Scaling analysis of delayed rejection MCMC methods

Mylène Bédard[*], **Randal Douc**[†], **Eric Moulines**[†]

Département de mathématiques et de statistique, Université de Montréal, H3C 3J7, Canada, e-mail: bedard@dms.umontreal.ca

SAMOVAR, CNRS UMR 5157 - Institut Télécom/Télécom SudParis, 9 rue Charles Fourier, 91000 Evry, e-mail: randal.douc@it-sudparis.eu

LTCI, CNRS UMR 8151 - Institut Télécom /Télécom ParisTech, 46, rue Barrault, 75634 Paris Cedex 13, France, e-mail: eric.moulines@telecom-paristech.fr

Abstract: In this paper, we study the asymptotic efficiency of the delayed rejection strategy. In particular, the efficiency of the delayed rejection Metropolis-Hastings algorithm is compared to that of the regular Metropolis algorithm. To allow for a fair comparison, the study is carried under optimal mixing conditions for each of these algorithms. After introducing optimal scaling results for the delayed rejection (DR) algorithm, we outline the fact that the second proposal after the first rejection is discarded, with a probability tending to 1 as the dimension of the target density increases. To overcome this drawback, a modification of the delayed rejection algorithm is proposed, in which the direction of the different proposals is fixed once for all, and the Metropolis-Hastings accept-reject mechanism is used to select a proper scaling along the search direction. It is shown that this strategy significantly outperforms the original DR and Metropolis algorithms, especially when the dimension becomes large. We include numerical studies to validate these conclusions.

AMS 2000 subject classifications: Primary 60F05; secondary 65C40.
Keywords and phrases: Random walk Metropolis, weak convergence, diffusion, correlated proposals, multiple proposals.

1. The Delayed Rejection strategy

Markov chain Monte Carlo (MCMC) methods are used to produce samples from an arbitrary distribution π known up to a scaling factor; see Robert and Casella (2004). The technique consists in sampling a Markov chain $\{X_k, k \geq 0\}$ on a state space X admitting π as its unique invariant distribution.

Metropolis-Hastings algorithms are an important class of MCMC algorithms. These algorithms allow for a lot of flexibility, as they can be used to sample from virtually any probability distribution of interest. Specifically, let us make an abuse of notation by letting π denote not only the target distribution, but

[*] This work is supported by the National Sciences and Engineering Research Council of Canada.
[†]This work is supported by the Agence Nationale de la Recherche (ANR, 212, rue de Bercy 75012 Paris) through the 2009-2012 project Big MC

also the target density on a state space X with respect to some measure μ. To become part of a Markov chain currently at a state x, a state y must survive two stages: it must first be selected as the candidate from the proposal distribution, and then be retained as a suitable value for the Markov chain (through the acceptance function). We then define a proposal density $q(x;\cdot)$ on X with respect to the same measure μ as before. The Metropolis-Hastings algorithm amounts to choosing the candidate y with probability

$$\alpha(x;y) \triangleq 1 \wedge \frac{\pi(y)q(y;x)}{\pi(x)q(x;y)} \; . \tag{1}$$

If the candidate y is rejected, the chain remains at the current state x. The acceptance function of this algorithm has been chosen to satisfy the reversibility property of the chain with respect to π, that is

$$\pi(x)q(x;y)\alpha(x;y) = \pi(y)q(y;x)\alpha(y;x) \; , \quad \forall \; (x,y) \in \mathsf{X} \times \mathsf{X} \; . \tag{2}$$

This property ensures that π is the invariant distribution of the Markov chain.

The delayed rejection Metropolis algorithm introduced by Tierney and Mira (1999) and further developed in Mira (2001a) for fixed dimensional problems (and Green and Mira (2001) for trans-dimensional problems), aims at improving the proposal scheme of the Metropolis-Hastings algorithm by learning from past rejections (in a given iteration) to choose the proposal distribution. It allows, upon the rejection of a move, to retry further candidates before incrementing time; it thus generates several successive trial values per iteration. In one iteration of the k-step delayed rejection algorithm, we start by proposing a candidate for the next state of the Markov chain. Upon the rejection of this value, another candidate is generated from a proposal density possibly different from the first one. This is repeated until a move is accepted, or until k candidates have been rejected.

In recent years, the delayed rejection algorithm has been successfully applied to a number of very challenging simulation problems in different areas, such as volatility modeling in financial econometrics (Raggi (2005)), gravitational wave searches (Umstätter *et al.* (2004); Trias, Vecchio and Veitch (2009)), object recognition (Harkness and Green (2000)), and time-series analysis (Haario *et al.* (2006)). However, there has not been much research about the theoretical properties of these algorithms. Furthermore, it is not totally obvious how to select the different proposal distributions to improve maximally the performance of the method. In this paper, we focus on the two-step random walk delayed rejection algorithm mentioned in Green and Mira (2001); we derive asymptotic optimal scaling results allowing us to determine the proposal scalings for which this algorithm shall mix optimally.

The first theoretical results about the optimal scaling issue for MCMC algorithms have been obtained by Roberts, Gelman and Gilks (1997) in the case of the Metropolis algorithm with a Gaussian increment density (see also Gelfand and Mitter (1991) for related weak convergence results of Metropolis algorithms). In their article, the authors focused on finding the best scale for the

algorithm when sampling from T-dimensional product target densities (with T large). Although the target densities considered form a rather restrictive class, the results obtained have been illuminating for both researchers and practitioners. They opened the door to several generalizations and nowadays, researchers are starting to consider increasingly realistic target models, some of them including correlated components; see Roberts and Rosenthal (2001), Bédard (2007), Bédard and Rosenthal (2008), Beskos, Roberts and Stuart (2009), Mattingly, Pillai and Stuart (2012).

The optimal scaling results published in the literature have mainly been restricted to the Metropolis algorithm, the Metropolis-adjusted Langevin algorithm (MALA) and, very recently, to the so-called Hybrid Monte Carlo algorithms; see Mattingly, Pillai and Stuart (2012) and the references therein. One of the goals of this work is to demonstrate that the scaling limits can be useful in the context of delayed rejection Metropolis algorithms, in which the successive candidates are obtained from random walks, possibly with different scales. We first consider the case where the successive candidates are independent. The scaling analysis shows that such an algorithm is prone to be inefficient in large dimension because, rather unexpectedly, the probability of accepting the second candidate once the first one has been rejected vanishes as the dimension goes to infinity. This is due to the specific form of the acceptance ratio, which is designed to preserve the reversibility of the kernel with respect to the stationary distribution π. This suggests generating a candidate which is correlated with the first one. It is in particular shown that proposing a second candidate antithetically can be very successful; this conclusion is supported by our computer experiments.

The paper is organized as followed. Section 2 introduces the two-step random walk delayed rejection algorithm of Green and Mira (2001); optimal scaling results for this algorithm are given in Theorem 1 (proof is postponed to Appendix B). Section 3 introduces the two-step random walk delayed rejection algorithm with common proposal; optimal scaling results are presented in Theorem 4 (proof is postponed to Appendix C). We conclude with simulation studies that validate our findings in Section 4 and with a discussion in Section 5.

2. The two-step random walk delayed rejection algorithm

In this section, we consider the original two-step delayed rejection algorithm. Denoting the current state of the chain by $X = x$, one iteration of this algorithm is defined as follows

Algorithm 1 (Delayed Rejection Algorithm)**.**
(a) Draw a candidate Y^1 from a proposal distribution $q^1(x;\cdot)$;
(b) Accept this proposal with probability $\alpha_1(x, Y^1)$, where

$$\alpha_1(x; y^1) \triangleq 1 \wedge \frac{\pi(y^1)q^1(y^1; x)}{\pi(x)q^1(x; y^1)} \,. \tag{3}$$

(c) Upon the rejection of the previous candidate, draw a second candidate Y^2 *from the transition kernel* $q^2(x, Y^1; \cdot)$ *and accept it with probability* $\alpha_2(x, Y^1; Y^2)$*, where*

$$\alpha_2(x, y^1; y^2) = 1 \wedge \frac{\pi\left(y^2\right) q^1(y^2; y^1)\left[1 - \alpha_1(y^2; y^1)\right] q^2\left(y^2, y^1; x\right)}{\pi\left(x\right) q^1\left(x; y^1\right)\left[1 - \alpha_1\left(x; y^1\right)\right] q^2\left(x, y^1; y^2\right)} \,. \tag{4}$$

The rejection of the first candidate suggests that the proposal distribution from which it has been generated is not very well suited to the target distribution. Therefore, it might be worth freezing time and proposing another value y^2 for the chain (possibly depending on the first move); this candidate is accepted with probability $\alpha_2(x, y^1; y^2)$ and is rejected otherwise. Upon the rejection of both proposed values y^1 and y^2, we repeat the current value x in our sample, increment time and start over again with q^1. The acceptance probability α_2 is chosen such that the generated Markov chain is reversible with respect to the target distribution π.

The two-step method outlined illustrates the basic idea of the delayed rejection Metropolis algorithm, but can of course be extended to the general k-step delayed rejection algorithm. This is achieved by adjusting the acceptance probability of every additional layer such as to preserve the reversibility property of the chain.

Among possible strategies of learning upon a first stage rejection, Tierney and Mira (1999) suggest choosing an initial independence proposal density $q^1(x; y^1) = q^1(y^1)$ that is believed to be well-fitted to the target distribution, and to use a second stage random walk proposal $q^2(x, y^1; y^2) = q^2(y^2 - x)$. If q^1 is a good approximation of π, the first candidate will be most often accepted. In the eventuality where q^1 is not a good proposal, then the random walk component will protect the chain from the bad mixing of an independence MCMC with poor proposal distribution. Tierney and Mira (1999) also discuss an approach based on the model-trust region method of optimization. Although potentially effective, these two strategies remain difficult to study from a theoretical perspective.

An appealing specific case of the delayed rejection methods later introduced in Green and Mira (2001) is the two-step random walk delayed rejection algorithm. This method consists in proposing a candidate y^1 in the first stage of the iteration by using a symmetrical increment density, i.e. $q^1(x; y^1) = q^1(y^1 - x)$ with $q^1(-z) = q^1(z)$. If the candidate is accepted, which happens according to the usual acceptance probability

$$\alpha_1(x; y^1) = 1 \wedge \pi(y^1)/\pi(x) \,, \tag{5}$$

then we move to the next iteration. In the case where y^1 is rejected, a second candidate y^2 is drawn from a different symmetrical increment density $q^2(x, y^1; y^2) = q^2(y^2 - x)$ with $q^2(-z) = q^2(z)$; the second candidate is then accepted with probability

$$\alpha_2(x, y^1; y^2) = 1 \wedge \frac{\pi\left(y^2\right) q^1\left(y^1 - y^2\right)\left[1 - \pi(y^1)/\pi(y^2)\right]_+}{\pi\left(x\right) q^1\left(y^1 - x\right)\left[1 - \pi(y^1)/\pi(x)\right]_+} \,, \tag{6}$$

where $a_+ = \max(a, 0)$.

