathbf{y}))/\pi(\mathbf{x})]_+} \\ &= 1 \wedge \frac{\pi(\mathbf{y})[1 - \pi(\mathbf{y} + \ell_1\ell_2^{-1}(\mathbf{x} - \mathbf{y}))/\pi(\mathbf{y})]_+}{\pi(\mathbf{x})[1 - \pi(\mathbf{x} + \ell_1\ell_2^{-1}(\mathbf{y} - \mathbf{x}))/\pi(\mathbf{x})]_+},\end{aligned}$$

with $\bar{\Psi}$ *as in (7).*

Bédard et al. [2010] shows that to optimize efficiency among all possible candidates $\mathbf{Y}_1, \mathbf{Y}_2$ along a common search axis, the asymptotically optimal proposal variances should be of the form $\sigma_i^2 = \ell_i^2/d \triangleq \tilde{\ell}_i^2/(\mathcal{I}d)$, $i = 1, 2$, with $\tilde{\ell}_1 = -\tilde{\ell}_2 = 2.37$; in other words, the second candidate should be exactly symmetrical to the first one around the current value $\mathbf{x}$. This optimal tuning yields an asymptotically optimal acceptance rate of 0.46 (i.e., an acceptance rate of 0.23 for both first- and second-level candidates), as well as a speed measure that is doubled compared to that of the RWM method (relative efficiency of 2.64 versus 1.32) when Assumption (A1) is satisfied. Although the asymptotic behaviors of the DR-A and MTM-HR algorithms with $K = 2$ are generally extremely close, there exist significant differences in their implementations.

The DR-A algorithm with $K > 2$ has yet to be considered; an extension where K candidates are successively generated along a common search axis is not particularly appealing, as marginal efficiency gains from the inclusion of additional candidates would necessarily decrease with K. It would however be possible to consider extensions in which additional candidates would be moving in a different (but deterministically chosen) direction, see Bédard et al. [2010].

On a related subject, the pinball sampler introduced in Robert and Mengersen [2003], is a method based on self-avoiding particle filters that shares many similarities with the DR-A sampler. This method generates simultaneously, at time n, a vector of N random variables $\mathbf{X}_{1:N}[n]$, where N is the size of the sample one wishes to obtain. The idea is to implement a Metropolis-within-Metropolis sampler in which the particles are connected via a repulsive proposal. Using symmetry arguments, a deterministic delayed rejection mechanism is included (with the possibility of having $K > 2$), which is referred to as the pinball effect.

Before considering empirical studies in Sections 4 to 8, where we further investigate the relationships between random walk versions of the Metropolis, MTM, MTM-HR, and DR-A algorithms, we address the computational effort of these methods.

3.4. Computational effort

Many researchers have developed new samplers in an effort to improve upon basic methods such as the RWM algorithm; nonetheless, when taking computational complexity into account, it is often the case that traditional samplers remain competitive. In implementing MCMC methods, it is generally the evaluation of the target density π at specific values that requires the greatest effort. MCMC methods with pools of proposals are not conceptually complex, but involve the evaluation of π at several points, which generally affects the empirical performance of the algorithm.

It is usually difficult to compare the empirical efficiency of samplers without considering specific examples with specific estimates. Accordingly, the convention is to scale the chosen efficiency measures by the number of points at which the density needs to be evaluated during a given iteration. In the case of the MTM algorithm with independent candidates, this number rapidly increases with the number of candidates in a given iteration. In a K-try Metropolis sampler, a pool of K candidates is generated, along with a shadow sample of size $K-1$; efficiency measures are thus divided by a factor of $2K-1$.

Just as the previous MTM method, the MTM-HR algorithm requires evaluating the target density at $2K-1$ points in a given iteration. The design of this method however guarantees that all candidates are along a common search axis, which likely reduces the overhead introduced by the computation of several likelihoods. The exact extent of the bargain in evaluating several likelihoods is however difficult to quantify, but practitioners should recognize the conservative character of the factor $2K-1$ in this case.

The DR-A algorithm involves evaluating the target density at 3 points for some, but not all iterations. Indeed, if the first-level candidate $\mathbf{Y}_1$ is accepted, then a second-level candidate is not required. Accordingly, for optimally tuned DR-A algorithms (i.e. a first-level acceptance rate of 0.23, see Section 3.3), we should correct efficiency measures by a factor of 2.5 (instead of 3) to account for computational effort. All likelihoods are again evaluated along a common search axis, so the computational effort in evaluating the target density at three values (instead of one in the RWM method) is generally modest. Because of the sequential rather than parallel strategy used to generate additional samples, we thus expect an additional efficiency gain of the DR-A versus the MTM-HR. However, it is still difficult to quantify in general terms.

Following the previous discussion we realize that although samplers with pools of proposals are more efficient from a theoretical viewpoint, it is still unclear whether these methods are worth implementing in practice. This uncertainty is mainly due to the complexity of quantifying gains in terms of computational effort that stem from the evaluation of several likelihoods in a common search axis. Specifically, we strongly suspect that correcting efficiency by a factor of $2K-1$ is too conservative in such cases, but to what extent? We attempt to provide a partial answer to this question by considering four empirical studies in which the performances of the algorithms previously discussed are compared. Based on the theoretical results previously exposed, we shall focus on cases where the pool of candidates is formed of two candidates only. Apart from the fact that extensions of the DR-A method to the case $K>2$ are a separate ongoing line of research, this decision is mainly based on the fact that the most significant efficiency gains are obtained when going from $K=1$ to $K=2$ candidates in both the MTM and MTM-HR algorithms.

4. BAYESIAN ANALYSIS FOR A LOGISTIC REGRESSION MODEL

In this section, we apply various local MCMC methods to estimate a logistic regression model for the number of survivals in a sample of 79 subjects suffering from a certain illness. The dataset, which has been analyzed by Dellaportas et al. [2002], illustrates the effects of the patients' condition (more or less severe) and the treatment received (antitoxin or not) on survival.

