

ACCELERATING METROPOLIS–HASTINGS ALGORITHMS BY DELAYED ACCEPTANCE

MARCO BANTERLE

Department of Medical Statistics
London School of Hygiene and Tropical Medicine
Keppel St, Bloomsbury, London WC1E 7HT, UK

CLARA GRAZIAN

Dipartimento di Economia
Università degli Studi “Gabriele D’Annunzio”
Viale Pindaro, 42, 65127 Pescara, Italy

ANTHONY LEE

School of Mathematics, University of Bristol
University Walk, Bristol BS8 1TW, UK

CHRISTIAN P. ROBERT*

Department of Statistics, University of Warwick
Gibbet Hill Road, Coventry CV4 7AL, UK

ABSTRACT. MCMC algorithms such as Metropolis–Hastings algorithms are slowed down by the computation of complex target distributions as exemplified by huge datasets. We offer a useful generalisation of the Delayed Acceptance approach, devised to reduce such computational costs by a simple and universal divide-and-conquer strategy. The generic acceleration stems from breaking the acceptance step into several parts, aiming at a major gain in computing time that out-ranks a corresponding reduction in acceptance probability. Each component is sequentially compared with a uniform variate, the first rejection terminating this iteration. We develop theoretical bounds for the variance of associated estimators against the standard Metropolis–Hastings and produce results on optimal scaling and general optimisation of the procedure.

1. Introduction. When running an MCMC sampler such as Metropolis–Hastings algorithms ([24]), the complexity of the target density required by the acceptance ratio may lead to severe slow-downs in the execution of the algorithm. A direct illustration of this difficulty is the simulation from a posterior distribution involving a large dataset of n points for which the computing time is at least of order $O(n)$. Several solutions to this issue have been proposed in the literature ([15, 20, 32, 42]), taking advantage of the likelihood decomposition

$$\prod_{i=1}^n \ell(\theta|x_i) \tag{1}$$

2010 *Mathematics Subject Classification.* Primary: 68U20, 65C40; Secondary: 62C10.

Key words and phrases. Large Scale Learning and Big Data, MCMC algorithms, likelihood function, acceptance probability, mixtures of distributions, Jeffreys prior.

* Corresponding author: Christian P. Robert.

to handle subsets of the data on different processors (CPU), graphical units (GPU), or even computers. However, there is no consensus on the method of choice, some leading to instabilities by removing most prior inputs and others to approximations delicate to evaluate or even to implement.

Our approach here is to delay acceptance (rather than rejection as in [41]) by sequentially comparing parts of the MCMC acceptance ratio to independent uniforms, in order to stop earlier the computation of the aforesaid ratio, namely as soon as one term is below the corresponding uniform.

More formally, consider a generic Metropolis–Hastings algorithm where the acceptance ratio $\pi(y)q(y,x)/\pi(x)q(x,y)$ is compared with a $\mathcal{U}(0,1)$ variate to decide whether or not the Markov chain switches from the current value x to the proposed value y ([24]). If we now decompose the ratio as an arbitrary product

$$\pi(y)q(y,x)/\pi(x)q(x,y) = \prod_{k=1}^d \rho_k(x,y) \quad (2)$$

where the only constraint is that the functions ρ_k are all positive and satisfy the balance condition $\rho_k(x,y) = \rho_k(y,x)^{-1}$ and then accept the move with probability

$$\prod_{k=1}^d \min \{ \rho_k(x,y), 1 \} , \quad (3)$$

i.e. by successively comparing uniform variates u_k to the terms $\rho_k(x,y)$, the motivation for our delayed approach is that the same target density $\pi(\cdot)$ is stationary for the resulting Markov chain.

The mathematical validation of this simple if surprising result can be seen as a consequence of [5]. This paper reexamines [10], where the idea of testing for acceptance using an approximation before computing the exact likelihood was first suggested. In [5], the original proposal density q is used to generate a value y that is tested against an approximate target $\tilde{\pi}$. If accepted, y can be seen as coming from a pseudo-proposal \tilde{q} that simply is formalising the earlier preliminary step and is then tested against the true target π . The validation in [5] follows from standard detailed balance arguments; we will focus formally on this point in Section 2.

In practice, sequentially comparing those probabilities with d uniform variates means that the comparisons stop at the first rejection, implying a gain in computing time if the most costly items in the product (2) are saved for the final comparisons.

Examples of the specific two-stage Delayed Acceptance as defined by [5] can be found in [13], in the pMCMC context, and in [37].

The major drawback of the scheme is that Delayed Acceptance *efficiently* reduces the computing cost only when the approximation $\tilde{\pi}$ is “good enough” or “flat enough”, since the probability of acceptance of a proposed value will always be smaller than in the original Metropolis–Hastings scheme. In other words, the original Metropolis–Hastings kernel dominates the new one in Peskun’s ([21]) sense. The most relevant question raised by [5] is thus how to achieve a proper approximation; note in fact that while in Bayesian statistics a decomposition of the target is always available, by breaking original data in subsamples and considering the corresponding likelihood parts or even by just separating the prior, proposal and likelihood ratio into different factors, these decompositions may just lead to a deterioration of the algorithm properties without impacting the computational efficiency.

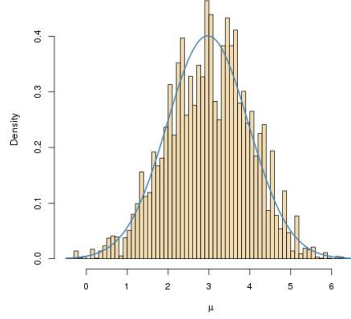


FIGURE 1. Fit of a two-step Metropolis–Hastings algorithm applied to a normal-normal posterior distribution $\mu|x \sim N(x/(\{1 + \sigma_\mu^{-2}\}), 1/\{1 + \sigma_\mu^{-2}\})$ when $x = 3$ and $\sigma_\mu = 10$, based on $T = 10^5$ iterations and a first acceptance step considering the likelihood ratio and a second acceptance step considering the prior ratio, resulting in an overall acceptance rate of 12%

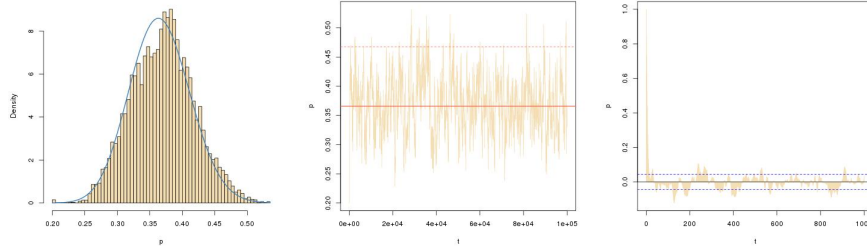


FIGURE 2. (left) Fit of a multiple-step Metropolis–Hastings algorithm applied to a Beta-binomial posterior distribution $p|x \sim Be(x + a, n + b - x)$ when $N = 100$, $x = 32$, $a = 7.5$ and $b = .5$. The binomial $\mathcal{B}(N, p)$ likelihood is replaced with a product of 100 Bernoulli terms and an acceptance step is considered for the ratio of each term. The histogram is based on 10^5 iterations, with an overall acceptance rate of 9%; (centre) raw sequence of successive values of p in the Markov chain simulated in the above experiment; (right) autocorrelogram of the above sequence.

However, even in these simple cases, it is possible to find examples where Delayed Acceptance may be profitable. Consider for instance resorting to a costly non-informative prior distribution (as illustrated in Section 5.3 in the case of mixtures); here the first acceptance step can be solely based on the ratio of the likelihoods and the second step, which involves the ratio of the priors, does not require to be computed when the first test leads to rejection. Even more often, the converse decomposition applies to complex or just costly likelihood functions, in that the prior ratio may first be used to eliminate values of the parameter that are too unlikely for the prior density. As shown in Figure 1, a standard normal-normal

example confirms that the true posterior and the histogram resulting from such a simulated sample are in agreement.

In more complex settings, as for example in “Big Data” settings where the likelihood is made of a very large number of terms, the above principle also applies to any factorisation of the like of (1) so that each individual likelihood factor can be evaluated separately. This approach increases both the variability of the evaluation and the potential for rejection, but, if each term of the factored likelihood is sufficiently costly to compute, the decomposition brings some improvement in execution time. The graphs in Figure 2 illustrate an implementation of this perspective in the Beta-binomial case, namely when the binomial $\mathcal{B}(N, p)$ observation $x = 32$ is replaced with a sequence of N Bernoulli observations. The fit is adequate on 10^5 iterations, but the autocorrelation in the sequence is very high (note that the ACF is for the 100 times thinned sequence) while the acceptance rate falls down to 9%. (When the original $y = 32$ observation is (artificially) divided into 10, 20, 50, and 100 parts, the acceptance rates are 0.29, 0.25, 0.12, and 0.09, respectively.) The gain in using this decomposition is only appearing when each Bernoulli likelihood computation becomes expensive enough.

On one hand, the order in which the product (3) is explored determines the computational efficiency of the scheme, while, on the other hand, it has no impact on the overall convergence of the resulting Markov chain, since the acceptance of a proposal does require computing all likelihood values. It therefore makes sense to try to optimise this order by ranking the entries in a way that improves the execution speed of the algorithm (see Section 3.2).

We also stress that the Delayed Acceptance principle remains valid even when the likelihood function or the prior are not integrable over the parameter space. Therefore, the prior may well be improper. For instance, when the prior distribution is constant, a two-stage acceptance scheme reverts to the original Metropolis–Hastings one.

Similar questions about efficiency of Delayed Acceptance schemes were considered in [35] and in [33], where the authors propose to build $\hat{\pi}$ using a weighted average of the previous evaluations of the likelihood computed at the k nearest neighbours in the parameter space; as the chain progresses, pairs of proposed parameters and their corresponding likelihood are stored in a KD -tree to efficiently search for the nearest neighbours and adaptively build the surrogate distribution.

Finally, while the Delayed Acceptance methodology is intended to cater to complex likelihoods or priors, it does not bring a solution *per se* to the “Big Data” problem in that (a) all terms in the product must eventually be computed; and (b) all terms previously computed (i.e., those computed for the last accepted value of the parameter) must be either stored for future comparison or recomputed. [7] combine Delayed Acceptance and the factorisation in (3) with control variate ideas to address this issue: provided each term in the decomposition can be bounded from below, the corresponding independent Bernoulli draws can be efficiently (sub-)sampled via thinning ([8]), avoiding thus the need to compute every ρ_k term. See instead [3], for a review on different parallel ways of handling massive datasets.