In both stages, candidates are thus generated according to random walk components. One may, for example, choose to use two increment distributions from the same family (e.g. Gaussian) but with different scales, $q^i(y-x) = \sigma_i^{-1} q((y-x)/\sigma_i)$, $i = 1, 2$ with $q(z) = q(-z)$. An intuitive approach would consist in being more aggressive with the first candidate and thus attempting to perform a large move; if this move is rejected, then a second candidate generated from a distribution with a smaller scale could be proposed. Under this setting, the algorithm gets to choose the most appropriate scale for the proposal distribution *locally*.

Following the above description of the two-step random walk delayed rejection (RWDR) algorithm, there would thus be two main questions of interest: (i) does the RWDR algorithm compare favorably to the usual Metropolis algorithm, and (ii) what are the proposal scales leading to an optimal mixing of this algorithm over its state space? In order to answer the first question, we obviously need answers to the second one. Indeed, if we are to put the RWDR methods against the Metropolis algorithm, it would only be fair to do so under optimal settings for both algorithms. The remaining of this section thus focuses on studying this optimal scaling issue for the two-step random walk delayed rejection algorithm.

To facilitate the comparison to the Metropolis algorithm, we shall work under assumptions that are similar to those specified in Roberts, Gelman and Gilks (1997). The first of these assumptions describes the form of the target density:

(**A1**) The target density is formed of $T+1$ i.i.d. components:

$$\pi(\mathbf{x}) = \prod_{t=0}^{T} f(x_t) \ . \tag{7}$$

The one-dimensional probability density function f is a positive twice continuously differentiable function, $[\ln f]''$ is bounded Lipschitz and

$$\int f(x) |[\ln f]'(x)|^4 \mathrm{d}x < \infty \ .$$

Solving the optimal scaling problem in finite-dimensional settings is a very difficult task, and has been achieved for a specific class of target distributions only. Optimal scaling results are more generally obtained by considering the high-dimensional distributions that arise from letting $T \to \infty$. Under this framework, adjustments of the proposal distributions become necessary and are now discussed.

We denote by $\left(\mathbf{X}[n] \triangleq (X_t[n])_{t=0}^{T}, n \in \mathbb{N}\right)$ the sequence of Markov chains on $(\mathbb{R}^{T+1}, T \in \mathbb{N})$ defined by the RWDR algorithm with target distribution π given in (7).

Define $\mathcal{F} = (\mathcal{F}_n, n \geq 0)$, the natural filtration of the Markov chain $\mathbf{X}$, i.e. for any $n \geq 0$,

$$\mathcal{F}_n \triangleq \sigma(\mathbf{X}[m], m = 0, \ldots, n) \ . \tag{8}$$

The assumptions above give rise to a particular case of the classical delayed rejection sampling, which is presented in more detail:

Algorithm 2 (Two-step RWDR algorithm with independent proposals)**.**
(a) Given the current state $\mathbf{X}[n]$ *of the Markov chain at time* n*, two proposals*

$$\left(\mathbf{Y}^i[n+1]\right)_{i=1,2} \triangleq \left(Y_t^i[n+1], 0 \le t \le T\right)_{i=1,2} ,$$

are generated according to

$$Y_t^i[n+1] = X_t[n] + T^{-1/2} U_t^i[n+1] , \quad 0 \le t \le T, \quad i = 1,2 \tag{9}$$

where

(a) for any $t \in \{0, \dots, T\}$, $\begin{pmatrix} U_t^1[n+1] \\ U_t^2[n+1] \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \ell_1^2 & 0 \\ 0 & \ell_2^2 \end{bmatrix}\right)$,

(b) The $T+1$ *random vectors* $\{(U_t^1[n+1], U_t^2[n+1])\}_{t=0}^T$ *are conditionally independent given* $\mathcal{F}_n$*, where* $\mathcal{F}_n$ *is defined in* (8).

(b) The first proposal $\mathbf{Y}^1[n+1]$ *is accepted with probability* $\alpha_1(\mathbf{X}[n]; \mathbf{Y}^1[n+1])$ *where*

$$\alpha_1(x; y^1) = 1 \wedge \frac{\pi(y^1)}{\pi(x)} . \tag{10}$$

(c) Upon rejection of the first candidate $\mathbf{Y}^1[n+1]$*, propose the second candidate* $\mathbf{Y}^2[n+1]$ *and accept it with probability:*

$$\alpha_2(\mathbf{X}[n], \mathbf{Y}^1[n+1]; \mathbf{Y}^2[n+1])$$

where

$$\alpha_2(x, y^1; y^2) = 1 \wedge \frac{\pi\left(y^2\right) q^1\left(y^1 - y^2\right) \left[1 - \pi(y^1)/\pi(y^2)\right]_+}{\pi\left(x\right) q^1\left(y^1 - x\right) \left[1 - \pi(y^1)/\pi(x)\right]_+} , \tag{11}$$

and q^1 *is the density of a centered Gaussian random vector of dimension* $T+1$ *with covariance matrix* $\ell_1^2 \mathrm{I}_{T+1}/T$ *and* I_{T+1} *is the* $(T+1)$*-dimensional identity matrix. Otherwise, reject it.*

Remark 1. *In Step a, both candidates are generated simultaneously whereas in practice, a given iteration might only require the simulation of the first candidate. This is convenient here to define the algorithm, but is of course not required in its implementation.*

Before going further, it is worth outlining the form of the covariance matrix of the proposal distribution stated above. The proposal variances ℓ_i^2/T, $i = 1, 2$ are decreasing functions of the dimension; this means that as T increases, the algorithm generates candidates that are increasingly conservative. This rescaling of the space keeps the acceptance rates from going to 0. To match this space adjustment, it is also necessary to speed time up. Formally, denote by ζ_T the

projection on the first coordinate, that is $\zeta_T : \mathbb{R}^{T+1} \to \mathbb{R}$ such that $\zeta_T(\mathbf{x}) = x_0$. Consider the progressive cadlag process $W_T \triangleq (W_T[s], s \in \mathbb{R}^+)$ defined by

$$s \mapsto \; W_T[s] = \zeta_T \left[\mathbf{X} \left(\lfloor Ts \rfloor \right) \right] \; . \tag{12}$$

It has been realized over the years that asymptotically, $s \mapsto W_T[s]$ most often behaves according to Langevin diffusion processes as T goes to infinity, and that these limiting processes only differ in the form of their speed measure. In order to lighten general diffusive limit proofs, Bédard, Douc and Moulines (2012) have recently introduced some conditions establishing weak convergence (in the Skorokhod topology) of a general class of MCMC algorithms with multiple proposals to the appropriate Langevin diffusion process. These conditions are recalled in Appendix A.

Before stating the first theorem, let us introduce a few quantities of interest. The conditional probability of accepting the first move is given by $\bar{\alpha}_1(\mathbf{X})$, where

$$\bar{\alpha}_1(\mathbf{x}) = \mathbb{E}\left[A_1\left(L_{0,T}(\mathbf{x}, \mathbf{U}^1) \right) \right] \tag{13}$$

with

$$A_1(u) = 1 \wedge \mathrm{e}^u \; , \tag{14}$$

and, for $s \in \{0, \ldots, T\}$, $L_{s,T}(\mathbf{x}, \mathbf{u})$ is the log-likelihood ratio

$$L_{s,T}(\mathbf{x}, \mathbf{u}) = \sum_{t=s}^{T} \left\{ \ln f(x_t + T^{-1/2} u_t) - \ln f(x_t) \right\} \; . \tag{15}$$

For $\kappa \in \mathbb{R}$, define

$$\tilde{\alpha}_1(\kappa) \triangleq \mathbb{E}\left[A_1(\kappa G - \kappa^2/2) \right] \; , \tag{16}$$

where G is a standard Gaussian variable, and let

$$\mathcal{I} \triangleq \int [\ln f]'(x) f(x) \mathrm{d}x \; . \tag{17}$$

Hereafter, weak convergence in the Skorokhod topology (see Billingsley (1999)) is denoted by $\Rightarrow$ and the standard Brownian process is denoted by $(B[s], s \in \mathbb{R}^+)$.

Theorem 1. *Assume (A1) and consider Algorithm 2. Suppose that the algorithm starts in stationarity, i.e.* $\mathbf{X}[0]$ *is distributed according to the target density* π.

Consider the first component of the rescaled process $\{W_T[t], t \geq 0\}$ *defined in* (12). *Then* $\{W_T[t], t \geq 0\} \Rightarrow \{W[t], t \geq 0\}$, *where* $W[0]$ *is distributed according to the density* f *in (A1) and* $\{W[t], t \geq 0\}$ *satisfies the Langevin stochastic differential equation (SDE)*

$$\mathrm{d}W[t] = \lambda_1(\ell_1)^{1/2} \mathrm{d}B[t] + \frac{1}{2} \lambda_1(\ell_1) \left[\ln f \right]' (W[t]) \mathrm{d}t \; ,$$

with

$$\lambda_1(\ell_1) = \ell_1^2 \tilde{\alpha}_1(\ell_1 \mathcal{I}^{1/2}) \; , \tag{18}$$

where $\tilde{\alpha}_1$ *and* $\mathcal{I}$ *are defined in* (16) *and* (17) *respectively.*

The proof is postponed to Appendix B. It is implicit from this theorem that when both proposals are independent, the probability of accepting the second proposal goes to zero as the dimension becomes large. This may be deduced from the expression for the speed of the limiting diffusion, which coincides with the result reported in Roberts, Gelman and Gilks (1997) for the Metropolis algorithm. Indeed, note that we may write

$$\tilde{\alpha}_1(\ell_1 \mathcal{I}^{1/2}) = 2\Phi\left(-\frac{\ell_1}{2}\sqrt{\mathcal{I}}\right) ,$$

where Φ is the cumulative distribution function of a standard normal random variable. The corresponding asymptotically optimal proposal variance and acceptance rate are provided in the following corollary.

Corollary 2. *In the setting of Theorem 1, the proposal scale ℓ_1 optimizing the speed measure of the diffusion is given by $\ell_1^\star = 2.39\mathcal{I}^{-1/2}$. The expected acceptance rate of the algorithm satisfies*

$$\tilde{\alpha}_1(\ell_1^\star \mathcal{I}^{1/2}) = 0.234 .$$

In Theorem 1, the dependence on ℓ_2 has vanished as $T \to \infty$; this is due to the form of the second-level proposal variance, ℓ_2^2/T. This scaling results in proposed increments that are too agressive for second-level candidates, so setting ℓ_2 to any value yields a second-level acceptance rate converging to 0.