We consider the full logistic regression model where the number of survivals is the response variable, and where the patient's condition (a) and the received treatment (b), are the explanatory variables. The full model is given by

$$Y_{ij} \sim Bin(n_{ij}, p_{ij}) \,, \quad \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \mu + a_i + b_j + (ab)_{ij}, \quad i,j = 1,2,$$

where Y_{ij}, n_{ij} and p_{ij} respectively represent the number of survivals, the total number of patients and the probability of survival given condition i and treatment j. The parameter of the model may be expressed as $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)' = (\mu, a_2, b_2, (ab)_{22})'$ by considering the intercept μ as the coefficient for a baseline group (condition less severe with no treatment), and by interpreting the other parameters as incremental effects for other groups compared to the baseline group.

As in Dellaportas et al. [2002], we use a Bayesian approach and rely on a $\mathcal{N}(0, 8I_4)$ prior distribution for $\boldsymbol{\beta}$. The posterior distribution may thus be expressed as

$$\pi(\boldsymbol{\beta}|\text{data}) \propto \frac{\left(\mathrm{e}^{\beta_0+\beta_1+\beta_2+\beta_3}\right)^6}{\left(1+\mathrm{e}^{\beta_0+\beta_1+\beta_2+\beta_3}\right)^{21}} \frac{\left(\mathrm{e}^{\beta_0+\beta_1}\right)^4}{\left(1+\mathrm{e}^{\beta_0+\beta_1}\right)^{26}} \frac{\left(\mathrm{e}^{\beta_0+\beta_2}\right)^{15}}{\left(1+\mathrm{e}^{\beta_0+\beta_2}\right)^{20}} \frac{\left(\mathrm{e}^{\beta_0}\right)^5}{\left(1+\mathrm{e}^{\beta_0}\right)^{12}} \mathrm{e}^{-\frac{1}{2(8)}\sum_{i=0}^3 \beta_i^2}.$$

We compare the performances of four samplers in the estimation of the model parameter $\boldsymbol{\beta}$: a random walk Metropolis (RWM) algorithm, a two-step multiple-try RWM algorithm with independent candidates (MTM), a hit-and-run version of the two-step multiple-try RWM algorithm (MTM-HR), and a delayed rejection algorithm with antithetic candidates (DR-A). Gaussian proposal distributions are chosen in all four cases. To allow for a fair comparison, we rely on optimal versions of the various algorithms. This is achieved by tuning the acceptance rates of the algorithms to be as close as possible to the asymptotically optimal acceptance rates (AOARs) mentioned in Section 3. Numerical exploration confirms that these acceptance rates are close to optimal, even in the current low-dimensional, correlated context.

To estimate $\boldsymbol{\beta}$, we perform runs of 5,104,900 iterations with a starting value of $\boldsymbol{\beta}[0] = \mathbf{0}$. The first 5,000 iterations are discarded as burn-in; the rest of the sample is divided into batches of 5,100 values, of which the last 100 are discarded in order to create approximately independent batches. Each of the 1,000 batches is used to estimate the parameters of the model, and we then record the sample mean of these 1,000 estimates to obtain a global estimate. As a measure of efficiency for the algorithms, we also record the sample variance of the 1,000 batch estimators obtained; this is represented by the second term of the MC-MSE in (1), and approximates the asymptotic variance of the MCMC estimator. As a second measure of efficiency, we record the AQV of each algorithm, as defined in (2).

Candidates may be expressed as $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} + \sigma\mathbf{z}$, where $\boldsymbol{\beta}$ is the current value and $\mathbf{z}$ comes from a $\mathcal{N}(0, I_4)$. The target density at $\tilde{\boldsymbol{\beta}}$ is thus

$$\begin{aligned}\pi(\tilde{\boldsymbol{\beta}}|\text{data}) \propto & \frac{\left(\mathrm{e}^{\sum_{i=0}^3 \beta_i + \sigma\sum_{i=0}^3 z_i}\right)^6}{\left(1+\mathrm{e}^{\sum_{i=0}^3 \beta_i + \sigma\sum_{i=0}^3 z_i}\right)^{21}} \frac{\left(\mathrm{e}^{\sum_{i=0}^1 \beta_i + \sigma\sum_{i=0}^1 z_i}\right)^4}{\left(1+\mathrm{e}^{\sum_{i=0}^1 \beta_i + \sigma\sum_{i=0}^1 z_i}\right)^{26}} \\ & \times \frac{\left(\mathrm{e}^{\sum_{i=0,2} \beta_i + \sigma\sum_{i=0,2} z_i}\right)^{15}}{\left(1+\mathrm{e}^{\sum_{i=0,2} \beta_i + \sigma\sum_{i=0,2} z_i}\right)^{20}} \frac{\left(\mathrm{e}^{\beta_0+\sigma z_0}\right)^5}{\left(1+\mathrm{e}^{\beta_0+\sigma z_0}\right)^{12}} \mathrm{e}^{-\frac{1}{2(8)}\sum_{i=0}^3 (\beta_i+\sigma z_i)^2}.\end{aligned}$$

Evaluating the posterior density at $\tilde{\boldsymbol{\beta}}$ involves terms of the form $\sum \beta_i + \sigma \sum z_i$. Summations involving β_is are already available from the computation of the posterior density at $\boldsymbol{\beta}$. In a

TABLE 3: Estimates of the parameters and their corresponding simulation variances for the logistic regression model of Section 4.