The plan of the paper is as follows: in Section 2, we validate the decomposition of the acceptance step into a sequence of decisions, arguing about the computational gains brought by this generic modification of Metropolis–Hastings algorithms and further analysing the relation between the proposed method and the Metropolis–Hastings algorithm in terms of convergence properties and asymptotic variances of

statistical estimates. In Section 4 we briefly state the relations between Delayed Acceptance and other methods present in the literature. In Section 3 we aim at giving some intuitions on how to improve the behaviour of Delayed Acceptance by ranking the factors in a given decomposition to achieve optimal computational efficiency and finally give some preliminary results in terms of optimal scaling for the proposed method. Then Section 5 studies Delayed Acceptance within three realistic environments, the first one made of logistic regression targets, the second one alleviating the computational burden from a Geometric Metropolis adjusted Langevin algorithm and a third one handling an original analysis of a parametric mixture model via genuine Jeffreys priors. Section 6 concludes the paper.

2. Validation and convergence of delayed acceptance. In this section, we establish that Delayed Acceptance is a valid Markov chain Monte Carlo scheme and analyse on a theoretical basis the differences with the original version.

2.1. The general scheme. We assume for simplicity that the target distribution π and the proposal distributions $Q(x, \cdot)$ all admit densities w.r.t. the Lebesgue or counting measures. We also denote by π the target density and let $q(x, y)$ denote the proposal density.

Let $(X_n)_{n \geq 1}$ be a Markov chain evolving on \mathbf{X} with Metropolis–Hastings Markov transition kernel P associated with q and π , i.e. for $A \in \mathcal{B}(\mathbf{X})$

$$P(x, A) := \int_A q(x, y) \alpha(x, y) dy + \left(1 - \int_{\mathbf{X}} q(x, y) \alpha(x, y) dy\right) \mathbf{1}_A(x),$$

where

$$\alpha(x, y) := 1 \wedge r(x, y), \quad r(x, y) := \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}.$$

Above, $\alpha(x, y)$ is known as the Metropolis–Hastings acceptance probability and $r(x, y)$ as the Metropolis–Hastings acceptance ratio.

We consider the class of “Delayed acceptance” Markov kernels associated with P , which are defined by factorisations of the function r as

$$r(x, y) = \prod_{k=1}^d \rho_k(x, y) \tag{4}$$

with all components in the product satisfying $\rho_k(x, y) = \rho_k(y, x)^{-1}$. The associated Delayed Acceptance Markov kernel is then defined as

$$\tilde{P}(x, A) := \int_A q(x, y) \tilde{\alpha}(x, y) dy + \left(1 - \int_{\mathbf{X}} q(x, y) \tilde{\alpha}(x, y) dy\right) \mathbf{1}_A(x),$$

where

$$\tilde{\alpha}(x, y) := \prod_{k=1}^d \{1 \wedge \rho_k(x, y)\}.$$

We will denote by $(\tilde{X}_n)_{n \geq 1}$ the Markov chain associated with \tilde{P} .

The order in which the sequence of functions ρ_k appears in the factorisation (4) is important for algorithmic specification, as can be seen in Algorithm 1. It means that $\rho_1(x, Y)$ is evaluated first, $\rho_2(x, Y)$ second, and so on until $\rho_d(x, Y) = r(x, Y) / \prod_{k=1}^{d-1} \rho_k(x, Y)$ which is last, with the motivation that “early rejection” can allow computational savings by avoiding the computation of the subsequent $\rho_k(x, Y)$.

Algorithm 1 Delayed AcceptanceTo sample from $\tilde{P}(x, \cdot)$:

1. Sample $Y \sim Q(x, \cdot)$.
2. For $k = 1, \dots, d$:
 - With probability $1 \wedge \rho_k(x, Y)$ continue, otherwise stop and output x .
3. Output Y .

2.2. Validation. The first lemma is a standard representation leading to the validation of the Delayed Acceptance Markov chain:

Lemma 2.1. *For any Markov chain with transition kernel Π of the form*

$$\Pi(x, A) = \int_A q(x, y)a(x, y)dy + \left(1 - \int_{\mathbf{X}} q(x, y)a(x, y)dy\right) \mathbf{1}_A(x),$$

and satisfying detailed balance, the function $a(\cdot)$ satisfies (for π -a.a. x, y)

$$\frac{a(x, y)}{a(y, x)} = r(x, y).$$

Proof. This follows immediately from the detailed balance condition

$$\pi(x)q(x, y)a(x, y) = \pi(y)q(y, x)a(y, x).$$

□

The Delayed Acceptance Markov chain $(\tilde{X}_n)_{n \geq 1}$ is then associated with the intended target:

Lemma 2.2. *$(\tilde{X}_n)_{n \geq 1}$ is a π -reversible Markov chain.*

Proof. From Lemma 2.1 it suffices to verify that $\tilde{\alpha}(x, y)/\tilde{\alpha}(y, x) = r(x, y)$. Indeed, we have

$$\begin{aligned} \frac{\tilde{\alpha}(x, y)}{\tilde{\alpha}(y, x)} &= \frac{\prod_{k=1}^d \{1 \wedge \rho_k(x, y)\}}{\prod_{k=1}^d \{1 \wedge \rho_k(y, x)\}} \\ &= \prod_{k=1}^d \frac{1 \wedge \rho_k(x, y)}{1 \wedge \rho_k(y, x)} \\ &= \prod_{k=1}^d \rho_k(x, y) = r(x, y), \end{aligned}$$

since $\rho_k(y, x) = \rho_k(x, y)^{-1}$ and $(1 \wedge a)/(1 \wedge a^{-1}) = a$ for $a \in \mathbb{R}_+$.

□

Remark 1. It is immediate to show that

$$\tilde{\alpha}(x, y) = \prod_{k=1}^d \{1 \wedge \rho_k(x, y)\} \leq 1 \wedge \prod_{k=1}^d \rho_k(x, y) = 1 \wedge r(x, y) = \alpha(x, y),$$

since $(1 \wedge a)(1 \wedge b) \leq (1 \wedge ab)$ for $a, b \in \mathbb{R}_+$.

2.3. Comparisons of the kernels P and \tilde{P} . Given a probability measure μ , let us denote

$$\begin{aligned}\mu(f) &:= \int_{\mathbf{E}} f(x) \mu(\mathrm{d}x), \quad L^2(\mathbf{E}, \mu) := \{f : \mu(f^2) < \infty\} \\ L_0^2(\mathbf{E}, \mu) &:= \{f \in L^2(\mathbf{E}, \mu) : \mu(f) = 0\}.\end{aligned}$$

For a generic Markov kernel $\Pi : \mathbf{E} \times \mathcal{B}(\mathbf{E})$ with unique invariant probability measure μ , we define

$$\mathrm{var}(f, \Pi) := \lim_{n \rightarrow \infty} \mathrm{var} \left(n^{-\frac{1}{2}} \sum_{i=1}^n [f(X_i) - \mu(f)] \right),$$

where $(X_n)_{n \geq 1}$ is a Markov chain with Markov kernel Π initialised with $X_1 \sim \mu$.

Remark 2. One can immediately conclude from the construction of \tilde{P} that $\mathrm{var}(f, P) \leq \mathrm{var}(f, \tilde{P})$ for any $f \in L^2(\mathbf{X}, \pi)$, using Peskun ordering ([21, 40]), since $\tilde{\alpha}(x, y) \leq \alpha(x, y)$ for any $(x, y) \in \mathbf{X}^2$.

For any $f \in L^2(\mathbf{E}, \mu)$ we define the Dirichlet form associated with a μ -reversible Markov kernel $\Pi : \mathbf{E} \times \mathcal{B}(\mathbf{E})$ as

$$\mathcal{E}_{\Pi}(f) := \frac{1}{2} \int \mu(\mathrm{d}x) \Pi(x, \mathrm{d}y) [f(x) - f(y)]^2.$$

The (right) spectral gap of a generic μ -reversible Markov kernel has the following variational representation

$$\mathrm{Gap}(\Pi) := \inf_{f \in L_0^2(\mathbf{E}, \mu)} \frac{\mathcal{E}_{\Pi}(f)}{\langle f, f \rangle_{\mu}}.$$

which leads to the following comparison lemma:

Lemma 2.3. (lemma 34, [1]) *Let Π_1 and Π_2 be μ -reversible Markov transition kernels of μ -irreducible and aperiodic Markov chains, and assume that there exists $\varrho > 0$ such that for any $f \in L_0^2(\mathbf{E}, \mu)$*

$$\mathcal{E}_{\Pi_2}(f) \geq \varrho \mathcal{E}_{\Pi_1}(f),$$

then

$$\mathrm{Gap}(\Pi_2) \geq \varrho \mathrm{Gap}(\Pi_1)$$

and

$$\mathrm{var}(f, \Pi_2) \leq (\varrho^{-1} - 1) \mathrm{var}_{\mu}(f) + \varrho^{-1} \mathrm{var}(f, \Pi_1) \quad f \in L_0^2(\mathbf{E}, \mu).$$

We will need the following condition in the sequel, which imposes a uniform lower bound on each $\rho_k(x, y)$ when $\alpha(x, y) = 1$:

Condition 1. *Defining $A := \{(x, y) \in \mathbf{X}^2 : r(x, y) \geq 1\}$, there exists a $c > 0$ such that*

$$\inf_{(x, y) \in A} \min_{k \in \{1, \dots, d\}} \rho_k(x, y) \geq c.$$

Intuitively, this condition ensures that when the acceptance probability $\alpha(x, y)$ is 1 then the acceptance probability $\tilde{\alpha}(x, y)$ is uniformly lower bounded by a constant. Reversibility then implies that $\tilde{\alpha}(x, y)$ is uniformly lower bounded by a constant multiple of $\alpha(x, y)$ for all $x, y \in \mathbf{X}$. Ultimately, this allows one to show, using Lemma 2.3, that \tilde{P} is not too different from P .

Proposition 1. *Assume Condition 1. Then Lemma 2.3 holds with $\Pi_1 = P$, $\Pi_2 = \tilde{P}$, $\mu = \pi$ and $\varrho = c^{d-1}$.*

Proof. Let $(x, y) \in A$, defined in Condition 1. Since $r(x, y) \geq 1$, we have $\alpha(x, y) = 1$. On the other hand, from Condition 1,

$$\begin{aligned}\tilde{\alpha}(x, y) &= \prod_{k=1}^d 1 \wedge \rho_k(x, y) \\ &= \prod_{k: \rho_k(x, y) < 1} \rho_k(x, y) \geq c^{|\{k: \rho_k(x, y) < 1\}|} \geq c^{d-1},\end{aligned}$$

since at least one $\rho_k(x, y) \geq 1$ whenever $r(x, y) \geq 1$.