Remark 2. *It might be of interest to use a different scale adjustment fo second-level proposal variances. By letting the first-level proposal variance be ℓ_1^2/T (as before) and the second-level proposal variance take the form ℓ_2^2/T^2 (instead of ℓ_2^2/T), we still obtain an algorithm that asymptotically behaves according to a Langevin diffusion process. For large T, the speed measure of the diffusion satisfies*

$$\lambda(\ell_1, \ell_2) = 2\ell_1^2\Phi\left(-\frac{\ell_1}{2}\mathcal{I}^{1/2}\right) + \frac{2\ell_2^2}{T}\Phi\left(-\frac{\ell_2}{2\ell_1}\right)\left\{1 - 2\Phi\left(-\frac{\ell_1}{2}\mathcal{I}^{1/2}\right)\right\} .$$

In order to maximize the efficiency gain from second-level candidates, the parameters ℓ_1, ℓ_2 should be chosen so as to maximize the speed measure $\lambda(\ell_1, \ell_2)$, leading to asymptotically optimal scales of $\ell_1^\star = 2.39\mathcal{I}^{-1/2}$ and $\ell_2^\star = 5.66\mathcal{I}^{-1/2}$. It should however be noticed that the impact of second-level candidates is tuned down by a factor of T^{-1}; this means that even if an optimal value for ℓ_2 can be found, the generated increments will be too small to have an impact on the overall mixing of the chain. The $1/T^2$ rate thus results in an efficiency gain coming from second-level candidates that converges to 0.

The random walk delayed rejection algorithm with independent proposals yields second-level candidates that are always rejected (when the second-level proposal variance is ℓ_2^2/T), or second-level increments that become unsignificant in high dimensions (when the second-level proposal variance is ℓ_2^2/T^2, or any rate smaller than $1/T$). The second step in the delayed rejection mechanism thus becomes useless in this limit.

3. The two-step random walk delayed rejection algorithm with common proposal

A major problem with the previous version of the delayed rejection algorithm lies in the fact that both tries are generated independently of each other. In other words, the information brought by the first rejection remains partly unused when it comes to proposing a second candidate. The rejection of the first candidate might mean that the proposed increment was either too aggressive in the direction considered, or even that attempting any move in this direction is unlikely to result in the acceptance of the candidate. To make the most of the information brought by the rejection of a first-level candidate, it is sensible to propose a second-level candidate that depends on the first one. We could thus have better chances of accepting a candidate if we were trying to move from the current value along the same direction with a more conservative scale, or simply in the opposite direction. We assume in the sequel that:

DR-Ca Conditionally to the current state $X = x$, Y^1 is sampled according to $q^1(x;\cdot)$. The second proposal is then set to $Y^2 = \Psi(x; Y^1)$ where Ψ is a measurable function from $\mathsf{X} \times \mathsf{X} \to \mathsf{X}$. The function Ψ is chosen such that Y^2 is conditionally distributed according to $q^2(x;\cdot)$ given the current state.

DR-Cb There exists a measurable function $\bar{\Psi} : \mathsf{X} \times \mathsf{X} \to \mathsf{X}$ such that $Y^1 = \bar{\Psi}(x, Y^2)$.

In words, both proposals are sampled from the same random element using different transformations. The only constraint is that, given x and Y^2, it is possible to compute Y^1 and vice versa. Starting from the idea that we wish to use a common random element to construct two candidates, the existence of functions Ψ and $\bar{\Psi}$ satisfying DR-Ca and DR-Cb is a mild restriction in practice. These functions are just a way to mathematically express the relationship between Y^1 and Y^2.

Supposing that the current state of the chain is $X = x$, one iteration of the delayed rejection Metropolis-Hastings algorithm with common random number is defined as follows.

Algorithm 3 (DR-Common)**.**
(a) Draw Y^1 according to $q^1(x;\cdot)$
(b) Accept the first move with probability $\alpha_1(x; Y^1)$, where α_1 is defined in (3).
(c) If the first move is rejected, compute $Y^2 = \Psi(x, Y^1)$ and accept this second proposal with probability $\bar{\alpha}_2(x; Y^2)$ where

$$\bar{\alpha}_2(x;y) = 1 \wedge \frac{\pi(y)\left[1-\alpha_1\left(y;\bar{\Psi}(y,x)\right)\right] q^2(y;x)}{\pi(x)\left[1-\alpha_1\left(x;\bar{\Psi}(x,y)\right)\right] q^2(x;y)}\,, \tag{19}$$

and reject otherwise.

Because the two proposals are now correlated, the acceptance ratio of the second candidate should be corrected to preserve the reversibility property of

the Markov chain. Since the dependence introduced does not affect the first candidate, the first-level acceptance ratio remains as before. It is worthwhile noting (and essential to the scaling results below) that the first-level proposal density q^1 does not appear in the acceptance ratio $\bar{\alpha}_2$.

For the general DR algorithm, the reversibility property of the chain with respect to π is satisfied if, for all $(x, y^1, y^2) \in \mathsf{X} \times \mathsf{X} \times \mathsf{X}$,

$$\begin{aligned}\pi(x)q^1(x;y^1)[1-\alpha_1(x;y^1)]q^2(x,y^1;y^2)\alpha_2(x,y^1;y^2) = \\ \pi(y^2)q^1(y^2;y^1)[1-\alpha_1(y^2;y^1)]q^2(y^2,y^1;x)\alpha_2(y^2,y^1;x)\ .\end{aligned}$$

The construction of Y^1, Y^2 from the current state x and the random element Z defines the functions Ψ and $\bar{\Psi}$ linking both candidates together. Since knowledge of x and Y^2 deterministically leads to $Y^1 = \bar{\Psi}(x, Y^2)$, then we may express the previous equation in terms of x and y^2 only :

$$\begin{aligned}\pi(x)[1-\alpha_1(x;\bar{\Psi}(x;y^2))]q^2(x;y^2)\alpha_2(x;y^2) = \\ \pi(y^2)[1-\alpha_1(y^2;\bar{\Psi}(y^2;x))]q^2(y^2;x)\alpha_2(y^2;x)\ , \quad \forall\ (x,y^2) \in \mathsf{X}\times\mathsf{X}\ .\end{aligned}$$

Note that, as in Green and Mira (2001), the return path from y^2 to x does not go through y^1. It is now easily shown that, with an acceptance ratio as in (19), the kernel is reversible with respect to π:

Theorem 3. *Under assumptions DR-Ca and DR-Cb, the DR-Common algorithm described above satisfies the detailed balance condition and hence induces a reversible chain with stationary distribution π.*

The second-level acceptance ratio thus guarantees reversibility regardless of the functions Ψ and $\bar{\Psi}$, provided of course that these are the mathematical functions linking Y^1 and Y^2 together.

This algorithm is very simple to implement with random walk increments. In this case, the first candidate is set to $Y^1 = x + \ell_1 Z$, where Z is a symmetric random vector and ℓ_1 is a scale factor. Taking

$$\Psi(x,y) = x + \ell_2\ell_1^{-1}\,(y-x)\ , \tag{20}$$

the second candidate is therefore given by $Y^2 = \Psi(x, Y^1) = x + \ell_2 Z$ and the proposal kernel q^2 is also symmetric. In such a case, $Y^1 = \bar{\Psi}(x, Y^2)$ where

$$\bar{\Psi}(x,y) = x + \ell_1\ell_2^{-1}\,(y-x)\ . \tag{21}$$

The scale ℓ_2 should of course be different from ℓ_1. We might, for instance, choose $0 \le \ell_2 \le \ell_1$ if we are willing to try a second step heading in the same direction but with a smaller scale. Another possibility consists in setting $\ell_2 = -\ell_1$, in which case the two candidates are antithetic. As we will see below, antithetic proposals should generally be preferred. In all cases, the second-level acceptance ratio might be expressed as

$$\bar{\alpha}_2(x;y) = 1 \wedge \frac{\pi(y)\left[1-\pi\left[\bar{\Psi}(y,x)\right]/\pi(y)\right]_+}{\pi(x)\left[1-\pi\left[\bar{\Psi}(x,y)\right]/\pi(x)\right]_+}\ . \tag{22}$$

As was done for the random walk delayed rejection algorithm with independent candidates, we would be interested in comparing the performance of this new version to the performance of the Metropolis algorithm. Since we are now making use of more information brought by the rejection of a first-level candidate, we expect this algorithm to be more efficient than the RWDR with independent candidates.

To facilitate the comparison between the different sampling algorithms, we shall determine the form of asymptotically optimal proposal scalings. To this end, we consider the random walk delayed rejection sampling with common proposal described below:

Algorithm 4 (Two-step RWDR algorithm with common proposal)**.**
(a) Given the current state $\mathbf{X}[n]$ *of the Markov chain at time* n*, two proposals*

$$\left(\mathbf{Y}^i[n+1]\right)_{i=1,2} \triangleq \left(Y_t^i[n+1], 0 \le t \le T\right)_{i=1,2} ,$$

are generated according to

$$Y_t^i[n+1] = X_t[n] + T^{-1/2} U_t^i[n+1] , \quad 0 \le t \le T, \quad i = 1, 2 \tag{23}$$

where

(a) for any $t \in \{0, \ldots, T\}$, $\begin{pmatrix} U_t^1[n+1] \\ U_t^2[n+1] \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \ell_1^2 & \ell_1\ell_2 \\ \ell_1\ell_2 & \ell_2^2 \end{bmatrix}\right)$,

(b) The $T+1$ *random vectors* $\{(U_t^1[n+1], U_t^2[n+1])\}_{t=0}^T$ *are conditionally independent given* $\mathcal{F}_n$*, where* $\mathcal{F}_n$ *is defined in* (8).

(b) The first proposal $\mathbf{Y}^1[n+1]$ *is accepted with probability* $\alpha_1(\mathbf{X}[n]; \mathbf{Y}^1[n+1])$ *where* $\alpha_1(x; y^1)$ *is defined in* (10).

(c) Upon rejection of the first candidate $\mathbf{Y}^1[n+1]$*, propose the second candidate* $\mathbf{Y}^2[n+1]$ *and accept it with probability:*

$$\bar{\alpha}_2(\mathbf{X}[n]; \mathbf{Y}^2[n+1])$$

where

$$\begin{aligned} \bar{\alpha}_2(x; y) &= 1 \wedge \frac{\pi(y)\left[1 - \pi\left[\bar{\Psi}(y, x)\right]/\pi(y)\right]_+}{\pi(x)\left[1 - \pi\left[\bar{\Psi}(x, y)\right]/\pi(x)\right]_+} \\ &= 1 \wedge \frac{\left[\pi(y)/\pi(x) - \pi\left[\bar{\Psi}(y, x)\right]/\pi(x)\right]_+}{\left[1 - \pi\left[\bar{\Psi}(x, y)\right]/\pi(x)\right]_+} , \end{aligned}$$

and $\bar{\Psi} : \mathsf{X}^{T+1} \times \mathsf{X}^{T+1} \to \mathsf{X}^{T+1}$ *such that for all* $(x, y) \in \mathsf{X}^{T+1} \times \mathsf{X}^{T+1}$,

$$\bar{\Psi}(x, y) = x + \ell_1 \ell_2^{-1}(y - x) .$$

Otherwise, reject it.