		β_0	β_1	β_2	β_3
RWM	Estimate	-0.3186	-1.4535	1.4118	-0.5875
	(Simul. var.)	(0.00345)	(0.00812)	(0.00702)	(0.01627)
MTM	Estimate	-0.3205	-1.4487	1.4133	-0.5920
	(Simul. var.)	(0.00209)	(0.00453)	(0.00414)	(0.00923)
MTM-HR	Estimate	-0.3204	-1.4480	1.4122	-0.5927
	(Simul. var.)	(0.00205)	(0.00445)	(0.00406)	(0.00863)
DR-A	Estimate	-0.3227	-1.4455	1.4151	-0.5951
	(Simul. var.)	(0.00222)	(0.00484)	(0.00436)	(0.00968)

TABLE 4: Proposal variances selected and acceptance rates obtained for the samplers implemented. AQVs, net AQVs, and computational time (in seconds) are provided. Asymptotically optimal acceptance rates (AOARs) are also included for comparison.

	σ^2	Acc. rate	AQV	Time	Net AQV	AOAR
RWM	0.35	0.223	0.1976	592	0.1976	0.23
MTM	0.45	0.311	0.3297	1635	0.1194	0.32
MTM-HR	0.35	0.405	0.3785	1236	0.1813	0.46
DR-A	0.35	0.404[a]	0.3771	794	0.2812	0.46

[a] The first- and second-level acceptance rates respectively are 0.223 and 0.180.

RWM algorithm, all that is needed to obtain the acceptance probability are thus the summations involving z_is. It turns out that once these summations are available, evaluating the posterior density at values $\boldsymbol{\beta} + \kappa\sigma\mathbf{z}$ ($\kappa \in \mathbb{R}$) is straight-forward. Obtaining the acceptance probabilities of the MTM-HR and DR-A (for which $\kappa = -1, 1$) at a reasonable computational cost therefore becomes possible. This is generally true, among other cases, for logistic regression analyses.

Table 3 provides estimates of $\boldsymbol{\beta}$ using the four samplers, along with estimates of their asymptotic variance (hereafter referred to as simulation variance). Table 4 contains the proposal variances used, along with the resulting acceptance rates and AQVs. The time (in seconds) for running the various methods is provided (using `system.time` on the freeware `R`), along with an adjusted measure of the AQV that takes into account the computation time of each method. The adjustment is simple: by consering the RWM algorithm as the baseline method, the net AQV for the MTM sampler (say) is obtained by applying the factor time(RWM)/time(MTM).

On the one hand, by naively looking at the simulation variances and AQVs in Tables 3 and 4, one could be tempted to conclude that all three samplers with pools of proposals are significantly more efficient than the RWM algorithm. On the other hand, by naively adjusting for computational effort according to the rule of thumb mentioned in Section 3.4 (i.e. by applying a factor of 3 in the present case), one might incorrectly claim that the RWM algorithm remains the best

available option.

In the present setting, this rule of thumb is reasonable for the MTM method with independent candidates since the target density should be evaluated at three independent values per iteration; this is supported by the running times in Table 4. Based on net performances, the MTM sampler with independent candidates is thus less efficient than the usual RWM algorithm.

In the case of the MTM-HR method, dividing the AQV by a factor of 3 would be too conservative an adjustment, but the extra computational effort required for implementing this sampler is hard to quantify theoretically. For this specific example, running the MTM-HR is twice as long as the RWM. Given that the theoretical efficiency is doubled, the RWM and MTM-HR algorithms are equivalent.

Finally, the DR-A strategy seems to constitute the best of the four available options: its AQV is roughly twice that of the RWM method. We find that dividing the AQV by a factor of 3 would again be too conservative an adjustment. This algorithm does not generate a second candidate at every iteration, but only when the first candidate is rejected; the correction factor should thus be closer to 2.5 than to 3. Further gains are also available from the symmetry between the candidates. In this case, running the DR-A algorithm results in an increase of 35% in terms of computational time. Given that the AQV is almost doubled, this results in a net AQV that is about 40% higher than for the RWM sampler. The DR-A algorithm seems like the most efficient option to estimate the parameters of this model.

It is interesting to note that the tunings of the RWM, MTM, MTM-HR, and DR-A algorithms seem quite robust to the dimension of the target density and the correlation between its components: despite the fact that the target density is a four-dimensional, correlated density, the relationships among the various samplers are concordant with the theoretical results available for infinite-dimensional target densities with independent components. Finally, despite the similarities between the MTM-HR and DR-A methods, the former appears to be significantly more demanding computationally.

5. CLASSICAL INFERENCE FOR A LINEAR REGRESSION MODEL

We now analyze a dataset concerning the cost of construction of nuclear power plants (Example G, Cox and Snell [1981]). We possess information about the capital cost of 32 light water reactor power plants constructed in the U.S.A., each of which is associated to 10 explanatory variables.

According to the analysis in Cox and Snell [1981] and Brazzale et al. [2007], this dataset is modeled using a linear regression model in which the quantitative variables are log-transformed. The model is described using 6 explanatory variables that have been deemed significant (see the ANOVA in Cox and Snell [1981]). Of particular interest is how the capital cost depends on the cumulative number of power plants constructed by each architect-engineer.

As in Brazzale et al. [2007], the errors are assumed to be distributed according to a Student distribution with 4 degrees of freedom. The model is thus expressed as

$$f\left(\mathbf{y};\boldsymbol{\beta},\sigma\right)d\mathbf{y}=\sigma^{-m}\prod_{i=1}^{m}h\left(\frac{y_i-X_i\boldsymbol{\beta}}{\sigma}\right)dy_i,$$

where $h\left(\cdot\right)$ is the Student(4) density, $\boldsymbol{\beta}=\left(\beta_1,\ldots,\beta_7\right)'$ is the vector of parameters (including an intercept), and X_i is the ith row of the 32×7 design matrix X (containing the chosen explanatory variables plus a constant).