From Lemma 2.1, when $(x, y) \in A$, we have

$$\tilde{\alpha}(y, x) = \tilde{\alpha}(x, y)/r(x, y) \geq c^{d-1}\alpha(x, y)/r(x, y) = c^{d-1}\alpha(y, x)$$

and thus $\tilde{\alpha}(x, y) \geq c^{d-1}\alpha(x, y)$ for any $(x, y) \in A$. Now consider $(x, y) \notin A$, which implies that $(y, x) \in A$. Then combining this result with Lemma 2.1, $\tilde{\alpha}(y, x) = \tilde{\alpha}(x, y)/r(x, y) \geq c^{d-1}\alpha(x, y)/r(x, y) = c^{d-1}\alpha(y, x)$. So $\tilde{\alpha}(x, y) \geq c^{d-1}\alpha(x, y)$ for any $(x, y) \in \mathbf{X}^2$.

It follows that

$$\begin{aligned}\mathcal{E}_{\tilde{P}}(f) &= \int_{\mathbf{X}} \pi(dx) \tilde{P}(x, dy) (f(x) - f(y))^2 \\ &= \int_{\mathbf{X}} \pi(dx) P(x, dy) \frac{\tilde{\alpha}(x, y)}{\alpha(x, y)} (f(x) - f(y))^2 \\ &\geq c^{d-1} \int_{\mathbf{X}} \pi(dx) P(x, dy) (f(x) - f(y))^2 = c^{d-1} \mathcal{E}_P(f),\end{aligned}$$

and we conclude. \square

The implication of this result is that, if P admits a right spectral gap, then so does \tilde{P} , whenever Condition 1 holds. Furthermore, and irrespective of whether or not P admits a right spectral gap, quantitative bounds on the asymptotic variance of MCMC estimates using $(\tilde{X}_n)_{n \geq 1}$ in relation to those using $(X_n)_{n \geq 1}$ are available.

2.4. Modification of a given factorisation. The easiest way to use the above result is to modify any candidate factorisation. Given a factorisation of the function r

$$r(x, y) = \prod_{k=1}^d \tilde{\rho}_k(x, y),$$

satisfying the balance condition, we can define a sequence of functions ρ_k such that both $r(x, y) = \prod_{k=1}^d \rho_k(x, y)$ and Condition 1 holds. To that effect, take an arbitrary $c \in (0, 1]$ and define $b := c^{\frac{1}{d-1}}$. Then, if we set

$$\rho_k(x, y) := \min \left\{ \frac{1}{b}, \max \{b, \tilde{\rho}_k(x, y)\} \right\}, \quad k \in \{1, \dots, d-1\},$$

it then follows that one must define

$$\rho_d(x, y) := \frac{r(x, y)}{\prod_{k=1}^{d-1} \rho_k(x, y)}.$$

From this modification, we deduce the following result:

Proposition 2. *Using this scheme, Lemma 2.3 holds with $\Pi_1 = P$, $\Pi_2 = \tilde{P}$, $\mu = \pi$ and $\varrho = c^2$.*

Proof. We note that $\inf_{(x,y) \in \mathbb{X}^2} \prod_{k=1}^{d-1} 1 \wedge \rho_k(x, y) \geq b^{d-1} = c$ and that

$$\rho_d(x, y) = \frac{r(x, y)}{\prod_{k=1}^{d-1} \rho_k(x, y)} \geq b^{d-1} r(x, y) = cr(x, y).$$

With $A := \{(x, y) \in \mathbb{X}^2 : r(x, y) \geq 1\}$, it follows that $\inf_{(x,y) \in A} \rho_d(x, y) \geq c$, and so $\inf_{(x,y) \in A} \tilde{\alpha}(x, y) \geq c^2$. We conclude along the same lines as in the proof of Proposition 1. \square

While this modification ensures that one can take $\varrho = c^2$ in Proposition 1, it is too general to suggest that using \tilde{P} can be more computationally efficient than using P when the cost of evaluating each ρ_k is taken into account. Indeed, Proposition 2 holds when the functions $\tilde{\rho}_k$ are chosen completely arbitrarily. Of course in practice, one should choose $\tilde{\rho}_k$ and hence ρ_k so that they are in some sense in agreement with r .

We will show in the next example that a certain class of $\tilde{\rho}_k$'s are beneficial, namely those which correspond to Metropolis-Hastings acceptance ratios with “flattened” surrogate target densities. On the other hand, it is far from difficult to come up with surrogate target densities for which unmodified use of $\tilde{\rho}_k$ can lead to disastrous performance.

2.5. Example: Unmodified surrogate targets. One common usage ([5]) of Delayed Acceptance is to substitute a surrogate target $\bar{\pi}$ for π in $\rho_1(x, y)$. We consider the case $d = 2$ and a random walk proposal to examine Condition 1 in this context. Here we have $q(x, y) = q(y, x)$ and so

$$\alpha(x, y) = 1 \wedge \frac{\pi(y)}{\pi(x)},$$

while

$$\rho_1(x, y) = 1 \wedge \frac{\bar{\pi}(y)}{\bar{\pi}(x)}, \quad \rho_2(x, y) = 1 \wedge \frac{\pi(y)\bar{\pi}(x)}{\pi(x)\bar{\pi}(y)}.$$

Considering $(x, y) \in A = \{(x, y) \in \mathbb{X}^2 : r(x, y) \geq 1\}$ we require $c > 0$ satisfying simultaneously

$$\frac{\bar{\pi}(y)}{\bar{\pi}(x)} \geq c, \quad \frac{\pi(y)\bar{\pi}(x)}{\pi(x)\bar{\pi}(y)} \geq c.$$

The first of these says that $\bar{\pi}(y)/\bar{\pi}(x)$ cannot be too small when $\pi(y) \geq \pi(x)$. The second says that $\bar{\pi}(y)/\bar{\pi}(x)$ should not be a large multiple of $\pi(y)/\pi(x)$. There are a large variety of choices of $\bar{\pi}$ that allow one to take $c = 1$. For example, $\bar{\pi}(x) = \pi(x) + C$ for some constant $C \geq 0$ and $\bar{\pi}(x) \propto \pi(x)^\beta$ for some $\beta \in [0, 1]$. Note that $\beta = 0$ corresponds to $\bar{\pi}$ being a constant function and $\beta = 1$ corresponds to $\bar{\pi} \propto \pi$. In between, one can think of $\bar{\pi}$ as being a flattened version of π .

2.6. Counter-example: Failure to reproduce geometric ergodicity. Consider the case $\pi(x) = \mathcal{N}(x; 0, 1)$ and $\bar{\pi}(x) = \mathcal{N}(x; 0, \sigma^2)$ with $Q(x, \cdot)$ a normal distribution with mean x and fixed variance for each $x \in \mathbb{R}$. Here we have

$$\rho_1(x, y) = \exp \left\{ \frac{(x-y)(x+y)}{2\sigma^2} \right\}, \quad \rho_2(x, y) = \exp \left\{ \frac{(1-\sigma^2)(y-x)(y+x)}{2\sigma^2} \right\}.$$

[17] showed that a random-walk Metropolis-Hastings chain for targets with super-exponential tails is geometrically ergodic. We now exploit this result to derive that, if $\sigma^2 < 1$, then the unmodified delayed acceptance Markov chain is not geometrically ergodic.

Proposition 3. *The unmodified Delayed Acceptance Markov chain using the factorisation into ρ_1 and ρ_2 as above is not geometrically ergodic when $\sigma < 1$.*

Intuitively, when x is large $P(x, (-\infty, x)) \approx \frac{1}{2}$ but $\lim_{x \rightarrow \infty} \tilde{P}(x, \{x\}) = 1$ because $\rho_1(x, y)$ takes on smaller and smaller values for $y > x$ and $\rho_2(x, y)$ takes on smaller and smaller values for $y < x$.

Proof. From [30, Theorem 5.1], it suffices to show that $\pi\text{-ess inf}_{x \in \mathbb{X}} \tilde{P}(x, \{x\}^{\mathbb{G}}) = 0$, i.e. that for any $\tau \in (0, 1)$ we can find $A \subseteq \mathbb{X}$ such that $\pi(A) > 0$ and $\sup_{x \in A} \tilde{P}(x, \{x\}^{\mathbb{G}}) \leq \tau$. Let $B_s(z)$ denote the ball of radius s around z . Given $\tau \in (0, 1)$, we define

$$r := \sup\{s > 0 : Q(x, B_s(x)) < \tau/3\},$$

and

$$A := \left\{x : x > \frac{r}{2} - \frac{\sigma^2 \log(\tau/3)}{r(1-\sigma^2)}\right\} \cap \left\{x : Q(x, \mathbb{R}_-) < \frac{\tau}{3}\right\}.$$

Then

$$\begin{aligned} \tilde{P}(x, \{x\}^{\mathbb{G}}) &= \tilde{P}(x, B_r(x) \setminus \{x\}) + \tilde{P}(x, B_r^{\mathbb{G}}(x)) \\ &\leq \frac{\tau}{3} + \int_{B_r^{\mathbb{G}}(x)} Q(x, dy) \tilde{\alpha}(x, y) \\ &\leq \frac{2\tau}{3} + \int_{B_r^{\mathbb{G}}(x) \cap \mathbb{R}_+} Q(x, dy) \tilde{\alpha}(x, y) \\ &\leq \frac{2\tau}{3} + \sup_{y \in B_r^{\mathbb{G}}(x) \cap \mathbb{R}_+} \tilde{\alpha}(x, y) \\ &= \frac{2\tau}{3} + \sup_{y \in B_r^{\mathbb{G}}(x) \cap \mathbb{R}_+} [1 \wedge \rho_1(x, y)] [1 \wedge \rho_2(x, y)]. \end{aligned}$$

Now let $x \in A$, $y \in B_r^{\mathbb{G}}(x) \cap \mathbb{R}_+$ and assume $y < x$. It follows that $\rho_2(x, y)$ attains its maximum when $y = x - r$ and therefore

$$\begin{aligned} \rho_2(x, y) &\leq \exp\left\{\frac{(1-\sigma^2)r(r-2x)}{2\sigma^2}\right\} \\ &\leq \exp\left\{\frac{(1-\sigma^2)r}{2\sigma^2} \left[\frac{2\sigma^2 \log(\tau/3)}{r(1-\sigma^2)}\right]\right\} = \frac{\tau}{3}. \end{aligned}$$

Similarly, let $x \in A$, $y \in B_r^{\mathbb{G}}(x) \cap \mathbb{R}_+$ and assume $y > x$. It follows that $\rho_1(x, y)$ attains its maximum when $y = x + r$ and therefore

$$\begin{aligned} \rho_1(x, y) &\leq \exp\left\{-\frac{r(2x+r)}{2\sigma^2}\right\} \\ &\leq \exp\left\{-\frac{r}{2\sigma^2} \left(2r - \frac{2\sigma^2 \log(\tau/3)}{r(1-\sigma^2)}\right)\right\} \\ &\leq \exp\left\{\frac{r}{2\sigma^2} \left(\frac{2\sigma^2 \log(\tau/3)}{r(1-\sigma^2)}\right)\right\} \leq \frac{\tau}{3}, \end{aligned}$$

since $\log(\tau/3) < 0$ and $\sigma^2 < 1$. Therefore,

$$\sup_{y \in B_r^{\mathbb{G}}(x) \cap \mathbb{R}_+} [1 \wedge \rho_1(x, y)] [1 \wedge \rho_2(x, y)] \leq \frac{\tau}{3}$$

so $\tilde{P}(x, \{x\}^{\mathbb{G}}) \leq \tau$ and we conclude. \square

The same argument can be made for much more general targets and proposals, albeit at the expense of brevity and clarity. We refrain from such a generalisation as our purpose here is to demonstrate that the DA chain may fail to inherit geometric ergodicity and that the simple proposed modification of the Delayed Acceptance kernel provided in Section 2.4 allows one to avoid this.