Similarly to the notation used earlier, the conditional probability of accepting the first move is given by (13). The conditional probability of rejecting the first move and accepting the second one is equal to $\bar{\alpha}_2(\mathbf{X}[n])$, where

$$\bar{\alpha}_2(\mathbf{x}) = \mathbb{E}\left[A_2\left(L_{0,T}(\mathbf{x}, \mathbf{U}^1), L_{0,T}(\mathbf{x}, \mathbf{U}^2), L_{0,T}(\mathbf{x}, (1-\ell_1\ell_2^{-1})\mathbf{U}^2)\right)\right] \tag{24}$$

with $L_{0,T}$ as defined in (15) and

$$A_2(u, v, w) = [1 - \mathrm{e}^u]_+ \wedge [\mathrm{e}^v - \mathrm{e}^w]_+ \ . \tag{25}$$

Define, for $(\kappa_1, \kappa_2, \kappa_3) \in \mathbb{R}^3$,

$$\tilde{\alpha}_2(\kappa_1, \kappa_2, \kappa_3) \triangleq \mathbb{E}\left[A_2(\kappa_1 G - \kappa_1^2/2, \kappa_2 G - \kappa_2^2/2, \kappa_3 G - \kappa_3^2/2)\right] \ , \tag{26}$$

where G is a standard Gaussian variable. We obtain the following result.

Theorem 4. *Assume (A1) and consider Algorithm 4. Suppose that the algorithm starts in stationarity, i.e.* $\mathbf{X}[0]$ *is distributed according to the target density* π.

Consider the first component of the rescaled process $\{W_T[t], t \geq 0\}$ *defined in* (12). *Then* $\{W_T[t], t \geq 0\} \Rightarrow \{W[t], t \geq 0\}$, *where* $W[0]$ *is distributed according to the density* f *in (A1) and* $\{W[t], t \geq 0\}$ *satisfies the Langevin stochastic differential equation (SDE)*

$$\mathrm{d}W[t] = \lambda^{1/2}(\ell_1, \ell_2)\mathrm{d}B[t] + \frac{1}{2}\lambda(\ell_1, \ell_2)\left[\ln f\right]'(W[t])\mathrm{d}t \ ,$$

with $\lambda(\ell_1, \ell_2) \triangleq \lambda_1(\ell_1) + \lambda_2(\ell_1, \ell_2)$, $\lambda_1(\ell_1) = \ell_1^2\tilde{\alpha}_1(\ell_1\mathcal{I}^{1/2})$, *and*

$$\lambda_2(\ell_1, \ell_2) = \ell_2^2\tilde{\alpha}_2(\ell_1\mathcal{I}^{1/2}, \ell_2\mathcal{I}^{1/2}, (\ell_2 - \ell_1)\mathcal{I}^{1/2}) \ . \tag{27}$$

The quantities $\tilde{\alpha}_1$, $\mathcal{I}$, *and* $\tilde{\alpha}_2$ *are defined in* (16), (17), *and* (26), *respectively.*

The proof of this theorem is postponed to Appendix C.

As previously mentioned, $\tilde{\alpha}_1\left(\ell_1\mathcal{I}^{1/2}\right)$ is the average acceptance rate of the first move in stationarity, whereas $\tilde{\alpha}_2\left(\ell_1\mathcal{I}^{1/2}, \ell_2\mathcal{I}^{1/2}, (\ell_2 - \ell_1)\mathcal{I}^{1/2}\right)$ is the average acceptance rate of the second move. Note that, in this case, the limiting acceptance rate of the second move does not vanish: by proposing correlated proposals, the second move can be accepted with positive probability in this high-dimensional limit.

From this theorem, we already notice that the random walk delayed rejection algorithm with common proposal is always more efficient than that with independent proposals. The actual asymptotically optimal proposal variances and acceptance rates, which are valid for general target distributions with i.i.d. components satisfying Assumption (A1), are provided in the following corollary.

Corollary 5. *In the setting of Theorem 4, the proposal scales* ℓ_1 *and* ℓ_2 *optimizing the speed measure of the diffusion are given by* $\ell_1^\star = -\ell_2^\star = 2.39\mathcal{I}^{-1/2}$. *The first- and second-level expected acceptance rates of the algorithm respectively satisfy*

$$\tilde{\alpha}_1(\ell_1^\star\mathcal{I}^{1/2}) = \tilde{\alpha}_2(\ell_1^\star\mathcal{I}^{1/2}, \ell_2^\star\mathcal{I}^{1/2}, (\ell_2^\star - \ell_1^\star)\mathcal{I}^{1/2}) = 0.234 \ .$$

The optimal values observed in the corollary are perhaps surprising. The size of the first move should be exactly the same as for the Metropolis algorithm; the second candidate, which is proposed along the the same direction as the first candidate, should be exactly symmetrical to the first move with respect to the current state.

Versions considering random walk increments are natural choices for implementing the DR-Common algorithm described in Algorithm 3. Such versions are easy to implement and allow the study of scaling limits, which are not generally available; accordingly, we do not expect optimal scaling results to be obtainable for an arbitrary function Ψ.

Algorithm 4 is, among algorithms with random walk increments, a specific case that considers two candidates Y^1 and Y^2 along a common search direction. It would be possible to consider alternative forms for the covariance matrix in Algorithm 4. However, as far as optimal scaling results are concerned, we do not believe that there exists another covariance matrix that yields better results (higher efficiency and higher asymptotically optimal acceptance rate) under the framework considered (i.e. high-dimensional i.i.d. target distributions). With such target densities one cannot expect to consistently do better than by proposing a candidate as far away as possible from the rejected value, i.e. in the opposite direction.

Delayed rejection algorithms with random walk increments have many similarities with multiple-try Metropolis algorithms, and this also holds in the case of optimal scaling results. Arbitrary covariance matrices have been considered in Bédard, Douc and Moulines (2012) for the multiple correlated-try Metropolis algorithm, where it was established that extreme antithetic candidates result in an optimal covariance matrix (under the same framework as considered here). In that paper, an even more extreme form of dependency was considered through the MTM hit-and-run algorithm, which generates candidates deterministically (and simultaneously) along a search direction. It was concluded that this algorithm, which is the MTM equivalent of Algorithm 4, was the most efficiency method in that paper.

4. Numerical Simulations

4.1. Validation of Theorem 1

To illustrate the theoretical results stated in Theorem 1, we perform a simulation study on a simple standard Gaussian target distribution. Specifically, we suppose that $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_T)$ and use a two-step random walk delayed rejection algorithm with independent proposals (RWDR-I) to sample from this distribution. The first-stage RWDR-I candidates are distributed according to the optimally tuned $\mathcal{N}(\mathbf{x}, 5.66/T \times \mathrm{I}_T)$ (see Corollary 2), while the second-stage RWDR-I proposal distribution is taken to be a $\mathcal{N}\left(\mathbf{x}, \ell_2^2/T \times \mathrm{I}_T\right)$; we test various values for ℓ_2^2 ranging from $1/T$ to $0.3 \times T$.

For $T = 20$ and each of these second-level proposal variances, we run 300,000 iterations of the two-step random walk delayed rejection algorithm described

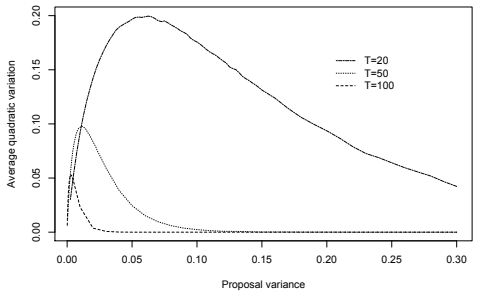

FIG 1. *Average quadratic variation as a function of proposal variances for the two-step RWDR algorithm with independent proposals, applied to sample from a Gaussian target. The dotted lines are the results of simulation studies in different dimensions.*

above and estimate efficiency by recording the average quadratic variation (or average square jumping distance) of the algorithm; similar simulations are then repeated for $T = 50, 100$. The average quadratic variation is a convenient efficiency measure as it is a function of the sample obtained only, *i.e.* it is independent of the specific estimates that we might be interested in obtaining from the Markov chain; it is computed as $\sum_{i=1}^{T} \sum_{j=1}^{N} \left(X_i[j] - X_i[j-1]\right)^2 / N$, where N is the number of iterations performed (see (Roberts and Rosenthal, 1998)).

The graph in Figure 1 displays average quadratic variations as a function of proposal variances for $T = 20, 50, 100$. As the dimension of the target distribution increases, the optimal proposal variance converges towards 0, and proposal variances located in the neighborhood of this optimal value tend to generate candidates that are automatically rejected, inducing average quadratic variations that are null.

Figure 2 aims at empirically comparing the performance of the Metropolis algorithm with a Gaussian proposal to that of the two-step RWDR-I algorithm described above. For the comparison to be fair, both algorithms should be tuned according to their respective optimal proposal variances. Accordingly, the first-stage proposal distribution of the RWDR-I algorithm is identical to the proposal distribution of the Metropolis algorithm, *i.e.* $\mathcal{N}\left(\mathbf{x}, 5.66/T \times \mathrm{I}_T\right)$; the second-stage candidates of the RWDR-I algorithm are generated according to a $\mathcal{N}\left(\mathbf{x}, 5.66^2/T^2 \times \mathrm{I}_T\right)$ (see Remark 2).

In Figure 2, we performed 200,000 iterations of the Metropolis and RWDR-I algorithms in dimensions $T = 5, 10, 20, 50, 100$, and recorded the average quadratic variations. Both curves quickly converge towards the asymptotic efficiency value, $\lambda_1(\hat{\ell}_1) = 2(5.66)\Phi(-\sqrt{5.66}/2)$; even in small dimensions, the efficiency gain from the RWDR-I algorithm does not seem worth its extra computational effort.

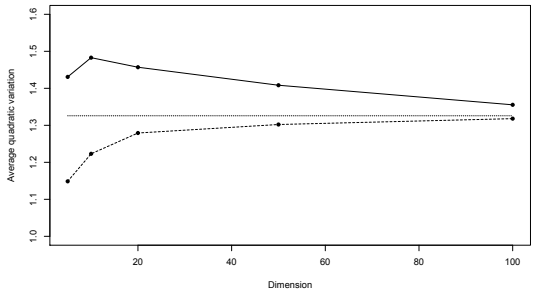

FIG 2. *Average quadratic variation of the Metropolis algorithm (bottom curve) and of the two-step RWDR algorithm with independent proposals (top curve) as a function of the target distribution dimension, T. The constant line represents the theoretical asymptotic efficiency.*

4.2. Validation of Theorem 4

To validate the theoretical results and to compare the efficiency of the RWDR with common proposal (RWDR-C) to that of the Metropolis algorithm, we consider the toy example discussed in the previous section where $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_T)$. As a first-level proposal distribution, we use a $\mathcal{N}(\mathbf{x}, 5.66/T \times \mathrm{I}_T)$; the second-level proposal distribution is taken to be a $\mathcal{N}\left(\mathbf{x}, \ell_2^2/T \times \mathrm{I}_T\right)$, where we test 100 different values for ℓ_2 ranging from $-\sqrt{15}$ to $-\sqrt{0.3}$ and from $\sqrt{0.1}$ to $\sqrt{5}$. For each of these second-level proposal scaling values, we run 300,000 iterations of a two-step RWDR-C algorithm and estimate efficiency by recording the average quadratic variation (AQV); we also record the proportion of second-level candidates that have been retained in the process (second-level acceptance rate).