We record the observed standardized residuals as $\mathbf{r}^0=\left(\mathbf{y}^0-X\mathbf{b}^0\right)/s^0$, where $\mathbf{b}$ contains the least squares regression coefficients and s represents the related error standard deviation: $s^2=\sum_{i=1}^{m}(y_i-\hat{y}_i)^2/(m-d)$ with $m=32$ and $d=7$. As the residuals $\mathbf{r}^0$ have an effect on the

precision of the estimates of $\boldsymbol{\beta}$ and σ, we use the model $f\left(\mathbf{y}\left|\mathbf{r}^0\right.;\boldsymbol{\beta},\sigma\right)$, obtained by conditioning on the identified standardized residuals $\mathbf{r}^0$:

$$f\left(\mathbf{b},s\left|\mathbf{r}^0\right.;\boldsymbol{\beta},\sigma\right)d\mathbf{b}\,ds = c\sigma^{-m}\prod_{i=1}^{m}h\left(\frac{sr_i^0 - X_i\left(\boldsymbol{\beta}-\mathbf{b}\right)}{\sigma}\right)s^{m-d-1}d\mathbf{b}\,ds,$$

where $m = 32$, $m-d-1 = 24$, and $c > 0$ is constant. Performing the hypothesis testing with the model $f\left(\mathbf{y};\boldsymbol{\beta},\sigma\right)$ would not take this available information into account.

We are interested here in β_6, the 6th regression coefficient. The observed standardized departure of data from parameter value is $t_6^0(\beta_6) = (b_6^0 - \beta_6)/c_{6,6}^{1/2}s^0$, where $c_{6,6}$ is the $(6,6)$ element of the matrix $(X'X)^{-1}$; this quantity obviously depends on the value β_6 being assessed.

To obtain a statistical interpretation of this measure, we need information about the distribution of possible values for the standardized departure $t_6 = (b_6 - \beta_6)/c_{6,6}^{1/2}s$ in the context where the true parameter value is β_6. The observed p-value is thus the percentage position of the data with respect to the hypothesized value β_6.

There exists various methods for approximating p-values in the classical framework. In the present context, computing exact p-values can be achieved by sampling the conditional model $f\left(\mathbf{b},s\left|\mathbf{r}^0\right.;\boldsymbol{\beta},\sigma\right)$ using MCMC methods. Due to invariance properties of the model, it in fact suffices to compare the observed standardized departure t_6^0 to the distribution of $t_6 = b_6/c_{6,6}^{1/2}\mathrm{e}^u$ obtained from the (reparameterized) null model with $\boldsymbol{\beta} = 0$ and $\sigma = 1$:

$$f\left(\mathbf{b},u\left|\mathbf{r}^0\right.\right)d\mathbf{b}\,du = c\prod_{i=1}^{m}h\left(\mathrm{e}^u r_i^0 + X_i\mathbf{b}\right)\mathrm{e}^{u(m-d)}d\mathbf{b}\,du \tag{8}$$

on $\mathbb{R}^{d+1}$. The previous reparameterization has been performed in order to avoid boundary problems when implementing MCMC methods.

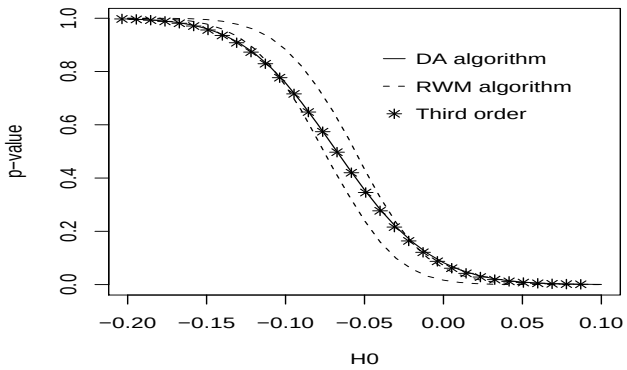
Sampling the latter density allows us to evaluate the p-value function $p(\beta_6)$ that gives the probability left to the data point under the hypothesis considered; this probability is computed as $p\left(\beta_6\right) = \{\#\, t_6\left(\mathbf{b},s\right) < t_6^0\left(\beta_6\right)\}/N$, where N is the size of the simulation and the numerator gives the number of instances $(\mathbf{b},s)$ yielding a value less than the observed $t_6^0(\beta_6)$.

5.1. Simulations

Exact p-values for this problem have been obtained in Bédard and Fraser [2009] using a directionally adaptive (DA) algorithm and a RWM sampler. The DA method consists in fixing the mode of the proposal density, and then adjusting its tails at each iteration to mimic the target density as closely as possible in a given direction. The potential of a (global) DA method is greater than that of a (local) RWM algorithm, but as expected its implementation requires more work: the mode of the target density needs to be identified and the Hessian needs to be evaluated at the mode. While the performance of the DA sampler was excellent for obtaining p-values in the current context, the RWM algorithm behaved erratically; the results obtained were also compared to the theoretical third-order p-values (see Figure 1).

This figure, reproduced from Bédard and Fraser [2009], shows two different runs of the RWM sampler (the dashed curves). The p-values for various hypotheses in a given curve were all obtained from a common RWM algorithm with $N = 4,000,000$. The instability of these curves suggests that the chain cannot freely move between the tails of the target density in various directions. A numerical exploration yielded the optimal proposal scaling $0.0001I_8$; the small proposal variance indicates that the i.i.d. assumption among the proposal components should be relaxed. Indeed, a bad convergence of the algorithm might be due to a few target components

FIGURE 1: Graph of p-values versus hypotheses H_0 obtained with the DA (solid) and RWM (dashed) algorithms, as well as third-order approximations (symbols).



exhibiting a smaller variance and restricting other components in the exploration of their state space, or to the unacknowledged correlation between certain pairs of target components.