3. Optimisation. When considering Markov Chain Monte Carlo methods in practice, their efficiency as measured by mixing properties and computational cost is a fundamental issue. This section addresses both perspectives in connection with Delayed Acceptance. Section 3.1 examines the proposal distribution and derives its optimal scaling from standard random-walk Metropolis–Hastings theory. Then Section 3.2 covers the ranking of the factors ρ_i , which drives the total computational cost of the procedure.

3.1. Optimising the proposal mechanism. The explorative performances of a random-walk MCMC are strongly dependent on its proposal distribution. As exemplified in [26], finding the optimal scale parameter does lead to efficient ‘jumps’ in the state space and the overall acceptance rate of the chain is directly connected to the average jump distance and to the asymptotic variance of ergodic averages. This provides practitioners with an approach to ‘auto-tune’ the resulting random-walk MCMC algorithm. Extending this calibration to the Delayed Acceptance scheme is equally important, on its own ground towards finding a reasonable scaling for the proposal distribution and to avoid comparisons with the standard Metropolis–Hastings version.

The original framework of [26] is centered on estimating a collection of expected functionals, say g , where a plausible criterion for the performances of the MCMC is the minimisation of the stationary integrated auto-correlation time (ACT) of the Markov chain, defined as

$$\tau_g = 1 + 2 \sum_{i=1}^{\infty} \text{Cor}(g(X_0), g(X_i))$$

where the index g stresses the dependence on the considered functional, which is connected to the asymptotic variance through $\text{var}(P, g) = \tau_g \times \text{var}_{\pi}(g)$ whenever the chain is ϕ -irreducible, aperiodic, and reversible, $\text{var}_{\pi}(g)$ is finite and $g \in L^2(\pi)$.

Research on this optimisation focus on two main cases:

- Consider the limit in the dimension of the target distribution toward ∞ , where [26] gave conditions under which each marginal chain converges toward a Langevin diffusion. Maximising the speed of that diffusion, say $h(\ell)$ where ℓ is a parameter of the scale of the proposal, implies a minimisation of the ACT and also that τ is *free* from the dependence on the functional, defining thus an independent measure of efficiency for the algorithm;
- [34] focus on unimodal elliptically symmetric targets and show that a proxy for the ACT in finite dimensions is the Expected Square Jumping Distance (ESJD), defined as

$$\mathbb{E} [\|X' - X\|_{\beta}^2] = \mathbb{E} \left[\sum_{i=1}^d \frac{1}{\beta_i^2} (X'_i - X_i)^2 \right]$$

where X and X' are two successive points in the chain and $\|\cdot\|_{\beta}$ represent the norm on the principal axes of the ellipse rescaled by the coefficients β_i so that every direction contributes equally.

An interesting result in [34] is that, as $d \rightarrow \infty$, the ESJD on one marginal component of the chain converges with the same speed as the diffusion process described in [26]. These authors furthermore show the asymptotic result holds for rather small dimension, roughly starting from $d = 5$.

Moreover, when considering efficiency for Delayed Acceptance, which is a technique tailored on costly computations, we need to focus on the execution time of the algorithm as well. We then proceed to define our measure of efficiency as

$$\mathbf{Eff} := \text{ESJD} / \text{cost per iteration} \quad (5)$$

similarly to [36] for Pseudo-Marginal MCMC.

Due to the complex acceptance ratio in Delayed Acceptance, an extension of the previous results requires rather stringent assumptions, albeit providing a proper guideline in practice. Section 5 will further demonstrate optimality extends beyond those conditions. Note that our assumptions are quite standard in the literature on the subject.

(H1) We assume for simplicity's sake that the Delayed Acceptance procedure operates on two factors only, i.e., that $r(x, y) = \rho_1(x, y) \times \rho_2(x, y)$. The acceptance probability of the scheme is thus

$$\tilde{\alpha}(x, y) = \prod_{i=1}^2 (1 \wedge \rho_i(x, y)).$$

We also consider the ideal setting where a computationally cheap approximation $\tilde{f}(\cdot)$ is available for $\pi(\cdot)$ and precise enough so that $\rho_2(x, y) = r(x, y) / \rho_1(x, y) = \pi(y) / \pi(x) \times \tilde{f}(x) / \tilde{f}(y) = 1$.

(H2) We further assume that the target distribution satisfies (A1) and (A2) in [26], which are regularity conditions on π and its first and second derivatives, and that $\pi(x) = \prod_{i=1}^n f(x_i)$.

(H3) We consider a random walk proposal $y = x + \sqrt{\ell^2/d} Z$, where $Z \sim \mathcal{N}(0, I_d)$. Note that Gaussianity can be easily relaxed to distributions with finite fourth moment and similar results are available for more heavy-tailed distributions ([18]).

(H4) Finally we assume that the cost of computing $\tilde{f}(\cdot)$, say c , is proportional to the cost of computing $\pi(\cdot)$, named C , with $c = \delta C$.

Normalising costs by setting $C = 1$, the average total cost per iteration of the Delayed Acceptance chain is $\delta + \mathbb{E}[\tilde{\alpha}]$ and the efficiency of the proposed method under the above conditions is

$$\mathbf{Eff}(\delta, \ell) = \frac{ESJD}{\delta + \mathbb{E}[\tilde{\alpha}]}$$

Lemma 3.1. *Under the above conditions (H1–H4) on the target $\pi(x)$, on the proposal $q(x, y)$ and on the factorised acceptance probability $\tilde{\alpha}(x, y) = \prod_{i=1}^2 (1 \wedge \rho_i(x, y))$ we have that*

$$\tilde{\alpha}(x, y) = (1 \wedge \rho_1(x, y))$$

and that as $d \rightarrow \infty$

$$\mathbf{Eff}(\delta, \ell) = \frac{h(\ell)}{\delta + \mathbb{E}[\tilde{\alpha}]} = \frac{2\ell^2 \Phi(-\frac{\ell\sqrt{I}}{2})}{\delta + 2\Phi(-\frac{\ell\sqrt{I}}{2})}$$

$$a(\ell) = \mathbb{E}[\tilde{\alpha}] = 2\Phi\left(-\frac{\ell\sqrt{I}}{2}\right)$$

where $I := \mathbb{E}\left[\left(\frac{(\pi(x))'}{\pi(x)}\right)^2\right]$ as defined in [26].

Proof. It is easy to see that (H1) implies $\tilde{f}(\cdot) = \pi(\cdot)$ and so $\rho_1(x, y) = r(x, y)$. Moreover, by definition, $\rho_2(x, y) = r(x, y)/\rho_1(x, y) = 1$ and hence the second test is always accepted. The acceptance rate reduces then to just the ratio $\tilde{f}(y)/\tilde{f}(x) = \rho_1(x, y)$.

The second part of the lemma follows directly from Theorem 1.1 in [26]. \square

Let us stress that almost all assumptions in the above Lemma can be relaxed and that performances are robust against small deviances from those assumptions, as shown by the literature on standard Metropolis–Hastings. Obtaining analytical results without such conditions, while possible, requires however a considerable mathematical effort.

We now state the main practical implication of Lemma 3.1.

Proposition 4. *If the conditions of Lemma 3.1 holds, the optimal average acceptance rate $\alpha^*(\delta)$ is independent of I .*

Proof. Consider $\mathbf{Eff}(\delta, \ell)$ in terms of $(\delta, a(\ell))$:

$$a = g(\ell) = 2\Phi\left(-\frac{\ell\sqrt{I}}{2}\right) \quad ; \quad \ell = g^{-1}(a) = -\Phi^{-1}\left(\frac{a}{2}\right) \frac{2}{\sqrt{I}}$$

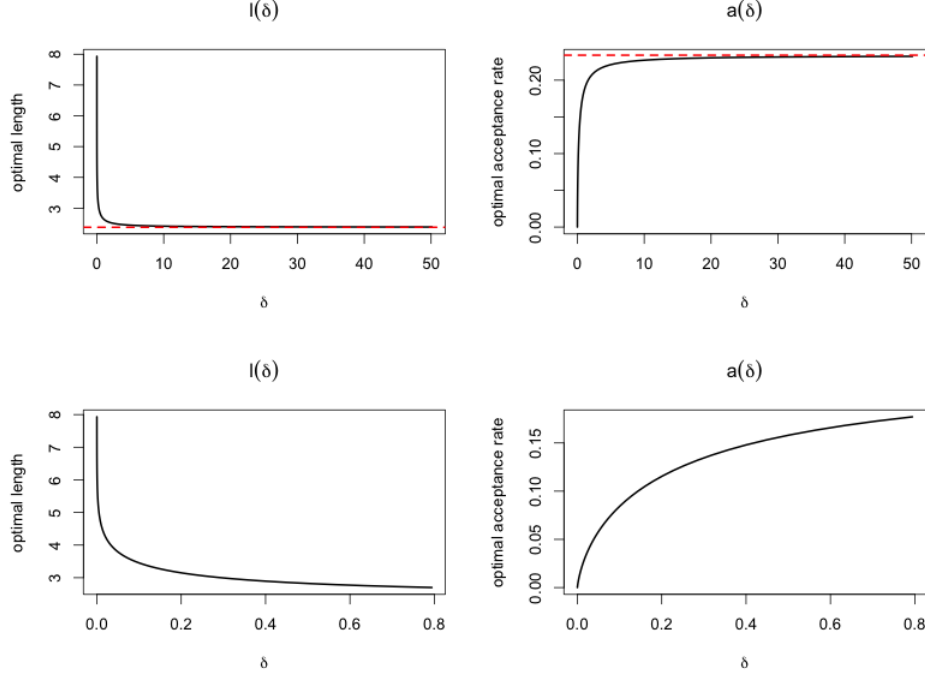
$$\mathbf{Eff}(\delta, a) = \frac{\frac{4}{I} \left[\Phi^{-1}\left(\frac{a}{2}\right)^2 a \right]}{\delta + a} = \frac{4}{I} \left\{ \frac{1}{\delta + a} \left[\Phi^{-1}\left(\frac{a}{2}\right)^2 a \right] \right\}$$

where we dropped the dependence on ℓ in a for notation's sake. It is now evident that to maximise $\mathbf{Eff}(\delta, a)$ in a we only need maximise $\left\{ \frac{1}{\delta + a} \left[\Phi^{-1}\left(\frac{a}{2}\right)^2 a \right] \right\}$, which is independent of I . \square