We perform these simulations for $T = 20, 50, 100$ and combine the three resulting curves of AQV versus ℓ_2 on a graph (Figure 3, left). We also include the theoretical efficiency curve of $\lambda_1(\ell_1^\star) + \lambda_2(\ell_1^\star, \ell_2)$ versus ℓ_2. In a similar fashion, the right graph of Figure 3 illustrates the relationship between the AQV and the second-level acceptance rates of the algorithm; we however focus here on second-level candidates that are antithetic, *i.e.* for which ℓ_2 is negative. The solid line represents the theoretical efficiency curve against $\tilde{\alpha}_2(\ell_1^\star \mathcal{I}^{1/2}, \ell_2 \mathcal{I}^{1/2}, (\ell_2 - \ell_1^\star)\mathcal{I}^{1/2})$, the second-level asymptotically expected acceptance rate evaluated at $\ell_1^\star$. In both graphs, we also include the theoretical efficiency curve of a Metropolis algorithm with a $\mathcal{N}(\mathbf{x}, 5.66/T \times \mathrm{I}_T)$ proposal distribution (which is independent of ℓ_2, and thus constant).

As expected from Corollary 5, a global mode occurs at $\ell_2^\star = -\ell_1^\star = -2.39$ on the left graph, hence it is optimal in the present setting to favor symmetrically antithetic candidates. We also notice a local mode around $\ell_2 = 1.2$; this value represents the optimal value for ℓ_2 in the case where the second-level candidate is restricted to move in the same direction as ℓ_1. A comparison with the constant efficiency curve of the Metropolis algorithm shows that an optimal tuning of the

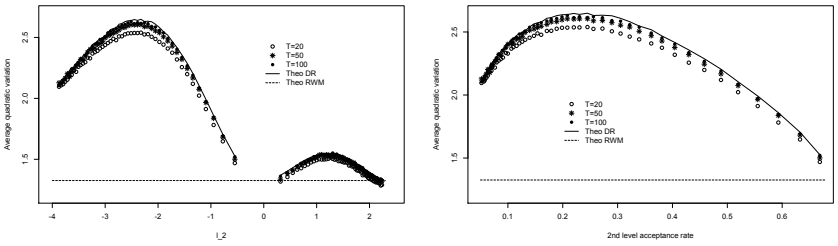

FIG 3. *Average quadratic variation as a function of ℓ_2 (left) and the second-level acceptance rate (right) for the two-step RWDR algorithm with common proposal, applied to sample from a Gaussian target. The symbols are the results of simulation studies in different dimensions, while the dotted lines represent the theoretical efficiency curves.*

proposal variances for the RWDR-C algorithm leads to a sampling algorithm that is twice as efficient as an optimally tuned Metropolis algorithm.

Before concluding this section, we finally use the toy example to compare the performances of the RWDR-I and RWDR-C algorithms described in Sections 4.1 and 4.2; to this end, we run simulation studies similar to those described in these sections. The left graph of Figure 4 illustrates the relationship between the average quadratic variation and the dimension T, for optimally tuned Metropolis, RWDR-I, and RWDR-C algorithms; even for small T, the RWDR-C algorithm is seen to be more efficient than the Metropolis and RWDR-I algorithms. The right graph of Figure 4 represents the AQV versus the global acceptance rates of the RWDR-I and RWDR-C algorithms in various dimensions. The symbols are used to illustrate the behavior of the algorithm with antithetic candidates, while the dotted lines are used for the independent candidates. Poorly tuned RWDR-C algorithms have a higher AQV than optimally tuned RWDR-I algorithms. It is however interesting to note that a poor tuning of the RWDR-C algorithm has a more significant impact than a poor tuning of the RWDR-I algorithm.

When taking the computational effort into account, the convention is to divide efficiency by the number of candidates to generate at every iteration; in an algorithm requiring two candidates per iteration, we would thus need to halve efficiency in adjusting for the computational cost. Given that a RWDR algorithm with common proposal does not require the generation of two candidates at every iteration, but only in about 76.6% of iterations (when tuned optimally), this signifies that this method shall generally be more efficient than the Metropolis algorithm, even when taking the computational effort of accepting/rejecting an extra candidate into consideration (a gain of approximately 15% in efficiency). If we add this to the fact that computing second-level acceptance probabilities is often extremely cheap (due to the relationship between $\ell_1^\star$ and $\ell_2^\star$), implementing the RWDR algorithm with common proposal might bring a significant advantage over the Metropolis algorithm, and could thus be favored in a large number of situations.

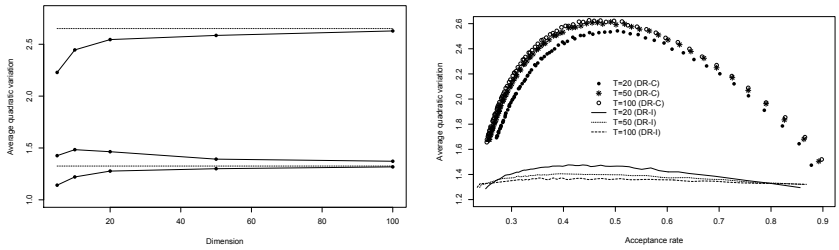

FIG 4. *Left: AQV of the Metropolis (bottom curve), the RWDR-I (middle curve), and RWDR-C (top curve) as a function of the target distribution dimension, T. The constant lines represent the corresponding theoretical asymptotic efficiencies. Right: AQV as a function of the acceptance rate for the RWDR-I and RWDR-C algorithms. The symbols are the results of simulation studies in different dimensions for the antithetic proposals, while the dotted lines are the simulation results for the independent proposals.*

TABLE 1
Number of latent membranous lupus nephritis cases (numerator) and total number of cases (denominator), for each combination of the covariates values

	IgA				
IgG	0	0.5	1	1.5	2
-3.0	0/1	-	-	-	-
-2.5	0/3	-	-	-	-
-2.0	0/7	-	-	-	0/1
-1.5	0/6	0/1	-	-	-
-1.0	0/6	0/1	0/1	-	0/1
-0.5	0/4	-	-	1/1	-
0.0	0/3	-	0/1	1/1	-
0.5	3/4	-	1/1	1/1	1/1
1.0	1/1	-	1/1	1/1	4/4
1.5	1/1	-	-	2/2	-

4.2.1. Logistic regression model with lupus data

We finally compare the performances of the RWM algorithm, RWDR algorithm with independent proposals (RWDR-I), and RWDR algorithm with antithetic proposals (RWDR-C) in the case of a logistic regression model. The data considered for that example is the same as Example 4.2 of Craiu and Lemieux (2007); the aim of the experiment is to predict the occurrence of latent membranous lupus in patients with the help of two clinical covariates, IgG3-IgG4 and IgA, that respectively measure the levels of type-G and type-A immunoglobulin. Table 1 has been reproduced from Craiu and Lemieux (2007) and shows the measurements of the covariates on 55 patients, of which 18 have been diagnosed with the disease.

We consider the following logistic regression model

$$\text{logit}\mathbb{P}\left[Y_i = 1\right] = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i},$$

where $X_i' = (1, X_{i1}, X_{i2})$ is the vector of covariates for the i-th patient. The selected prior distribution for $\beta = (\beta_0, \beta_1, \beta_2)'$ is a multivariate normal with mean $(0, 0, 0)'$ and covariance matrix $100^2 \mathrm{I}_3$. It follows that the posterior density for β satisfies

$$\pi(\beta|x, y) \propto \mathrm{e}^{-\frac{0.5}{100^2}\sum_{j=0}^{2}\beta_j^2} \prod_{i=1}^{55} \frac{[\exp(X_i'\beta)]^{y_i}}{1 + \exp(X_i'\beta)} \, .$$

We obtain samples from π using the RWM, RWDR-I, and RWDR-C algorithms. In all three cases, the proposal distribution $q(\cdot|\beta)$ is a normal centered at β; the covariance matrix of the proposal is given by $\sigma^2 \mathrm{I}_3$, where $\sigma > 0$ is dependent upon the chosen sampling method. As the target distribution is neither high-dimensional nor formed of i.i.d. components, we compare two approaches to tune the parameters σ of the algorithms considered. In the first approach, we tune σ so as to yield acceptance rates that are close to the theoretical rates introduced in Sections 2 and 3: 0.234 for the RWM algorithm and 0.468 for the RWDR-C algorithm. The target distribution being three-dimensional, it would not be fair to tune the RWDR-I to accept only 25% of the proposed candidates. We thus let $\sigma_1 = 2.25$ as for the RWM algorithm (see Table 2), and then we find the corresponding σ_2 using Remark 2. In the second approach, we select the proposal variances leading to optimal performances in terms of the average quadratic variation.

We would be interested in estimating β_1 and $p_{25} = \mathbb{1}_{\{\beta_1 > 25\}}$. For each algorithm, we start the chain at $\beta = 0$ and we draw a total of 3,064,800 values, of which the first 5,000 values in the chain are discarded as burn-in period. The remaining values are divided into $M = 300$ samples of size $N = 10,000$; this is achieved by discarding, between each sample, 200 values from the chain. Using the above samples, we then compute the Monte Carlo mean squared error (MC-MSE) of our estimates. By denoting the j-th replicate of β_1 within the i-th sample by b_{ij}, the Monte Carlo mean squared error is expressed as

$$\mathcal{M}(\beta_1) = (\bar{b}_{\cdot\cdot} - \mathbb{E}\left[\beta_1|\text{data}\right])^2 + \frac{1}{M-1}\sum_{i=1}^{M}(\bar{b}_{i\cdot} - \bar{b}_{\cdot\cdot})^2 \, ,$$

where $\bar{b}_{i\cdot} = \sum_{j=1}^{N} b_{ij}/N$ for $i = 1, \ldots, M$ and $\bar{b}_{\cdot\cdot} = \sum_{i=1}^{M}\sum_{j=1}^{N} b_{ij}/MN$. A similar expression may be obtained for the MC-MSE of p_{25}. As mentioned in Craiu and Lemieux (2007), we use $\mathbb{E}\left[\beta_1|\text{data}\right] \approx 13.57$ and $\mathbb{E}\left[p_{25}|\text{data}\right] \approx 0.073$, which have been obtained through numerical integration.

Table 2 provides, for each algorithm and for each tuning approach considered, the MC-MSE for β_1 and p_{25}, along with the average quadratic variation. It also specifies the proposal scalings used as well as the global acceptance rates obtained.

For this example, the RWDR-C algorithm seems to be performing much more efficiently than the RWM and RWDR-I algorithms. Efficiency is improved by a factor lying between 1.8 and 2, depending on the efficiency measure considered; a tuning impact (theoretical versus optimal) is present, as all efficiency measures

TABLE 2
For each algorithm (RWM, RWDR-I, RWDR-C) and tuning approach (theoretical, optimal), the following quantities are provided: proposal scalings, acceptance rate, MC-MSEs for β_1 and p_{25}, AQV.