5.2. Covariance estimation

Although the RWM algorithm should theoretically converge to the exact p-value, it is clear that in practice, it would take an unrealistic number of iterations to reach such a goal. Given the magnitudes of the proposed increments (based on the scaling $0.0001I_8$), including an estimate of the target covariance structure in the proposal distribution might reveal advantageous.

Unsurprisingly, using this same RWM sampler to estimate the covariance matrix is not an option due to the instability of the Markov chain. One could turn towards the adaptive Metropolis (AM) algorithm introduced by Haario et al. [2001], which updates the proposal covariance matrix at each iteration through a recursive formula. Relying on adaptive versions of algorithms with pools of proposals is however a delicate matter; although an adaptive version of the DR strategy has been introduced in Haario et al. [2006], no such result has been published, to our knowledge, about the MTM strategy. For the time being, we simply rely on the DA sampler (or the adaptive Metropolis algorithm) to obtain an estimate $\hat{\Sigma}$ of the target covariance matrix. This estimate will then be included in the proposal distribution of the RWM, MTM, MTM-HR, and DR-A algorithms in order to evaluate the performance of these methods.

As in Bédard and Fraser [2009], we are interested in testing $\beta_6 = -0.1, -0.01$, and 0.02; for each of these hypotheses, p-values are obtained using the DA and RWM samplers, as well as versions of the RWM, MTM, MTM-HR, and DR-A algorithms with proposal covariance proportional to $\hat{\Sigma}$. For local methods, a Gaussian proposal distribution is tuned to attain acceptance rates that are close to the AOARs available in the literature. Since an estimate of the covariance matrix is included in proposal distributions, we expect the theoretical AOARs to yield algorithms that are close to optimality, depite the violation of the i.i.d. assumption among the target components (see Roberts and Rosenthal [2001]). This assessment has been validated by a numerical optimality study that is not presented here.

Using the target density in (8), we generate a sample of size $N = 4,004,950$ for b_6, and discard the first 5,000 values as burn-in. The remaining values are divided into 4,000 batches

TABLE 5: p-values for testing various hypotheses for β_6 using various MCMC algorithms. The frequentist third-order p-values are included for comparison.

	$\beta_6 = -0.1$	$\beta_6 = -0.01$	$\beta_6 = 0.02$
Third-order	.75283	.10936	.03646
DA	.75697	.11693	.03730
(MC-MSE)	$(.0^3 4639)$	$(.0^3 266)$	$(.0^4 7498)$
RWM	.85794	.06137	.01116
(MC-MSE)	$(.0^1 957)$	$(.0^1 371)$	$(.0^2 135)$
RWM with covariance	.75650	.11699	.03787
(MC-MSE)	$(.0^2 367)$	$(.0^2 213)$	$(.0^3 592)$
MTM with covariance	.75792	.11714	.03747
(MC-MSE)	$(.0^2 222)$	$(.0^2 128)$	$(.0^3 371)$
MTM-HR with covariance	.75761	.11707	.03750
(MC-MSE)	$(.0^2 180)$	$(.0^2 102)$	$(.0^3 285)$
DR-A with covariance	.75687	.11637	.03794
(MC-MSE)	$(.0^2 183)$	$(.0^2 102)$	$(.0^3 287)$

each containing 1,000 values; the p-values are computed using the first 950 values in a batch, and the last 50 values are discarded to reduce correlation between batches. From the resulting vector of 4,000 p-values, the sample mean and MC-MSE in (1) are obtained for each sampler considered. The exact p-value required in (1) is approximated by the corresponding third-order p-value. The results of these simulations are recorded in Table 5. In the present context, it is clear that the RWM algorithm stands out as extremely poor; the other methods all yield good results.

Table 6 provides information about the efficiency and computational time of the various methods. The DA algorithm yields the best results, even when accounting for the running time. This sampler, although applicable in a general context, has been designed for computing p-values from smooth, unimodal target densities; it thus pays a particular attention to the tails of the target distribution.

A covariance estimate seems necessary in the current context to obtain good results with local algorithms. Among local methods, the MTM-HR and DR-A methods again yield the best results, with the DR-A our favourite due to its reduced computational intensity. In this case, evaluating the target density at various points along a given axis is not necessarily cheaper than evaluation at independent points. However, generating a pool of candidates has a lesser impact computationally than dealing with the covariance matrix at every iteration. Therefore, rather than implementing the plain RWM algorithm including $\hat{\Sigma}$, one might as well use a MTM-HR or DR-A version, which increase the net AQV by about 50% and 65% respectively.

The MTM method with a covariance estimate is not competitive, with a net AQV slightly below that of the RWM sampler with covariance. Why is there such a large difference between the times required to run the MTM and MTM-HR (or DR-A) algorithms? When relying on a covariance estimate, proposed increments are generated from a multivariate Gaussian distribution with correlated components, which is more intensive computationally than generating increments

TABLE 6: AQVs, computational times (in seconds), and net AQVs using different algorithms. The proposal scalings selected and acceptance rates obtained are also included.

	σ^2	Acc. rate	AQV	Time	Net AQV
DA		.717	1287.4	1074	360.81
RWM	.015	.243	$.0^3388$	301	$.0^3388$
RWM - covariance	.800	.230	138.7	1137	36.72
MTM - covariance	.970	.321	226.0	2411	28.21
MTM-HR - covariance	.800	.448	274.1	1478	55.82
DR-A - covariance	.800	.447[a]	273.9	1334	61.80

[a] The first- and second-level acceptance rates respectively are and 0.229 and 0.218.

from a distribution with independent components. In the MTM sampler, 3 such increments are generated at every iteration, against only one in the MTM-HR and DR-A algorithms. In this case, although the Student target does not directly allow for a bargain in computing the density at points in a common direction, the MTM-HR and DR-A algorithms remain the best options.