The optimal scale of the proposal $\ell^*(\delta)$ and the optimal acceptance rate $a^*(\delta)$ are thus given as functions of δ . In particular, as the relative cost of computing $\rho_1(x, y)$ with respect to $\rho_2(x, y)$ decreases, the proposed moves become bolder, in that ℓ^* increases and a^* decreases, since rejecting costs the algorithm little in terms of time, while every accepted move results in an almost independent sample. On the contrary when δ grows larger the chain rapidly approaches a Metropolis–Hastings behaviour, as it is no longer convenient to reject early. Figure 3 helps visualise the result.

3.2. Ranking the blocks. As mentioned at the end of Section 1, the order in which the factors $\rho_i(x, y)$ are tested has a strong influence on the performance of the algorithm. Delayed Acceptance was first developed in [10, 5] to speed up computations using a cheap approximation $\tilde{\pi}(\cdot)$ of the target distribution $\pi(\cdot)$ as a first step before computing the actual, and costly, Metropolis–Hastings ratio $\pi(y)/\pi(x)$ only in the cases where the acceptance test based on the approximation $\tilde{\pi}$ was satisfied. The main idea, namely to avoid the computation of the most costly parts as often as possible, remains relevant even for factorisations composed of more than two terms.

FIGURE 3. Two top panels: behaviour of $\ell^*(\delta)$ and $\alpha^*(\delta)$ as the relative cost varies. Note that for $\delta \gg 1$ the optimal values converges towards the values computed for the standard Metropolis–Hastings (dashed in red). Two bottom panels: close-up of the interesting region for $0 < \delta < 1$.



Consider an i.i.d. framework; the target (in x) is given by

$$\pi(x|Z) \propto p(x) \times L(Z|x) = p(x|Z) \times \prod_{i=1}^n f(z_i|x)$$

where $Z = (z_1, \dots, z_n)$ is an i.i.d. sample from $f(z|x)$ and $p(x)$ is the prior distribution for x , we can always consider the decomposition

$$r(x, y) = \prod_{i=1}^K \xi_i(x, y) \quad (6)$$

where each $\xi_i(x, y)$ is made of a small number of density ratio terms, with one including the prior and proposal ratios. In the limit, it is feasible if not necessarily efficient to consider the case $K = n + 1$ with

$$\xi_i(x, y) = \frac{f(z_i|y)}{f(z_i|x)} \quad i = 1, \dots, n \quad \text{and} \quad \xi_{n+1}(x, y) = \frac{p(y)q(y, x)}{p(x)q(x, y)}.$$

Assuming the computing cost is comparable for all terms, a solution for optimising the order of these factors ranks the entries according to the success rates observed so far, starting with the least successful values. Alternatively, the factorisation can start with the ratio that has the highest variance, since it is the most likely to be

rejected. (Note however that poor factorisations (6) lead to very low acceptance rates, as for instance when picking only outliers in a given group of observations.) Lastly, we can rank factors by their correlation with the full Metropolis–Hastings ratio; taking the argument to the limit, if the first factor has a perfect correlation with $r(\cdot, \cdot)$ then all the successive terms must be accepted and their order is hence of no interest.

This later setting is akin to considering the hypothetical optimal solution introduced in Section 3.1 with only two terms in the decomposition. Let a small number of the best scoring terms be merged to form ρ_1 and let the remaining factors become ρ_2 . $\rho_1(x, y) = \tilde{\pi}(y)/\tilde{\pi}(x)$ is then highly correlated to $r(x, y)$, $\rho_2(x, y) \sim 1$ for every (x, y) and hence $\tilde{\pi}(x)$ is a close approximation of the target, albeit probably flattened, which is exactly what we want (see Section 2).

As all these features can be evaluated for each subsample while running a chain with acceptance ratio factored as in (6), an implementation based on this intuition is then to take

$$\tilde{\pi}_{Z^*}(x) \propto p(x)^{m/n} \prod_{i=1}^m f(z_i^*|x) \quad \text{or} \quad \tilde{\pi}_{Z^*}(x) \propto \prod_{i=1}^m f(z_i^*|x)$$

with $m < n$, where $Z^* = (z_1^*, \dots, z_m^*)$ is a subsample of Z . At each iteration t of the Markov chain we compute all the $\xi_i(x^{(t-1)}, y)$ and $Z^{*(t)}$ is chosen as the subset that maximise the observed correlation between the values of $\frac{\tilde{\pi}_{Z^*}(x^{(j)})}{\tilde{\pi}_{Z^*}(x^{(t)})}$ ($j = 1, \dots, t-1$) and the full Metropolis–Hastings ratio (or whatever other selected criterion). As computing the real $\arg \max_{Z^* \subset Z}$ is expensive, in our practical implementation we resort to a *forward* selection scheme; starting with the factor ξ_i with the maximal correlation we build $Z^{*(t)}$ merging one term at a time until a desired correlation level is achieved, the observed correlation after including another term does not grow more than a small $\epsilon > 0$ or the size of $Z^{*(t)}$ has reached a critical point for computational purposes (e.g. 10% of the whole sample Z).

A relevant warning is that if we rearrange terms during the run, not only reordering but also merging them, in accordance to their correlation with the unmodified ratio, the resulting method has no theoretical guarantee since the kernel is potentially changing at each iteration depending on properties of previous samples ([11]).

As with standard adaptive MCMC ([28]) we resort thus to a *finite adaptation scheme*; we start with a fixed number of iterations to rank and rearrange the factors, followed by a fixed ordering to achieve ergodicity of the chain. We test this procedure in Section 5.1 on a simulated example.

Finally note that while we focused on the i.i.d. setting, in more complex cases where the ratio is factored and Delayed Acceptance can be applied, it is often the case that the optimal ordering of such factors is already known.

4. Relation with other methods.

4.1. Delayed acceptance and prefetching. Prefetching, as defined by [4], is a programming method that accelerates the convergence of a single MCMC chain by guessing future states in the path of a random walk Metropolis–Hastings Markov chain in order to use any additional computing power available, in the form of extra parallel processors, to calculate in advance necessary quantities (like the Metropolis–Hastings ratio) so that when the chain reaches a given state the computationally-heavy part of that iteration are ready.

Clearly the usefulness of this technique depends on our ability to guess the path of the chain correctly and hence many advanced prefetching strategies make use of the observed acceptance rate of the chain or even of a fast approximation $\tilde{\pi}$ of the target distribution to select the most likely future outcomes.

Since an in-depth exploration of prefetching is outside the scope of this work the reader is referred to [39] and citations therein for a complete discussion of the argument.

As mentioned above and demonstrated in [39, 2] if a cheap approximation $\tilde{\pi}$ of the target density is available, it can be used to select more likely future paths of the chain and this results in an efficient prefetching algorithm.

In our case the master process sequentially samples from the (Delayed Acceptance) chain by checking only the (assumed) inexpensive first approximation $\rho_1(x, y) = \tilde{\pi}(y)/\tilde{\pi}(x)$ while the other additional processors provide him the more expensive $\rho_2(x, y) = \pi(y)\tilde{\pi}(x)/\pi(x)\tilde{\pi}(y)$ computed beforehand thanks to prefetching. The theoretical properties of the chain are unchanged while the achievable speed-up may be substantial, especially for the first few additional processors.

4.2. Alternative procedure for delayed acceptance. In the case that every factor $\rho_i(x, y)$ has roughly the same computational cost, Philip Nutzman suggested (personal communication) that Delayed Acceptance can be slightly modified by taking the overall acceptance probability

$$\prod_{i=1}^d \min \{ \rho_i(x, y), 1 \} \quad \text{to be instead} \quad \min_{k=1, \dots, d} \left\{ \prod_{i=1}^k \rho_i(x, y), 1 \right\}.$$

Such a decomposition follows from the same idea that one would like to compute as few factors as possible once one realizes that the proposal is likely to be rejected. Under this modification the associated Markov chain still achieves the correct target in the stationary regime and the procedure satisfies detailed balance, provided the ordering of the terms is uniformly random.

An interesting consequence of this modification is that, as the number of factor increases, the acceptance rate eventually stabilises, while for the method described in Section 1 the acceptance rate decreases to zero. This property is indeed appealing, even though this procedure logically takes longer to complete when compared with the standard Delayed Acceptance (albeit less than the reference Metropolis–Hastings procedure).

The evident disadvantage of the modification in a general setting is that detailed balance implies that the factors are computed in a random order at each iteration, making vain any attempt to adapt in terms of the ordering (Section 3.2) or to set the order based on respective computational costs.

This drawback can be somewhat reduced by combining the above two approaches; consider the decomposition

$$\pi(y) q(y, x) / \pi(x) q(x, y) = \left[\prod_{i=1}^{d_1} \rho_i(x, y) \right] \times \left[\prod_{j=1}^{d_2} \phi_j(x, y) \right]$$

where $d_1 + d_2 = d$ and the factors ρ_i and ϕ_j represent respectively cheap factors and costly factors. By taking now

$$\min_{(m=1, \dots, d_1)} \left\{ 1, \prod_{i=1}^m \rho_i(x, y) \right\} \times \min_{(k=1, \dots, d_2)} \left\{ 1, \prod_{j=1}^k \phi_j(x, y) \right\}$$

the algorithm computes cheap factors first and expensive factors last, applying the symmetry requirement to satisfy detail balance inside each of both subsets. Clearly the above can be generalised to a larger number of subsets, each with d_i factors in it. Intuitively, this last modification can be explained as an early rejection of each of the intermediate acceptance/rejection steps inside a Delayed Acceptance scheme.