Tuning	Method	σ_1	σ_2	Acc. rate	$\mathcal{M}(\beta_1)$	$\mathcal{M}(p_{25})$	AQV
Theoretical	RWM	2.15	-	0.253	1.899	.00204	2.019
	RWDR-I	2.15	1.00	0.582	1.795	.00182	2.722
	RWDR-C	2.15	2.15	0.426	0.987	.00112	3.646
Optimal	RWM	2.60	-	0.196	1.710	.00171	2.078
	RWDR-I	2.60	2.00	0.364	1.160	.00124	3.095
	RWDR-C	2.60	2.60	0.337	0.863	.00090	3.790

(for all three algorithms) are improved under the optimal tuning scheme, but does not seem major. When tuned optimally, the RWDR-I algorithm offers an improvement between 40%-50%; in this case, tuning the RWDR-I optimally as opposed to theoretically seems to have a significant impact.

We note here that the target distribution considered is a low-dimensional, correlated distribution, which violates important assumptions under which the optimal scaling results of Theorem 1 and Corollary 5 were obtained. In spite of the violation of the assumptions, it is important to outline the fact that the theoretical tuning offers an interesting option for RWDR-C algorithms, as the results obtained under this tuning scheme are not too different from the optimal results obtained. The RWDR-C method displays a substantial improvement over the RWM and RWDR-I algorithms, and thus constitutes an efficient alternative to these methods. Due to the nature of antithetic proposals, the computation of second-level acceptance rates is often practically free. This is the case here, as once the target density is computed for a proposed value, one can directly compute the same quantity for the antithetic increment without affecting the computational intensity of the algorithm; this is achieved by a simple manipulation on the target density $\pi(\beta|x, y)$.

5. Discussion

General conclusions about the efficiency of delayed rejection methods can hardly be drawn without considering specific expressions for the target density and the computational intensity of the algorithm. Nonetheless, the asymptotic theory previously considered gives us good indications about the potential of these samplers.

In addition to providing users with optimal scaling values, the asymptotic theory derived in Section 2 could be seen as a guideline to improve the two-step RWDR algorithm with independent components. Theorem 1 warns us about the issues of this algorithm in high dimensions; according to the limiting diffusion obtained, it is a loss of time and resources to choose such a sampler over the RWM algorithm to sample from high-dimensional target densities. The main

problem in its proposal scheme seems to come from the fact that not enough of the available information is used in the generation of the second-level candidate. Since this method is asymptotically equivalent to the RWM algorithm but more expensive computationally, the latter remains a better option.

Nonetheless, it is worth mentioning that regardless of how large is d, the RWDR algorithm with independent components outperforms the "corresponding" RWM algorithm (i.e. a RWM sampler with the same first-level proposal) in the Peskun and covariance orderings (see Mira (2001b) and the references therein). In other words, given that one wishes to estimate the expected value of any squared integrable function (with respect to the target), the RWDR produces estimators that have a smaller asymptotic variance than the RWM in the CLT. This is explained by the fact that the Markov chain has a higher probability of moving away from the current position under the RWDR strategy, as discussed in Mira (2001b). This relative advantage washes out as d increases (expecially if no clever way to construct the higher level candidates is designed) and does not take into account the additional computational time required to run higher levels.

The potential in learning from rejected candidates to propose new candidates before incrementing time should however not be dismissed. By generating correlated candidates from a common proposal, it is possible to obtain a nontrivial limiting process involving second-level candidates, which improves on the RWM algorithm. It appears that to optimize the efficiency of this version of RWDR methods, one should favor antithetic candidates, where the second-level proposal is exactly symmetrical to the first-level proposed value around the current state. This means that upon the rejection of a candidate, one should propose an increment of equal magnitude, but in the opposite direction. In the current framework (high-dimensional i.i.d. target densities), the improvement over the RWM algorithm is significant: the asymptotic efficiency of the algorithm is doubled. Even in assuming that the computation of two acceptance probabilities (for the first- and second-level candidates) in a given iteration is twice as demanding as the computation of only one acceptance probability, the net gain is positive as delayed rejection methods do not require the generation of two candidates at every iteration performed. Moreover, as witnessed in the numerical studies, the relationship between the optimal values $\ell_1^\star$ and $\ell_2^\star$ shall often result in second-level acceptance probabilities that are computationally free, making the RWDR with common proposal an even more attractive option.

Although the asymptotic results of Section 3 are not directly applicable to the lupus example of Section 4.2.1, the results obtained with the RWDR with antithetic proposals still show a significant improvement over the RWM algorithm. Under the violation of the target density assumptions, antithetic proposals might not always be the optimal choice. Candidates in a common direction, for instance, might become optimal for some target densities with correlated components. Regardless of the optimal relationship between the first- and second-level proposals in specific problems, it seems like the random walk delayed rejection algorithm with common proposal shows a great deal of potential in many applications. The optimal correlation between the candidates in a given

iteration might still lead to a reduction in the cost of the second-level acceptance probability, even when the candidates are not symmetrically antithetic.

The results obtained here could be extended to delayed rejection algorithms with a larger number of tries per iteration. In the case of random walk increments, we could propose further candidates based on the random vector Z; this would yield a proposal scheme similar to the multiple-try Metropolis hit-and-run algorithm of Bédard, Douc and Moulines (2012), but in which the candidates are generated successively rather than simultaneously. However, given that the greatest gain in efficiency is obtained when proposing a second candidate that is exactly symmetrical to the first one around the current state x, then the marginal efficiency gain resulting from the inclusion of a third candidate in the same direction would necessarily be of lesser impact than the gain from a second candidate. In fact, we can reasonably expect that the marginal efficiency gain will decrease with each additional candidate in a given iteration. It is thus unlikely that futher candidates are worth the additional computational effort needed for their implementation.

Other types of generalizations would be possible and would require, for instance, determining how to choose the direction in which a third candidate should be proposed. Let us suppose that $Z = (z_1, z_2)$; then, the second candidate would be based on $(-z_1, -z_2)$. We could thus consider a new, non-random direction for a third candidate, say $(-z_1, z_2)$, as well as its opposite $(z_1, -z_2)$ for a fourth candidate. Again, such a generalization could be managed theoretically, but it is still unclear whether the marginal efficiency gain would be worth the extra computational effort.

Appendix A: Scaling Approximations - General Results

We recall in this section some theoretical results obtained in Bédard, Douc and Moulines (2012) that shall be useful for proving Theorems 1 and 4. Although these results were derived for the analysis of multiple-try Metropolis algorithms, they provide a set of conditions that might be used for the analysis of general MCMC algorithms involving auxiliary random variables. As a compromise between self-containment and conciseness, we include all of the results needed but omit their proofs; the reader may refer to Bédard, Douc and Moulines (2012) for more detail about concepts discussed in this section.

Recall that $\zeta_T : \mathbb{R}^{T+1} \to \mathbb{R}$ is the projection on the first coordinate, that is $\zeta_T(\mathbf{x}) = x_0$. For any function $h : \mathbb{R} \to \mathbb{R}$, define by Ph the function on $\mathbb{R}^{T+1}$

$$Ph : \mathbf{x} \mapsto Ph(\mathbf{x}) = h(x_0) = h \circ \zeta_T(\mathbf{x}) \; . \tag{A.1}$$

Let $(\mathbf{X}[n],\ n \in \mathbb{N})$ be homogeneous Markov chains taking values in $\mathbb{R}^{T+1}$ with transition kernel Q. For all $s \geq 0$, denote

$$W_T[s] = \zeta_T \left(\mathbf{X} \left[\lfloor Ts \rfloor \right] \right) \; . \tag{A.2}$$

Define $G_T = T(Q - I)$ and denote by C_c^∞ the set of compactly supported indefinitely continuously differentiable functions defined on $\mathbb{R}$. Let $\{\mathsf{F}_T\}_{T \geq 0}$ be a sequence of Borel subsets of $\mathbb{R}^T$ and consider the following assumptions:

(**B1**) For all $T \in \mathbb{N}$, the transition kernel Q has a unique stationary distribution denoted by π. Moreover, for any Borel non negative function h on $\mathbb{R}$,

$$\pi(Ph) = \int h(x_0) f(x_0) \mathrm{d}x_0 \ , \tag{A.3}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a probability density function.

(**B2**) $\lim_{T\to\infty} \pi\left(\mathbb{R} \times \mathsf{F}_T\right) = 1$.

(**B3**) There exist $p > 1$ such that for any $h \in C_c^\infty$,

$$\sup_{T\geq 0} \int \sup_{x_{1:T}\in\mathbb{R}^T} \left|G_T\left[Ph\right](\mathbf{x})\right|^p f(x_0)\mathrm{d}x_0 < \infty \ ,$$

where $x_{1:T} \triangleq (x_1, \ldots, x_T)$.

(**B4**) There exists a Markov process $(W[s], s \in \mathbb{R}^+)$ with cadlag sample paths and (infinitesimal) generator G such that C_c^∞ is a core for G and for any $h \in C_c^\infty$,

$$\lim_{T\to\infty} \int \sup_{x_{1:T}\in\mathsf{F}_T} \left|G_T\left[Ph\right](\mathbf{x}) - Gh(x_0)\right| f(x_0)\mathrm{d}x_0 = 0 \ .$$

Theorem 6. *Assume (**B1-4**). Then, $W_T \Longrightarrow W$ in the Skorokhod topology where $W[0]$ is distributed according to f.*

Now, consider a sequence of homogeneous Markov chains $(\mathbf{X}[n],\ n \in \mathbb{N})$ taking values in $\mathbb{R}^{T+1}$ with transition kernel Q satisfying, for any measurable bounded function h on $\mathbb{R}$

$$\begin{aligned} &Q\left[Ph\right](\mathbf{x}) - h(x_0) \\ &= \mathbb{E}\left[h(\zeta_T(\mathbf{X}[1])) \,|\, \mathbf{X}[0] = \mathbf{x}\right] - h(x_0) \\ &= \sum_{j=1}^{K} \mathbb{E}\left[\left\{h(x_0 + T^{-1/2}U^j) - h(x_0)\right\} \beta^j(\mathbf{x}, T^{-1/2}, x_0 + T^{-1/2}U^j)\right] \end{aligned} \tag{A.4}$$

where $\{U^j\}_{1\leq j\leq K}$ are random variables and, for $j \in \{1, \ldots, K\}$, $\beta^j : \mathbb{R}^{T+1} \times \mathbb{R} \times \mathbb{R} \to [0, 1]$,

$$(\mathbf{x}, \eta, y) \mapsto \beta^j(\mathbf{x}, \eta, y) \ , \tag{A.5}$$

are nonnegative measurable functions. When applied to delayed rejection algorithms, $\beta^j(\mathbf{x}, \eta, y)$ will be the average probability of accepting the j-th try at a given iteration when the Markov chain is in state $\mathbf{x}$; it is however not required to specify this function further at this stage.