6. HIGH-DIMENSIONAL POSTERIOR DENSITY

As in Roberts and Rosenthal [2009], consider the following statistical model:

$$\begin{array}{ccccc} & & \mu & & \\ & \swarrow & \downarrow & \searrow & \\ \theta_1 & \ldots & \ldots & \theta_d & \qquad \theta_i \sim \text{Cauchy}(\mu, A), \;\; (1 \le i \le d) \\ \downarrow & & \downarrow & \downarrow & \\ Y_{11}, \ldots, Y_{1r_1} & & Y_{d1}, \ldots, Y_{dr_d} & & \qquad Y_{ij} \sim \mathcal{N}(\theta_i, V), \;\; (1 \le j \le r_i) \end{array}$$

with priors $\mu \sim \mathcal{N}(0,1)$, $A \sim \Gamma^{-1}(1,1)$, and $V \sim \Gamma^{-1}(1,1)$. A Cauchy(m,s) is a translated and scaled Cauchy distribution with density proportional to $[1+((x-m)/s)^2]^{-1}$, $\Gamma^{-1}(a,b)$ is the inverse gamma distribution with density proportional to $e^{-b/x}x^{-(a+1)}$, and $\{Y_{ij}\}$ are observed data. This model leads to a $(d+3)$-dimensional posterior density $\pi(A, V, \mu, \theta_1, \ldots, \theta_d | \{Y_{ij}\})$.

We fix $d = 210$ and let r_i vary between 11 and 30. The posterior density is too complex for analytic computation, and numerical integration must be ruled out due to the high-dimensionality of the problem. This distribution is best sampled with a Metropolis-within-Gibbs algorithm (a classical Gibbs sampler cannot be used, as the Cauchy distribution destroys conjugacy). In fact, we propose to sample five groups of variables in turn: $\mu, A, V, \boldsymbol{\theta}_L, \boldsymbol{\theta}_U$. In other words, μ, A, V shall be updated individually while the θ_is will be splitted into two distinct groups: $L = \{i : 11 \le r_i \le 20\}$ and $U = \{i : 21 \le r_i \le 30\}$. Using $(\boldsymbol{\theta}_L, \boldsymbol{\theta}_U)$ allows more precise estimates (a common proposal variance for updating $\boldsymbol{\theta}$ would result in a chain that moves very slowly due to the discrepancy in variances when $r_i = 11$ or $r_i = 30$), while retaining a large enough dimensionality to rely on optimal scaling results.

We estimate two parameters, $(\theta_{48}, \theta_{172}) = (2.644, 17.564)$ using the four samplers initialized as follows: $\mu = 0$, $A = 1$, $\theta_i = r_i^{-1}\sum_{j=1}^{r_i} Y_{ij}$ $(1 \le i \le d)$, and $V = d^{-1}\sum_{i=1}^{d}\sum_{j=1}^{r_i}(Y_{ij} - \theta_i)^2/(r_i - 1)$. The simulations are as before: $N = 4,004,950$

TABLE 7: Comparison of AQVs, computational times (in seconds) and net AQVs using different algorithms. Estimates for $(\theta_{48}, \theta_{172})$ along with MC-MSEs are also provided.

	Acceptance rate		AQV	Time	Net AQV	θ_{48}	θ_{172}
	$\boldsymbol{\theta}_L$	$\boldsymbol{\theta}_U$				$(r_i = 15)$	$(r_i = 28)$
True value						2.644	17.564
RWM	.220	.251	.113	2478	.113	2.635	17.561
(MC-MSE)						(.0137)	(.00845)
MTM	.335	.338	.191	7591	.062	2.634	17.560
(MC-MSE)						(.0085)	(.00434)
MTM-HR	.440	.474	.227	4768	.118	2.635	14.560
(MC-MSE)						(.0075)	(.00434)
DR-A	.430[a]	.484[b]	.223	4810	.115	2.634	17.560
(MC-MSE)						(.0074)	(.00431)

[a]The first- and second-level acceptance rates respectively are 0.218 and 0.212 for updating $\boldsymbol{\theta}_L$.
[b]The first- and second-level acceptance rates respectively are 0.247 and 0.237 for updating $\boldsymbol{\theta}_U$.

iterations are performed, a burn-in period of 5,000 iterations is applied, and we consider $\tau = 4,000$ batches containing $\eta = 950$ values (50 sample values are discarded between each batch). From the resulting vector of 4,000 batch means, the sample means $(\overline{\theta}_{48}, \overline{\theta}_{171})$ and the related MC-MSEs in (1) are obtained for each sampler. Gaussian proposal distributions are tuned to attain acceptance rates that are close to the theoretical AOARs. Due to the independence assumption between the θ_is and high-dimensional context, this yields optimally mixing chains.

Due to the presence of the Cauchy distribution, there is no bargain in evaluating the target density at various points along a common axis. Although the MTM-HR and DR-A should theoretically lead to higher AQV measurements, the extra computational effort required may offset this advantage.

Table 7 provides estimates for $(\theta_{48}, \theta_{172})$, along with their MC-MSEs. The values obtained are exactly as prescribed by the theoretical results. AQVs, acceptance rates, and computational times are also included. As there is no available simplification for evaluating the target density at various points, we would expect the MTM-HR and DR-A to be at a disadvantage compared to the RWM. It is interesting to note that it only takes approximately twice as long to run a MTM-HR or DR-A compared to a RWM. However, AQVs and MC-MSEs are improved by a factor of two in both cases, hence no difference in net AQV is observed compared to the RWM algorithm. MTM strategy is again too intensive computationally to be worth implementing.