Remark 3. Interestingly if $d_l = 1 \forall l$ (l being the number of subsets considered) this procedure reduces to Delayed Acceptance, and for l that increases and $d_l > 1 \forall l$ this combined technique will have a even lower overall acceptance rate than standard Delayed Acceptance.

4.3. Delayed acceptance and slice sampling. As a final remark, we stress another analogy between our Delayed Acceptance algorithm and slice sampling ([19, 24]). Based on the same decomposition (1), slice sampling proceeds as follows

1. simulate $u_1, \dots, u_n \sim \mathcal{U}(0, 1)$ and set $\lambda_i = u_i \ell(\theta | x_i)$ ($i = 1, \dots, n$);
2. simulate θ' as a uniform under the constraints $\ell_i(\theta' | x_i) \geq \lambda_i$ ($i = 1, \dots, n$).

to compare with Delayed Acceptance which conversely

1. simulate $\theta' \sim q(\theta' | \theta)$;
2. simulate $u_1, \dots, u_n \sim \mathcal{U}(0, 1)$ and set $\lambda_i = u_i \ell(\theta' | x_i)$ ($i = 1, \dots, n$);
3. check that $\ell_i(\theta' | x_i) \geq \lambda_i$ ($i = 1, \dots, n$).

The differences between both schemes are thus that (a) slice sampling always accepts a move, (b) slice sampling requires the simulation of θ' under the constraints, which may prove infeasible, and (c) Delayed Acceptance re-simulates the uniform variates in the event of a rejection. In this respect, Delayed Acceptance appears as a “poor man’s” slice sampler in that values of θ' s are proposed until one is accepted.

5. Examples. To illustrate the improvement brought by Delayed Acceptance, we study three different realistic settings that reflect on the generality of the method. First, in Section 5.1 we consider a Bayesian analysis of a logistic regression model, to assess the computational gain brought by our approach in a “Big-Data” environment where obtaining the likelihood is the main computational burden. Secondly (Section 5.2) we examine a high dimensional toy Normal-Normal model, sample with a geometric Metropolis adjusted Langevin algorithm where the main computational cost comes from the proposal distribution which is position specific and involves derivatives of the density up till the third level, which are computed numerically at each iteration. Finally in Section 5.3 we investigate a mixture model where a formal Jeffreys prior is used, as it is not available in closed-form and does require an expensive approximation by numerical or Monte Carlo means. The latter example comes as a realistic setting where the prior itself is a burdensome object, even for small datasets.

5.1. Logistic regression. While a simple model, or due to its simplicity, logistic regression is widely used in applied statistics, especially in classification problems. The challenge in the Bayesian analysis of this model is not generic, since simple Markov Chain Monte Carlo techniques providing satisfactory approximations, but stems from the data-size itself. This explains why this model is used as a benchmark in some of the earlier accelerating papers ([15, 20, 32, 42]). Indeed, in “Big Data” setups, MCMC is deemed to be progressively inefficient and researchers are striving to keep simulation effective, focusing mainly on parallel computing and on sub-sampling but also on replacing the classic Metropolis scheme itself.

We tested the proposed method against the standard Metropolis–Hastings algorithm on 10^6 simulated data with a 100-dimensional parameter space. The proposal distribution is Gaussian: $y|x \sim \mathcal{N}(x, \Sigma)$ with Σ initialised to be $0.2 \times I_d$ (d being the dimension of the parameter space) and then adapted. The Metropolis–Hastings benchmark was made adaptive by targeting the asymptotic optimal acceptance rate of $\alpha^* = 0.234$ ([26]).

Delayed Acceptance was optimised first against the ordering of the factors as explained in Section 3; we split the data into subsamples of 10 elements and computed their empirical correlation with the full Metropolis–Hastings ratio as a criterion. Once these estimates were stable we merged into the surrogate target \tilde{f} the smallest number of subsamples needed to achieve a ≥ 0.85 correlation with $r(x, y)$. As soon as the ordering was fixed we computed δ , the relative cost of the obtained ρ_1 with respect to the full ratio, and run the chain for the remaining iterations optimising Σ against the optimal acceptance rate found through (5). We also added the modification explained in Section 2.4 with c set such that b was slightly lower than the optimal acceptance rate above.

Algorithm	rel. ESS (av.)	rel. ESJD (av.)	rel. Time (av.)
DA-MH over MH	1.1066	12.962	0.098

Algorithm	rel. gain (ESS) (av.)	rel. gain (ESJD) (av.)
DA-MH over MH	5.47	56.18

TABLE 1. Comparison between MH and MH with Delayed Acceptance on a logistic model. **ESS** is the effective sample size, **ESJD** the expected square jumping distance, **time** is the computation time.

We collected 100 repetitions of the experiment and the results are presented in Table 1. Before commenting the results we highlight the fact that this situation may seem not particularly appealing for Delayed Acceptance and in fact straight application of the method by randomly choosing the composition of ρ_1 and ρ_2 may lead to variable results. Further coding effort is required here in order to choose adaptively how to split the MH ratio. Borrowing from both Section 3.2 and the end of Section 3.1, i.e. by choosing during the brief burn-in of the chain which subset best represents the whole likelihood and then, based on how populated that subset is, targeting a specific acceptance ratio, produces both a completely automated MCMC version for this kind of data (*iid*) and better results under a time constraint.

As shown in Table 1, while the assumption made in Section 3 not completely satisfied, the relative efficiency of Delayed Acceptance is higher than for MH by a factor of almost 6. We measured efficiency through *effective sample size* (ESS, from the **coda** R package, [22]) or *expected square jumping distance* (ESJD). By choosing the first subsample to be *informative* on the whole ratio there is practically no loss on ESS (while the estimated ESJD actually increased) and, given the significantly reduced acceptance rate, the computing time is usually less than a fourth of the computing time of the corresponding optimal MH, taking into account the first part of chain used to determine the blocks ranking.

5.2. G-MALA with delayed acceptance.

5.2.1. *MALA and Geometric MALA*:. Random walk Metropolis–Hastings, while generic and popular, can struggle with posterior distributions in high dimensions or in the presence of high correlation between some components. In such cases it is inefficient, with low acceptance rate, poor mixing and highly correlated samples. Metropolis adjusted Langevin algorithm (MALA, see for instance [29]) has been devised to overcome these difficulties by taking advantage of the gradient of the target distribution in the proposal mechanism, making the Markov chain more robust with respect to the dimension of the problem and proposing broader moves with higher probability. MALA is based on a Langevin diffusion, with the target (the posterior distribution $\pi(\theta|y)$ in our case) as a stationary distribution, defined by the SDE

$$\frac{d\theta}{dt} = \frac{1}{2} \nabla_{\theta} \log(\pi(\theta|y)) + \frac{dB}{dt}$$

where B is a Brownian motion. Using a first-order discretisation the diffusion gives the following proposal mechanism:

$$\theta' = \theta^{(i-1)} + \varepsilon^2 \nabla_{\theta} \log(\pi(\theta^{(i-1)}|y))/2 + \varepsilon Z$$

where ε is the step-size for the Euler’s integration and $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This discretisation is then compensated by introducing an accept/reject probability similar to a Metropolis–Hastings algorithm.

This diffusion is isotropic and will hence still be inefficient for highly correlated components or with very different scales, as the step size ε is fixed across dimensions. [29] propose to alleviate the issue using a pre-conditioning matrix A so that the proposal becomes

$$\theta' = \theta^{(i-1)} + \varepsilon^2 A^T A \nabla_{\theta} \log(\pi(\theta^{(i-1)}|y))/2 + \varepsilon AZ.$$

[6] demonstrate however that defining this matrix in general can be difficult and that tuning on the go may result in an inappropriate asymptotic behaviour.

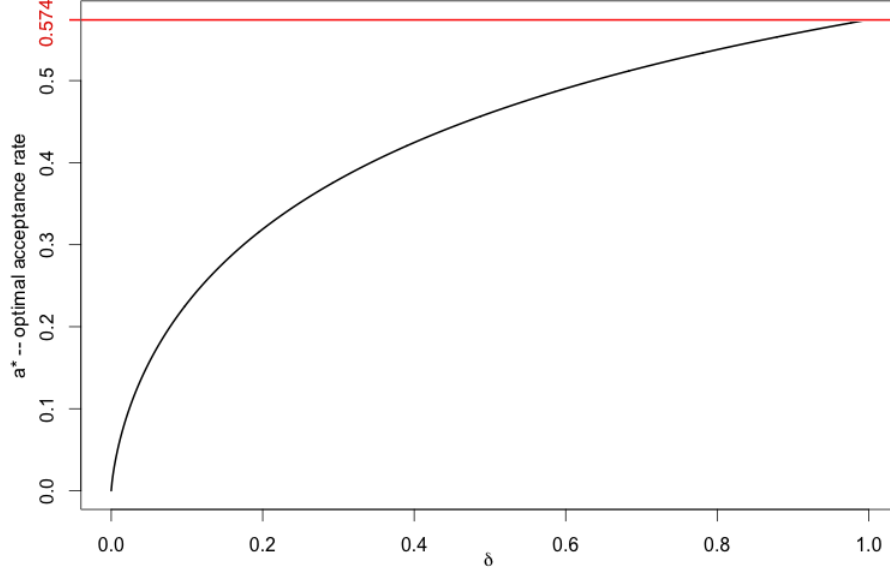
[12] propose the Geometric-MALA in order to overcome this difficulty, advising the use of a position specific metric for the matrix A , which takes advantage of the geometry of the target space that the chain is exploring. They suggest in particular the Fisher-Rao metric tensor. In terms of Bayesian inference, where the target distribution is the posterior density, this choice translates into $A^T A$ being the expected Fisher information matrix plus the negative Hessian of the log-prior.

This theoretically efficient solution also performs well in practice but comes with a serious computational burden in the fact that at every evaluation of the Metropolis–Hastings ratio derivatives up till the third order of our log-target distribution are needed and, in the event of them being analytically not available, expensive numerical approximations are to be computed (see equation (10) of [12]).

5.2.2. *Sampling with Delayed Acceptance and GMALA*:. Geometric-MALA represent a perfect application for Delayed Acceptance since we can naturally divide its acceptance ratio into the product of the posterior ratio and the ratio of the proposals, the latter to be only computed when the proposed point is associated with a relatively large posterior probability.