For $j \in \{1, \ldots, K\}$, $\eta \geq 0$, and $u \in \mathbb{R}$, define

$$\tilde{\beta}^j(\mathbf{x}, \eta, u) = \beta^j(\mathbf{x}, \eta, x_0 + \eta u) \ . \tag{A.6}$$

Alternatively, Eq. (A.6) can be rewritten as follows: for any $\mathbf{x} \in \mathbb{R}^{T+1}$, $\eta \in \mathbb{R}$ and $y \in \mathbb{R}$,

$$\tilde{\beta}^j(\mathbf{x}, \eta, y) = \beta^j(\mathbf{x}, \eta, (y - x_0)/\eta) \ , \tag{A.7}$$

with the convention $0/0 = 0$. Whereas $(W_T[s],\ s \in \mathbb{R}^+)$, defined in (A.2), is not itself a Markov process, it is a progressive $\mathbb{R}$-valued process and this section presents conditions under which $(W_T[s],\ s \in \mathbb{R}^+)$ converges in the Skorokhod topology to the solution of a Langevin SDE. As stated in Bédard, Douc and Moulines (2012), in practice (**B3-4**) may be obtained by checking the following practical assumptions:

(**C1**) There exist constants $\{a_j\}_{j=1}^K \in \mathbb{R}^K$ such that for all $j \in \{1, \ldots, K\}$,

$$\lim_{T\to\infty} \int \sup_{x_{1:T}\in \mathsf{F}_T} \left|\beta^j(\mathbf{x}, 0, x_0) - a_j\right| f(x_0)\mathrm{d}x_0 = 0 \ .$$

(**C2**) There exists a family $\{w^j\}_{j=1}^K$ of measurable functions $w^j : \mathbb{R} \to \mathbb{R}$ such that for all $j \in \{1, \ldots, K\}$,

$$\lim_{T\to\infty} \int \sup_{x_{1:T}\in \mathsf{F}_T} \left|\frac{\partial \beta^j}{\partial y}(\mathbf{x}, 0, x_0) - w^j(x_0)\right| f(x_0)\mathrm{d}x_0 = 0 \ . \tag{A.8}$$

(**C3**) There exists $p > 1$ such that for any $j \in \{1, \ldots, K\}$,

$$\sup_{T\geq 0} \int \sup_{x_{1:T}\in \mathbb{R}^T} \left|\frac{\partial \beta^j}{\partial y}(\mathbf{x}, 0, x_0)\right|^p f(x_0)\mathrm{d}x_0 < \infty \ , \tag{A.9}$$

$$\sup_{T\geq 0} \int \sup_{x_{1:T}\in \mathbb{R}^T} \left(\mathbb{E}\left[(U^j)^2 \sup_{0\leq\eta\leq T^{-1/2}} \left|\frac{\partial \tilde{\beta}^j}{\partial \eta}\left(\mathbf{x}, \eta, U^j\right)\right|\right]\right)^p f(x_0)\mathrm{d}x_0 < \infty \ , \tag{A.10}$$

$$\sup_{T\geq 0} \int \sup_{x_{1:T}\in \mathbb{R}^T} \left(\mathbb{E}\left[|U^j| \sup_{0\leq\eta\leq T^{-1/2}} \left|\frac{\partial^2 \tilde{\beta}^j}{\partial \eta^2}(\mathbf{x}, \eta, U^j)\right|\right]\right)^p f(x_0)\mathrm{d}x_0 < \infty \ . \tag{A.11}$$

(**C4**) For any $j \in \{1, \ldots, K\}$, $\mathbb{E}\left[U^j\right] = 0$ and $\mathbb{E}\left[|U^j|^3\right] < \infty$.

Theorem 7. *Assume (**B1-2**) and (**C1-4**). Then, $W_T \Longrightarrow W$ in the Skorokhod topology where $W[0]$ is distributed according to f and $(W[s],\ s \in \mathbb{R}^+)$ satisfies the Langevin SDE*

$$\mathrm{d}W[t] = \sqrt{\lambda}\mathrm{d}B[t] + \frac{1}{2}\lambda \left[\ln f\right]'(W[t])\mathrm{d}t \ , \tag{A.12}$$

with

$$\lambda = \sum_{j=1}^K \mathbb{V}\mathrm{ar}[U^j] a_j \ . \tag{A.13}$$

In addition, for any $x \in \mathbb{R}$,

$$\sum_{j=1}^K \mathbb{V}\mathrm{ar}[U^j] w^j(x) = \frac{\lambda}{2}\left[\ln f\right]'(x) \ . \tag{A.14}$$

Lemma 8. *Assume (**C1-4**). Then (**B3-4**) are satisfied where G is the generator of the Langevin diffusion* (A.12)

$$Gh(x) \triangleq \frac{\lambda}{2}\left(h'(x)\left[\ln f\right]'(x) + h''(x)\right) . \tag{A.15}$$

Theorems 6 and 7 will be used for the analysis of the RWDR with independent (respectively common) proposals. To this end, some further definitions, lemmas and propositions issued from Bédard, Douc and Moulines (2012) shall be useful. Let η such that

$$0 < \eta < 1/4 . \tag{A.16}$$

Then, define the sequence of sets $\{\mathsf{F}_T\}_{T=0}^{\infty}$ by

$$\mathsf{F}_T = \left\{x_{1:T} \in \mathbb{R}^T, |\mathcal{I}_T(x_{1:T}) - \mathcal{I}| \vee |\mathcal{J}_T(x_{1:T}) - \mathcal{I}| \vee \mathcal{S}_T(x_{1:T}) \leq T^{-\eta}\right\} , \tag{A.17}$$

where, for any $x_{1:T} \in \mathbb{R}^T$, we let

$$\mathcal{I}_T(x_{1:T}) = T^{-1}\sum_{t=1}^{T}\left\{\left[\ln f\right]'(x_t)\right\}^2 , \tag{A.18}$$

$$\mathcal{J}_T(x_{1:T}) = -T^{-1}\sum_{t=1}^{T}\left[\ln f\right]''(x_t) , \tag{A.19}$$

$$\mathcal{S}_T(x_{1:T}) = T^{-1/2}{\mathcal{I}_T}^{-1/2}(x_{1:T}) \sup_{t=1,\ldots,T}\left|\left[\ln f\right]'(x_t)\right| , \tag{A.20}$$

and $\mathcal{I}$ is as in (17). Lemma 9 is useful for checking Assumption (**B2**) and Lemma 10 for Assumptions (**C1-2**).

Lemma 9. *Assume (**A1**). Then (**B2**) is satisfied with F_T defined in* (A.17).

Lemma 10. *Let $A : \mathbb{R}^\ell \to \mathbb{R}$ be a bounded Lipschitz function. Let Γ be a $(\ell \times \ell)$ nonnegative symmetric matrix and $\{V_t = (V_t^1, \ldots, V_t^\ell)\}_{t=1}^T$ be i.i.d. ℓ-dimensional random vectors with zero-mean and covariance matrix Γ. For $i = 1, \ldots, \ell$, let $H^i : \mathbb{R}^2 \to \mathbb{R}$ be functions such that for all $x \in \mathbb{R}$, $y \mapsto H^i(x, y)$ is differentiable at $y = x$ and $H^i(x, x) = 0$. Finally, for $\mathbf{x} \in \mathbb{R}^{T+1}$ and $y \in \mathbb{R}$, let*

$$\Upsilon(\mathbf{x}, y) \triangleq \mathbb{E}\left[A\left\{\left(L_{1,T}(\mathbf{x}, V^i) + H^i(x_0, y)\right)_{i=1}^{\ell}\right\}\right] ,$$

where $L_{1,T}$ is the log-likelihood ratio defined in (15). *Then,*

(i) $\lim_{T\to\infty} \sup_{\mathsf{F}_T} |\Upsilon(\mathbf{x}, x_0) - a(A, \mathcal{I}\Gamma)| = 0$, *where $\mathcal{I}$ is defined in* (17) *and*

$$a(A, \Gamma) \triangleq \mathbb{E}\left[A\left\{\left(G^i - \mathbb{V}\mathrm{ar}[G^i]/2\right)_{i=1}^{\ell}\right\}\right] , \tag{A.21}$$

where $(G^1, \ldots, G^\ell) \sim N(0, \Gamma)$

(ii) If in addition A is differentiable and ∇A is a bounded Lipschitz function, then for all $\mathbf{x} \in \mathbb{R}^{T+1}$, the function $y \mapsto \Upsilon(\mathbf{x}, y)$ is differentiable at $y = x_0$ and

$$\lim_{T\to\infty} \sup_{\mathsf{F}_T} \left|\frac{\partial\Upsilon}{\partial y}(\mathbf{x}, x_0) - \left\langle \frac{\partial H}{\partial y}(x_0, x_0), a(\nabla A, \mathcal{I}\Gamma)\right\rangle\right| = 0 ,$$

where $\frac{\partial H}{\partial y}(x_0, y) = \left(\frac{\partial H^i}{\partial y}(x_0, y)\right)_{i=1}^{K}$.

Appendix B: Proof of Theorem 1

Denote by G^1 the generator of the Langevin diffusion $\mathrm{d}W[t] = \sqrt{\lambda_1}\mathrm{d}B[t] + \frac{1}{2}\lambda_1 \left[\ln f\right]'(W[t])\mathrm{d}t$, that is:

$$G^1 h(x) \triangleq \frac{\lambda_1}{2}\left(h'(x)\left[\ln f\right]'(x) + h''(x)\right) , \tag{B.1}$$

where λ_1 is defined in (18). Moreover, define $G_T = T(Q - I)$ where Q is the transition kernel of the Markov chain defined in Algorithm 2. To prove Theorem 1, we will apply Theorem 6. Under (**A1**), Assumption (**B1**) follows from standard properties of MCMC algorithms and (**B2**) is direct from Lemma 9. It remains to check (**B3-4**) for some $p > 1$. Considering Algorithm 2, G_T is decomposed into two terms: $G_T = G_T^1 + G_T^2$ where

$$G_T^1[Ph](\mathbf{x}) = T\int q^1(\mathbf{x};\mathbf{y}^1)\alpha_1(\mathbf{x};\mathbf{y}^1)\left[h(y_0^1) - h(x_0)\right]\mu(\mathrm{d}\mathbf{y}^1) \tag{B.2}$$

$$\begin{aligned} G_T^2[Ph](\mathbf{x}) = T\iint &\left[1 - \alpha_1(\mathbf{x};\mathbf{y}^1)\right]\alpha_2(\mathbf{x},\mathbf{y}^1;\mathbf{y}^2) \\ &q^1(\mathbf{x};\mathbf{y}^1)q^2(\mathbf{x},\mathbf{y}^1;\mathbf{y}^2)\left[h(y_0^2) - h(x_0)\right]\mu(\mathrm{d}\mathbf{y}^1)\mu(\mathrm{d}\mathbf{y}^2) \end{aligned} \tag{B.3}$$

where α_1 and α_2 are defined in (10) and (11), respectively. Note that $G_T^1 = T(Q^{RW} - I)$ where Q^{RW} is the transition kernel of a random walk Metropolis algorithm, that is, a particular case of the multiple correlated-try Metropolis (MTCM) algorithm with only one proposal. As shown in the proof of (Bédard, Douc and Moulines, 2012, Theorem 2), (**C1-4**) (and consequently (**B3-4**), according to Lemma 8) are satisfied with G_T replaced by G_T^1, and G replaced by G^1. To check (**B3-4**) with $G_T = G_T^1 + G_T^2$, it finally remains to show that for some $p > 1$,

$$\sup_{T\geq 0}\int \sup_{x_{1:T}\in\mathbb{R}^T} \left|G_T^2\left[Ph\right](\mathbf{x})\right|^p f(x_0)\mathrm{d}x_0 < \infty , \tag{B.4}$$

$$\lim_{T\to\infty}\int \sup_{x_{1:T}\in\mathsf{F}_T} \left|G_T^2\left[Ph\right](\mathbf{x})\right| f(x_0)\mathrm{d}x_0 = 0 . \tag{B.5}$$