7. A BIMODAL EXAMPLE

Consider a multivariate mixture of two normals with non-diagonal covariance structures :

$$\frac{1}{3}\mathcal{N}_{20}(\boldsymbol{\mu}_1, \Sigma_1) + \frac{2}{3}\mathcal{N}_{20}(\boldsymbol{\mu}_2, \Sigma_2) \,,$$

TABLE 8: Proposal variances, acceptance rates, AQVs, and computational times for the four samplers. The sample mean of X_1 and its MC-MSE are also provided.

	σ^2	Acc. rate	AQV	Time	Net AQV	$\overline{x}_1$	MC-MSE
RWM	0.85	0.224	2.920	1550	2.920	6.88	2.260
MTM	0.89	0.345	4.940	3760	2.036	7.07	2.005
MTM-HR	0.85	0.449	5.855	3475	2.612	6.96	2.123
DR-A	0.85	0.454[a]	5.930	3628	2.533	7.133	1.940

[a]The first- and second-level acceptance rates respectively are 0.229 and 0.225.

with $\boldsymbol{\mu}_1 = \mathbf{5}$ and $\boldsymbol{\mu}_2 = \mathbf{8}$. This is similar to the example of Section 4.1.2 in Casarin et al. [2013], to the exception that we rely on different covariance matrices Σ_1, Σ_2 : each covariance term is uniformly distributed in $[-0.1, 0.1]$, while the variance terms each satisfy $2 + \mathcal{U}[-0.1, 0.1]$.

We wish to estimate $\mathbb{E}[X_1]$ using the four samplers initialized at $\mathbf{X}[0] = \mathbf{5}$. To obtain relatively stable estimates, $N = 20,024,250$ iterations are performed, a burn-in period of 5,000 iterations is discarded, and we consider $\tau = 4,000$ batches containing $\eta = 950$ values (50 sample values are discarded between batches). From the resulting vector of 4,000 batch means, the sample mean and MC-MSE are obtained for each sampler. Gaussian proposal distributions are tuned to attain acceptance rates that are close to AOARs. Due to the "weak" covariance structure, this yields almost optimal chains.

Obviously, we cannot expect multiple-try and delayed rejection strategies to succeed where the plain RWM fails; in other words, these samplers are single-chain methods and thus two modes separated by a low-density region should preferably be sampled using multiple-chain methods such as in Casarin et al. [2013], or population-based methods such as in Jasra et al. [2007]. We could even possibly consider including the MTM-HR or DR-A in a multiple-chain method. For the target distribution introduced above, for which a RWM sampler succeeds in eventually visiting both modes, the results obtained appear in Table 8.

In terms of evaluating the target density at various points and computing the acceptance probability, the RWM method is computationally quite efficient. Although there is an efficiency gain resulting from the evaluation of the target density at several points along a common axis, the RWM remains significantly less demanding computationally than the other samplers. The target density is so easy to evaluate that the whole second-level acceptance rate in the delayed rejection method, for instance, is more expensive computationally than the evaluation of the target density itself. Although the AQVs of the DR-A and MTM-HR almost double compared to the RWM, the extra time required to run these samplers does not seem worth the effort. Furthermore, it is impossible to affirm, based on the estimates obtained, that one method is more accurate than the others. The DR-A has the largest bias, but the smallest MC-MSE. The MTM-HR has the smallest bias but, in spite of the large number of iterations performed, this ranking may vary from one run to another. The results are inconclusive, and we cannot affirm that algorithms with pools of proposals should be favored over the RWM algorithm in this multimodal setting.

If one was to include a covariance estimate in the proposal distribution, efficiency results would change dramatically. Generating a candidate would become much more demanding than the extra steps required by the MTM-HR or DR-A samplers. Since only one random vector per iteration is generated with these methods, we would obtain running times proportional to those in Section 5.

8. DISCUSSION

We described and compared four local RWM algorithms; three of these methods require pools of proposals for their implementation. The efficiency of these samplers is theoretically advantageous compared to the usual RWM algorithm. Of course, fancier methods usually imply a greater computational effort; it is however difficult to measure this quantity in general terms without being too conservative. As witnessed in this paper, the usual factor of 3 seems on the conservative side for accounting for the computational cost of MTM algorithms ($K = 2$), and accordingly is far too conservative for MTM-HR ($K = 2$) and DR-A strategies.

We tested these samplers in four different contexts: a Bayesian logistic regression model, a classical linear regression model, a high-dimensional Bayesian hierarchical model, and a bimodal target density. The first two examples were based on datasets found in the literature, while the last two dealt with simulated datasets. We used various measures of efficiency to assess the performance of the methods: the accuracy of the estimates (through their simulation variance or Monte Carlo mean squared errors), the average quadratic variations of the Markov chains, the running times of the algorithms, as well as the acceptance rates produced by the samplers.

In brief we have found that the DR-A algorithm performs at least as well, and in some cases much better, than the RWM method, with net efficiency gains going up to 65%. The theoretical superiority of the DR-A in terms of AQV is not surprising, as AQV is directly related to first-order autocorrelations of the chain, which in turn are related to Peskun ordering (see Mira [2001b]). Although the theoretical behaviors of the MTM-HR and DR-A algorithms are similar, the latter seems to outperform the former in many situations. The fact that the DR-A sampler does not generate two candidates at every iteration, but only upon the rejection of the first candidate, seems to have a relatively important impact on the computational cost of the method. The version of the MTM sampler considered here is not competitive, sometimes even when ignoring the extra computational effort required. The motivating example of Section 2 corroborates these rankings: the DR-A sampler only takes 30% longer to run than the RWM, leading to a net efficiency gain of 50%. In comparison, the net gain from using the MTM-HR is 10%, while the MTM suffers a net efficiency loss of 34%.