As described above, the computational bottleneck of the G-MALA lays in the computation of the third derivative of our log-target at the proposed point, while the computation of the posterior itself has usually a low relative cost. Moreover even with a non-symmetric efficient proposal mechanism (the discretised Langevin

FIGURE 4. Optimal acceptance rate for the DA-MALA algorithm as a function of δ . In red, the optimal acceptance rate for MALA obtained by [27] is met for $\delta = 1$.



diffusion) G-MALA is still close to a random walk and we expect the ratio of the proposal to be near 1, especially at equilibrium especially when ε is small. Therefore, the first ratio is inexpensive, relative to the second one, while the decision reached at the first stage should be consistent with the overall acceptance rate.

Given that optimal scaling for MALA in terms of the dimension d of the target differs from the random-walk setting (see [27]), we set the variance of the random-walk normal component as $\sigma_d^2 = \frac{\ell^2}{d^{1/3}}$. Borrowing from Section 3.1, we can obtain the optimal acceptance rate for the DA-MALA, through Equation (5), by maximising

$$\mathbf{Eff}(\delta, \ell) = \frac{h(\ell)}{\delta + \mathbb{E}[\tilde{\alpha}] - \delta \times \mathbb{E}[\tilde{\alpha}]} = \frac{2\ell^2 \Phi(-\frac{K\ell^3}{2})}{\delta + 2\Phi(-\frac{K\ell^3}{2}) \times (1 - \delta)}$$

or equivalently

$$\mathbf{Eff}(\delta, a) = - \left(\frac{2}{K} \right)^{\frac{2}{3}} \left[\frac{a \Phi^{-1} \left(\frac{a}{2} \right)^{\frac{2}{3}}}{\delta + a(1 - \delta)} \right].$$

In the above the computational cost per iteration is taken to be $c = \delta C$ for the posterior ratio, $C = 1$ for the proposal ratio (and hence $c + \mathbb{E}[\tilde{\alpha}](C - c)$ for the whole kernel), $h(\ell)$ is again the speed of the limiting diffusion process and K is a measure of roughness of the target distribution, depending on its derivatives. Since the optimal a^* is independent from K , we do not define it more rigorously, referring to [27]. Figure 4 shows that a^* decreases with δ , as is the case with random-walk Metropolis–Hastings. It reaches the known optimum for the standard MALA when $\delta = 1$.

5.2.3. *Simulation study:* To test the above assumptions we ran a toy MALA example where we drew 100 samples from a $\mathcal{N}_d(\theta, I)$, with $d = 10$; $\pi(\theta)$ was set to be

Algorithm	ESS (av.)	(sd)	ESJD (av.)	(sd)	time (av.)	(sd)
MALA	7504.48	107.21	5244.94	983.47	176078	1562.3
DA-MALA	6081.02	121.42	5373.253	2148.76	17342.91	6688.3

Algorithm	a (aver.)	ESS/time (aver.)	ESJD/time (aver.)
MALA	0.661	0.04	0.03
DA-MALA	0.09	0.35	0.31

TABLE 2. Comparison between standard geometric MALA and geometric MALA with Delayed Acceptance, with **ESS** the effective sample size, **ESJD** the expected square jumping distance, **time** the computation time and **a** the observed acceptance rate.

$\mathcal{N}_d(0, 100)$. Figure 5 presents an example run. We then repeated the experiment 100 times and computed an average efficiency gain, defined either as ESS or as the ESJD, over the computing time. We computed δ at each run by averaging a few computed derivatives, required by the proposal ratio. We then adapt ε to get the optimal acceptance rate, being conservative in order to avoid overflow issues with the first-order numerical integrator. Results are presented in Table 2. Delayed Acceptance exhibits improvement by a factor of 10 in this example, obtained almost for free in terms of coding time.

5.2.4. *HMC with Delayed Acceptance:* As a side note, while the reasoning applied to MALA does theory apply to Hamiltonian Monte Carlo (HMC), the computational gain obtained through Delayed Acceptance is only connected with avoiding some proposal computations. In a general HMC though (with both point-dependent and independent pre-conditioning matrices), proposing a new value still involves the computation of $L - 1$ (with L the number of steps in the discretised-Hamiltonian integration) derivatives, as only the starting point is recovered from the previous iteration, while computing the final Metropolis-Hastings ratio involves just the extra computation at the end point. Therefore, in this setting, the computational gain is much reduced.

5.3. Mixture model.

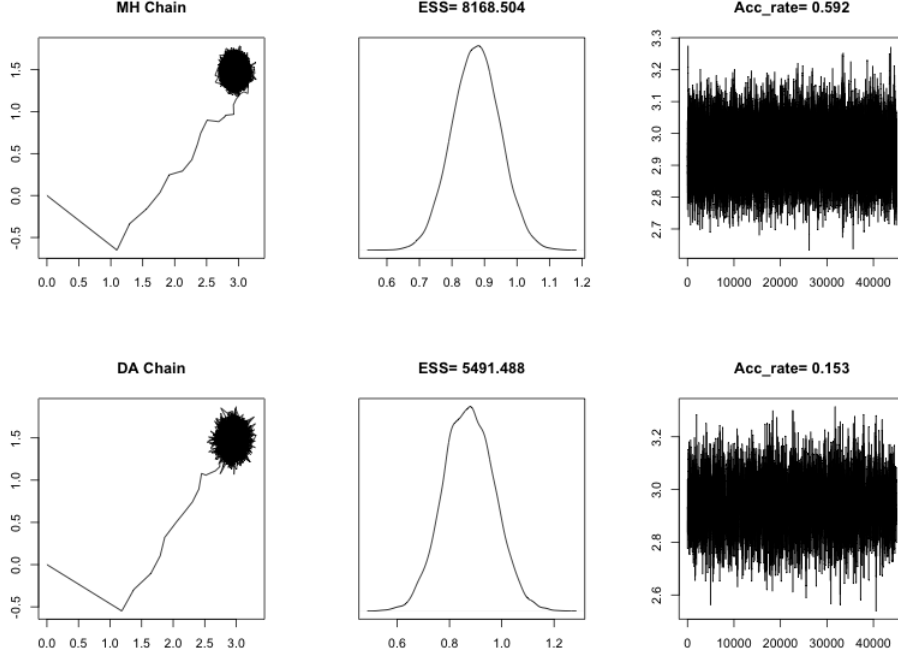
5.3.1. *Non-Informative inference on a Mixture Model:* Consider a standard mixture model ([16]) with a fixed number of components

$$\sum_{i=1}^k w_i f(x|\theta_i), \quad \text{with} \quad \sum_{i=1}^k w_i = 1. \quad (7)$$

This standard setting nonetheless offers a computational challenge in that the reference objective Bayesian approach based on the Fisher information and the associated Jeffreys prior ([14, 23]) is not readily available for computational reasons and has thus not been implemented so far. Proxies using Jeffreys priors on the components of (7) have been proposed instead, with the drawback that since they always lead to improper posteriors, ad hoc corrections have to be implemented ([9, 31, 38]).

When relying instead on dependent improper priors, it is not always the case that the posterior distribution is improper. For instance, [25] provide a location-scale representation that allows for some improper prior. In the current paper, we

FIGURE 5. Comparison between geometric MALA (top panels) and geometric MALA with Delayed Acceptance (bottom panels): marginal chains for two arbitrary components (left), estimated marginal posterior density for an arbitrary component (middle), 1D chain trace evaluating mixing (right).



consider instead the genuine Jeffreys prior for the complete set of parameters in (7), derived from the Fisher information matrix for the whole model. While establishing the analytical properness of the associated posterior is beyond the goal of the current paper, we handle large enough samples to posit that a sufficient number of observations is allocated to each component and hence the likelihood function dominates the prior distribution. (In the event the posterior remains improper, the associated MCMC algorithm should exhibit a transient behaviour.)

Therefore, this is an appropriate and realistic example for evaluating Delayed Acceptance since the computation of the prior density is clearly costly, relying on many integrals of the form:

$$-\int_{\mathcal{X}} \frac{\partial^2 \log \left[\sum_{i=1}^k w_i f(x|\theta_i) \right]}{\partial \theta_h \partial \theta_j} \left[\sum_{i=1}^k w_i f(x|\theta_i) \right] dx. \quad (8)$$

Indeed, these integrals cannot be computed analytically and thus their derivation involve numerical or Monte Carlo integration. This setting is such that the prior ratio—as opposed to the more common case of the likelihood ratio—is the costly part of the target evaluated in the Metropolis–Hastings acceptance ratio. Moreover, since the Jeffreys prior involves a determinant, there is no easy way to split the computation in more parts than “prior \times likelihood”. Hence, the Delayed Acceptance algorithm can be applied by simply splitting between the prior $p^J(\psi)$ and the likelihood $\ell(\psi|x)$ ratios, the latter being computed first. Moreover, since the

Algorithm	ESS (av.)	(sd)	ESJD (av.)	(sd)	time (av.)	(sd)
MH	1575.96	245.96	0.226	0.44	513.95	57.81
MH + DA	628.77	87.86	0.215	0.45	42.22	22.95

TABLE 3. Comparison using different performance indicators in the example of mixture estimation, based on 100 replicas of the experiments according to model (9) with a sample size $n = 500$, 10^5 MH simulations and 500 samples for the prior estimation. (“ESS” is the effective sample size, “time” is the computational time). The actual averaged gain ($\frac{ESS_{DA}/ESS_{MH}}{time_{DA}/time_{MH}}$) is 9.58, higher than the “double average” that the table above suggests as being around 5.

proposed prior is “non informative”, its influence on the definition of the posterior distribution should be small with respect to the likelihood function and, then, computing the likelihood ratio first should not have a substantial impact on the acceptance rate. However, the improper nature of the prior means using a second acceptance ratio solely based on the prior can create trapping states in practice, even though the method remains theoretically valid. We therefore opted for stabilising this second step by saving a small fraction of the likelihood, corresponding to 5% of the sample, to regularise this second acceptance ratio. This choice translates into Algorithm 2.