The latter condition comes from the fact that

$$\begin{aligned} &\lim_{T\to\infty}\int \sup_{x_{1:T}\in\mathsf{F}_T} \left|G_T^1\left[Ph\right](\mathbf{x}) + G_T^2\left[Ph\right](\mathbf{x}) - G^1 h(x_0)\right| f(x_0)\mathrm{d}x_0 \\ &\leq \lim_{T\to\infty}\int \sup_{x_{1:T}\in\mathsf{F}_T} \left|G_T^1\left[Ph\right](\mathbf{x}) - G^1 h(x_0)\right| f(x_0)\mathrm{d}x_0 \\ &\quad + \lim_{T\to\infty}\int \sup_{x_{1:T}\in\mathsf{F}_T} \left|G_T^2\left[Ph\right](\mathbf{x})\right| f(x_0)\mathrm{d}x_0 . \end{aligned}$$

Since $[\ln f]$" is bounded, there exist $\beta, \gamma > 0$ such that for all $(x, y) \in \mathbb{R}^2$,

$$|\ln f(y) - \ln f(x)| \leq \beta(y-x)^2 + \gamma|y-x| \ .$$

This implies that

$$\begin{aligned}
&\left[1 - \alpha_1(\mathbf{x}; \mathbf{y}^1)\right] \alpha_2(\mathbf{x}, \mathbf{y}^1; \mathbf{y}^2) \\
&\leq 1 \wedge \frac{\pi(\mathbf{y}^2)}{\pi(\mathbf{x})} \frac{q^1(\mathbf{y}^1 - \mathbf{y}^2)}{q^1(\mathbf{y}^1 - \mathbf{x})} \\
&\leq 1 \wedge \exp\left(\sum_{s=0}^{T} \left[\frac{-(y_s^1 - y_s^2)^2 + (y_s^1 - x_s)^2}{2\ell_1^2/T} + \beta(y_s^2 - x_s)^2 + \gamma|y_s^2 - x_s| \right] \right) \ .
\end{aligned} \tag{B.6}$$

Now, define

$$S_T = \frac{1}{T} \sum_{s=0}^{T} \left(\frac{(U_s^2)^2 - 2U_s^1 U_s^2}{2\ell_1^2} - \beta \frac{(U_s^2)^2}{T} - \gamma \frac{|U_s^2|}{T^{1/2}} \right) \ .$$

Plugging Eq. (B.6) into the expression (B.3) of G_T^2 yields, for any $\eta > 0$,

$$\begin{aligned}
\left| G_T^2[Ph](\mathbf{x}) \right| &\leq \mathrm{osc}(h) T \mathbb{E}\left[1 \wedge \exp(-T S_T)\right] \\
&\leq \mathrm{osc}(h) T \left(\exp(-T\eta) + \mathbb{P}\left[S_T < \eta\right] \right) \ ,
\end{aligned} \tag{B.7}$$

where $\mathrm{osc}(h)$ is the oscillation of the function h, and $\mathrm{osc}(h) < \infty$ as h is bounded. Markov's inequality for i.i.d. random variables $(U_s^1, U_s^2)_{0 \leq s \leq T}$, where $(U_s^1, U_s^2) \sim \mathcal{N}\left(0, \begin{pmatrix} \ell_1^2 & 0 \\ 0 & \ell_2^2 \end{pmatrix}\right)$, implies that for any $\epsilon > 0$, there exists $C_1, C_2, C_\epsilon > 0$ such that for all $T \geq 1$,

$$\begin{aligned}
\mathbb{P}\left[|S_T - (\ell_2^2/2\ell_1^2)| > \epsilon \right] &\leq \frac{\mathbb{E}\left[|S_T - (\ell_2^2/2\ell_1^2)|^4 \right]}{T^4 \epsilon^4} \\
&\leq \frac{T C_1 + 3T(T-1) C_2}{T^4 \epsilon^4} \leq C_\epsilon T^{-2} \ .
\end{aligned}$$

This implies that for some sufficiently small $\eta > 0$, there exists a constant $C > 0$ (that may depend on η) such that for any $T \geq 1$,

$$\mathbb{P}\left[S_T < \eta\right] \leq C T^{-2} \ .$$

Finally,

$$\left| G_T^2[Ph](\mathbf{x}) \right| \leq \mathrm{osc}(h) T \left(\mathrm{e}^{-T\eta} + C T^{-2} \right) \ .$$

The right-hand side, which does not depend on $\mathbf{x}$, is bounded with respect to T (showing (B.4)) and converges to 0 as T tends to ∞ (showing (B.5)). The proof of Theorem 1 follows.

Appendix C: Proof of Theorem 4

The transition kernel Q of the Markov chain associated to Algorithm 4 clearly satisfies

$$Q\left[Ph\right](\mathbf{x}) - h(x_0) = \sum_{j=1}^{2} \mathbb{E}\left[\left\{h(x_0 + T^{-1/2}U_0^j[1]) - h(x_0)\right\}\beta^j(\mathbf{x}, T^{-1/2}, x_0 + T^{-1/2}U_0^j[1])\right]$$

with

$$\beta^1(\mathbf{x}, \eta, y_0) = \mathbb{E}\left[1 \wedge \frac{\pi([y_0, x_{1:T} + T^{-1/2}U_{1:T}[1]])}{\pi_T(x_{0:T})}\right]$$
$$\beta^2(\mathbf{x}, \eta, y_0) = \mathbb{E}\left[A_2\left\{\left(L_{1,T}(\mathbf{x}, V_{1:T}^i) + H^i(x_0, y_0)\right)_{i=1}^3\right\}\right]$$

where A_2 is defined in (25), $L_{1,T}$ is defined in (15) and

$$V_{1:T}^1 = U_{1:T}^1\,, \quad H^1(x,y) = \ln f(\bar{\Psi}_0(x,y)) - \ln f(x)\,,$$
$$V_{1:T}^2 = U_{1:T}^2\,, \quad H^2(x,y) = \ln f(y) - \ln f(x)\,,$$
$$V_{1:T}^3 = (1 - \ell_1\ell_2^{-1})U_{1:T}^2\,, \quad H^3(x,y) = \ln f(\bar{\Psi}_0(y,x)) - \ln f(x)\,.$$

Again, Assumptions (**B1-2**) follow from standard properties of MCMC algorithms, Assumption (**A1**), and Lemma 9. Thus, to apply Theorem 7, we only need to check that $\{\beta^i\}_{i=1}^2$ satisfies (**C1-4**). Note that β^1 is the acceptance probability of a classical RW-MH algorithm, which can be seen as a MCTM algorithm with a single proposal. Thus, as seen in the proof of (Bédard, Douc and Moulines, 2012, Theorem 2), Assumptions (**C1-4**) are satisfied with β^1.

It remains to check that β^2 satisfies (**C1-4**). This is an easy adaptation of the proof of (Bédard, Douc and Moulines, 2012, Theorem 2). Nevertheless, to provide a self-contained proof, we quickly repeat the main arguments. Since A_2 is Lipschitz and bounded, and since $H^i(x,x) = 0$ for $1 \le i \le 3$, Lemma 10 and the Dominated Convergence Theorem ensure that β^2 satisfies (**C1-2**). Noting that the first and second order derivatives of A are all bounded and the fact that there exists a constant M such that for all $u \in \mathbb{R}$,

$$|[\ln f]'(u)| \le M|u|\,, \quad |[\ln f]''(u)| \le M\,,$$

we obtain the existence of constants C and D (which do not depend on $\mathbf{x}$ nor on η and u) such that for all $\eta \le T^{-1/2} \le 1$,

$$\sup_{x_{1:T}\in\mathbb{R}^T}\left|\frac{\partial\tilde{\beta}^2}{\partial\eta}(\mathbf{x},\eta,u)\right| \le C|u|(|x_0| + |u|) + D\,,$$
$$\sup_{x_{1:T}\in\mathbb{R}^T}\left|\frac{\partial^2\tilde{\beta}^2}{\partial\eta^2}(\mathbf{x},\eta,u)\right| \le C|u|^2(|x_0| + |u|)^2 + D\,,$$

where $\tilde{\beta}^2$ is defined in (A.6). This proves Assumption (**C3**) for any $p > 1$. (**C4**) is immediate.

References

Bédard, M. (2007). Weak convergence of Metropolis algorithms for non-i.i.d. target distributions. *Ann. Appl. Probab.* **17** 1222–1244.

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

Bédard, M. and Rosenthal, J. S. (2008). Optimal scaling of Metropolis algorithms: heading toward general target distributions. *Canad. J. Statist.* **36** 483–503.

Beskos, A., Roberts, G. and Stuart, A. (2009). Optimal scalings for local Metropolis-Hastings chains on nonproduct targets in high dimensions. *Ann. Appl. Probab.* **19** 863–898.

Billingsley, P. (1999). *Convergence of probability measures*, Second ed. *Wiley Series in Probability and Statistics: Probability and Statistics*. John Wiley & Sons Inc., New York. A Wiley-Interscience Publication.

Craiu, R. V. and Lemieux, C. (2007). Acceleration of the Multiple-Try Metropolis algorithm using antithetic and stratified sampling. *Stat. Comput.* **17** 109–120.

Gelfand, S. B. and Mitter, S. (1991). Weak convergence of Markov chain sampling methods and annealing algorithms to diffusions. *J. Optim. Theory Appl.* **68** 483–498.

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

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

Harkness, M. A. and Green, P. J. (2000). Parallel chains, delayed rejection and reversible jump MCMC for object recognition. In *British Machine Vision Conference*.

Mattingly, J., Pillai, N. and Stuart, A. (2012). Diffusion limits of random walk Metropolis in high dimensions. *Ann. Appl. Probab.* **22** 881–930.

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

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

Raggi, D. (2005). Adaptive MCMC for inference on affine stochastic volatility models with jumps. *Econometrics Journal* **8** 235–250.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*, 2nd ed. Springer, New York.

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

Roberts, G. O. and Rosenthal, J. S. (1998). Markov chain Monte Carlo: Some practical implications of theoretical results. *Canad. J. Statist.* **26** 5–32.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statist. Sci.* **16** 351–367.

Tierney, L. and Mira, A. (1999). Some adaptive Monte Carlo methods for Bayesian inference. *Stat Med.* **18** 2507-2515.

Trias, M., Vecchio, A. and Veitch, J. (2009). Delayed rejection schemes for efficient Markov-Chain Monte-Carlo sampling of multimodal distributions. *ArXiv e-prints*.

Umstätter, R., Meyer, R., Dupuis, R., Veitch, J., Woan, G. and Christensen, N. (2004). Estimating the parameters of gravitational waves from neutron stars using an adaptive MCMC method. *Classical and Quantum Gravity* **21** 1655–1675.