The analyses in this paper have been performed based on a traditional programming approach. In implementing a multiple-try strategy, one could however take advantage of the availability of multiple cores by parallelizing the algorithm. This could potentially have an impact on the efficiency balance, particularly for the MTM and MTM-HR samplers. The examples considered in the previous sections were also simulated in parallel on a system comprised of 4 cores, using the package `parallel` in `R`. In these particular cases, parallelizing the samplers results in running times significantly longer than under a linear programming approach. In fact, the computations performed in parallel are not extremely intensive and the time gained by parallelizing is largely offset by the time needed for the child processes to compile results and communicate with the parent process. When dealing with extremely demanding target distributions, a parallel approach may of course be worthwhile.

It is not clear whether or not the DR-A and MTM-HR methods could be favored over the RWM algorithm in general. However, they certainly cannot be discarded based on their computational intensity, as they often seem to constitute an interesting and efficient alternative to the RWM. On the one hand, we do not expect the DR-A and MTM-HR algorithms to consistently outperform the RWM algorithm. When the target density is extremely cheap to evaluate, the RWM method will likely remain the best available option. On the other hand, situations in which the DR-A and MTM-HR samplers outperform the RWM algorithm are certainly not exceptions. From the examples of Sections 5 and 7, we deduce that algorithms with pools of proposals do not solve fundamental problems faced with the RWM algorithm (badly designed proposal covariance matrix, for instance). However, in several situations, they certainly seem to offer a significant im-

provement on performance.

ACKNOWLEDGEMENTS

We are grateful to the Associate Editor and anonymous referees for constructive comments. This work has been supported by the Fonds québécois de la recherche sur la nature et les technologies.

BIBLIOGRAPHY

M. Bédard, R. Douc, E. Moulines, Scaling analysis of delayed rejection MCMC methods, 2010. To appear in Methodology and Computing in Applied Probability.

M. Bédard, R. Douc, E. Moulines, Scaling analysis of multiple-try MCMC methods, Stochastic Processes and their Applications 122 (2012) 758–786.

M. Bédard, D. Fraser, On a directionally adjusted Metropolis-Hastings algorithm, International Journal of Statistical Sciences 9 (2009) 33–57.

M. Bédard, J. Rosenthal, Optimal scaling of Metropolis algorithms: heading toward general target distributions, Canadian Journal of Statistics 36 (2008) 483–503.

A. Brazzale, A. Davison, N. Reid, Applied Asymptotics: Case Studies in Small-Sample Statistics, volume 23, Cambridge University Press, 2007.

R. Casarin, R. Craiu, F. Leisen, Interacting multiple try algorithms with different proposal distributions, Statistics and Computing 23 (2013) 185–200.

D. Cox, E. Snell, Applied Statistics: Principles and Examples, volume 2, Chapman & Hall/CRC, 1981.

R. Craiu, C. Lemieux, Acceleration of the multiple-try Metropolis algorithm using antithetic and stratified sampling, Statistics and Computing 17 (2007) 109–120.

P. Dellaportas, J. Forster, I. Ntzoufras, On Bayesian model and variable selection using MCMC, Statistics and Computing 12 (2002) 27–36.

D. Frenkel, G. Mooij, B. Smit, Novel scheme to study structural and thermal properties of continuously deformable molecules, Journal of Physics: Condensed Matter 4 (1992) 3053–3076.

D. Frenkel, B. Smit, Understanding molecular simulations, Academic Press, 1996.

P. Green, A. Mira, Delayed rejection in reversible jump Metropolis-Hastings, Biometrika 88 (2001) 1035–1053.

H. Haario, M. Laine, A. Mira, E. Saksman, DRAM: efficient adaptive MCMC, Statistics and Computing 16 (2006) 339–354.

H. Haario, E. Saksman, J. Tamminen, An adaptive Metropolis algorithm, Bernoulli (2001) 223–242.

W.K. Hastings, Monte Carlo sampling methods using Markov chains and their application, Biometrika 57 (1970) 97–109.

A. Jasra, D. Stephens, C. Holmes, On population-based simulation for static inference, Statistics and Computing 17 (2007) 263–279.

J.S. Liu, F. Liang, W.H. Wong, The multiple-try method and local optimization in Metropolis sampling, Journal of the American Statistical Association 95 (2000) 121–134.

J. Mattingly, N. Pillai, A. Stuart, Diffusion limits of random walk Metropolis in high dimensions, Annals of Applied Probability 22 (2012) 881–930.

A. Mira, On Metropolis-Hastings algorithms with delayed rejection, Metron LIX (2001a) 231–241.

A. Mira, Ordering and improving the performance of Monte Carlo Markov chains, Statistical Science 16 (2001b) 340–350.

A. Mira, D. Sargent, A new strategy for speeding Markov chain Monte Carlo algorithms, Statistical Methods and Applications. 12 (2003) 49–60.

C. Pasarica, A. Gelman, Adaptively scaling the Metropolis algorithm using expected squared jumped distance, Statistica Sinica 20 (2010) 343–364.

C. Robert, K. Mengersen, Iid sampling with self-avoiding particle filters: the pinball sampler, Bayesian Statistics 7 (2003) 277–292.

G. Roberts, J. Rosenthal, Optimal scaling for various Metropolis-Hastings algorithms, Statistical Science 16 (2001) 351–367.

G. Roberts, J. Rosenthal, Examples of adaptive MCMC, Journal of Computational and Graphical Statistics 18 (2009) 349–367.

G.O. Roberts, A. Gelman, W.R. Gilks, Weak convergence and optimal scaling of random walk Metropolis algorithms, Annals of Applied Probability 7 (1997) 110–120.

Z. Tan, Monte Carlo integration with acceptance-rejection, Journal of Computational and Graphical Statistics 15 (2006) 735–752.

L. Tierney, A. Mira, Some adaptive Monte Carlo methods for Bayesian inference, Statistics in Medicine 18 (1999) 2507–2515.

Received 9 July 2009
Accepted 8 July 2010