Algorithm 2 Metropolis–Hastings with Delayed Acceptance for Mixture Models

Set $\ell_2(\cdot|x) = \prod_{i=1}^{\lfloor pn \rfloor} \ell(\cdot|x_i)$ and $\ell_1(\cdot|x) = \prod_{i=\lfloor pn \rfloor+1}^n \ell(\cdot|x_i)$ where $p \in (0, 1)$

1. Simulate $\psi' \sim q(\psi'|\psi)$;
 2. Simulate $u_1, u_2 \sim \mathcal{U}(0, 1)$ and set $\lambda_1 = u_1 \ell_1(\psi|x)$;
 3. **if** $\ell_1(\psi'|x) \leq \lambda_1$, repeat the current parameter value and return to 1;
 else set $\lambda_2 = u_2 \ell_2(\psi|x) p^J(\psi)$;
 4. **if** $\ell_2(\psi'|x) p^J(\psi') \geq \lambda_2$ accept ψ' ;
 else repeat the current parameter value and return to 1.
-

5.3.2. *Simulation study:* An experiment comparing a standard Metropolis–Hastings implementation with a Metropolis–Hastings version relying on Delayed Acceptance is summarised in Table 3. Data were simulated from the following Gaussian mixture model:

$$f(y|\theta) = 0.10\mathcal{N}(-10, 2) + 0.65\mathcal{N}(0, 5) + 0.25\mathcal{N}(15, 7). \quad (9)$$

Both the standard Metropolis–Hastings and the Delayed Acceptance version are adapted against their respective optimal acceptance rate, which is computed to be 2%, given that δ is empirically established to be 0.01 using 500 samples for the Monte Carlo estimation of the prior. As a consequence the MH+DA algorithm will produce less unique samples in the total 10^5 iterations of the chain, as reflected in the lesser ESS in Table 3, but this is counterbalanced by the impressive decrease in computing time, leading again to an overall gain in terms of ESS/ t of about 9.

6. Conclusion. We introduced in this paper Delayed Acceptance, a generic and easily implemented modification of the standard Metropolis–Hastings algorithm that splits the acceptance rate into more than one step in order to increase the computational efficiency of the resulting MCMC, under the sole condition that the Metropolis–Hastings ratio can be factorised this way.

The choice of splitting the target distribution into parts ultimately depends on the respective costs of computing the said parts and of reducing theoretically the overall acceptance rate and expected square jump distance (ESJD). Still, this generic alternative to the standard Metropolis–Hastings approach can be considered on a customary basis, since it both requires very little modification in programming and can be easily tested against the basic version, both empirically and theoretically by the results of Section 2. The Delayed Acceptance algorithm presented in Section 1 can significantly decrease the computational time *per se* as well as the overall acceptance rate. Nevertheless, the examples presented in Section 5 suggest that the gain in terms of computational time is not linear in the reduction of the acceptance rate, especially in the presence of optimisation techniques like in Section 3.

Furthermore, our Delayed Acceptance algorithm does naturally merge with the widening range of prefetching techniques, in order to make use of parallelisation and reduce the overall computational time even more significantly. Most settings of interest are open to take advantage of the proposed method, if mostly in the situation of Bayesian statistics where the target density and/or the Metropolis–Hastings ratio always allow for a natural factorisation. The case when the likelihood function can be factorised in an useful way represents the best gain brought by our solution, in terms of computational time, and it may easily improve even more by exploiting parallelisation techniques.

Acknowledgments. The authors are grateful to the reviewers for their comments and suggestions. Thanks to Christophe Andrieu for a very helpful discussion on an earlier version of the manuscript. The massive help provided by Jean-Michel Marin and Pierre Pudlo towards an implementation on a large cluster has been fundamental in the completion of this work. Thanks to Samuel Livingstone for suggesting the Geometric MALA example and also thanks to Philip Nutzman for an interesting conversation including the suggestion of the method proposed in (4.2). Christian P. Robert research was partly financed by Agence Nationale de la Recherche (ANR, 212, rue de Bercy 75012 Paris) on the 2012–2015 ANR-11-BS01-0010 grant and by a 2016–2021 senior chair grant of Institut Universitaire de France. Marco Banterle’s PhD was funded by Université Paris Dauphine.

REFERENCES

- [1] C. Andrieu, A. Lee and M. Vihola et al., [Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers](#), *Bernoulli*, **24** (2018), 842–872.
- [2] E. Angelino, E. Kohler, A. Waterland, M. Seltzer and R. Adams, Accelerating MCMC via parallel predictive prefetching, *UAI’14 Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, (2014), 22–31.
- [3] R. Bardenet, A. Doucet and C. Holmes, On Markov chain Monte Carlo methods for tall data, *The Journal of Machine Learning Research*, **18** (2017), Paper No. 47, 43 pp.
- [4] A. Brockwell, [Parallel Markov chain Monte Carlo simulation by pre-fetching](#), *J. Comput. Graphical Stat.*, **15** (2006), 246–261.
- [5] J. Christen and C. Fox, [Markov chain Monte Carlo using an approximation](#), *Journal of Computational and Graphical Statistics*, **14** (2005), 795–810.

- [6] O. F. Christensen, G. O. Roberts and J. S. Rosenthal, [Scaling limits for the transient phase of local Metropolis–Hastings algorithms](#), *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67** (2005), 253–268.
- [7] R. Cornish, P. Vanetti, A. Bouchard-Côté, G. Deligiannidis and A. Doucet, Scalable Metropolis-Hastings for exact Bayesian inference with large datasets, arXiv preprint, [arXiv:1901.09881](#).
- [8] L. Devroye, *Nonuniform Random Variate Generation*, Springer-Verlag, New York, 1986.
- [9] J. Diebolt and C. P. Robert, [Estimation of finite mixture distributions by Bayesian sampling](#), *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **56** (1994), 363–375.
- [10] C. Fox and G. Nicholls, Sampling conductivity images via MCMC, *The Art and Science of Bayesian Image Analysis*, (1997), 91–100.
- [11] A. Gelfand and S. Sahu, [On Markov chain Monte Carlo acceleration](#), *J. Comput. Graph. Statist.*, **3** (1994), 261–276.
- [12] M. Girolami and B. Calderhead, [Riemann manifold Langevin and Hamiltonian Monte Carlo methods](#), *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **73** (2011), 123–214.
- [13] A. Golightly, D. Henderson and C. Sherlock, [Delayed acceptance particle MCMC for exact inference in stochastic kinetic models](#), *Statistics and Computing*, **25** (2015), 1039–1055.
- [14] H. Jeffreys, *Theory of Probability*, 1st ed. The Clarendon Press, Oxford, 1939.
- [15] A. Korattikara, Y. Chen and M. Welling, Austerity in MCMC land: Cutting the Metropolis-Hastings budget, In *ICML 2014, International Conference on Machine Learning*, (2014), 181–189.
- [16] G. MacLachlan and D. Peel, *Finite Mixture Models*, John Wiley, New York, 2000.
- [17] K.L. Mengersen and R. Tweedie, [Rates of convergence of the Hastings and Metropolis algorithms](#), *Ann. Statist.*, **24** (1996), 101–121.
- [18] P. Neal and G. O. Roberts, [Optimal scaling of random walk Metropolis algorithms with non-Gaussian proposals](#), *Methodology and Computing in Applied Probability*, **13** (2011), 583–601.
- [19] R. Neal, Markov chain Monte Carlo methods based on ‘slicing’ the density function, Tech. rep., University of Toronto, 1997.
- [20] W. Neiswanger, C. Wang and E. Xing, Asymptotically exact, embarrassingly parallel MCMC, arXiv preprint, 2013, [arXiv:1311.4780](#).
- [21] P. Peskun, [Optimum Monte Carlo sampling using Markov chains](#), *Biometrika*, **60** (1973), 607–612.
- [22] M. Plummer, N. Best, K. Cowles and K. Vines, CODA: Convergence diagnosis and output analysis for MCMC, *R News*, **6** (2006), 7–11.
- [23] C. P. Robert, *The Bayesian Choice*, 2nd ed. Springer-Verlag, New York, 2001.
- [24] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer-Verlag, New York, 2004.
- [25] C. P. Robert and D. M. Titterton, Reparameterisation strategies for hidden Markov models and Bayesian approaches to maximum likelihood estimation, *Statistics and Computing*, **8** (1998), 145–158.
- [26] G. O. Roberts, A. Gelman and W. R. Gilks, [Weak convergence and optimal scaling of random walk Metropolis algorithms](#), *Ann. Appl. Probab.*, **7** (1997), 110–120.
- [27] G. O. Roberts and J. S. Rosenthal, [Optimal scaling for various Metropolis-Hastings algorithms](#), *Statist. Science*, **16** (2001), 351–367.
- [28] G. O. Roberts and J. S. Rosenthal, [Coupling and ergodicity of adaptive MCMC](#), *J. Applied Proba.*, **44** (2005), 458–475.
- [29] G. O. Roberts and O. Stramer, [Langevin diffusions and Metropolis–Hastings algorithms](#), *Methodology and Computing in Applied Probability*, **4** (2002), 337–357.
- [30] G. O. Roberts and R. Tweedie, [Geometric convergence and central limit theorems for multi-dimensional Hastings and Metropolis algorithms](#), *Biometrika*, **83** (1996), 95–110.
- [31] K. Roeder and L. Wasserman, [Practical Bayesian density estimation using mixtures of Normals](#), *J. American Statist. Assoc.*, **92** (1997), 894–902.
- [32] S. Scott, A. Blocker, F. Bonassi, M. Chipman, E. George and R. McCulloch, Bayes and big data: The consensus Monte Carlo algorithm *EFaBBayes 250 Conference*, **16** (2013).
- [33] C. Sherlock, A. Golightly and D. A. Henderson, [Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods](#), *Journal of Computational and Graphical Statistics*, **26** (2017), 434–444.

- [34] C. Sherlock and G. O. Roberts, [Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets](#), *Bernoulli*, **15** (2009), 774–798.
- [35] C. Sherlock, A. Thiery and A. Golightly, Efficiency of delayed-acceptance random walk Metropolis algorithms, arXiv preprint, 2015, [arXiv:1506.08155](#).
- [36] C. Sherlock, A. Thiery, G. O. Roberts and J. S. Rosenthal, [On the efficiency of pseudo-marginal random walk Metropolis algorithms](#), *The Annals of Statistics*, **43** (2015), 238–275.
- [37] A. Y. Shestopaloff and R. M. Neal, MCMC for non-linear state space models using ensembles of latent sequences, arXiv preprint, 2013, [arXiv:1305.0320](#).
- [38] M. Stephens, *Bayesian Methods for Mixtures of Normal Distributions*, Ph.D. thesis, University of Oxford, 1997.
- [39] I. Strid, [Efficient parallelisation of Metropolis–Hastings algorithms using a prefetching approach](#), *Computational Statistics & Data Analysis*, **54** (2010), 2814–2835.
- [40] L. Tierney, [A note on Metropolis-Hastings kernels for general state spaces](#), *Ann. Appl. Probab.*, **8** (1998), 1–9.
- [41] L. Tierney and A. Mira, Some adaptive Monte Carlo methods for Bayesian inference, *Statistics in Medicine*, **18** (1998), 2507–2515.
- [42] X. Wang and D. Dunson, Parallel MCMC via Weierstrass sampler, arXiv preprint, 2013, [arXiv:1312.4605](#).

E-mail address: marco.banterle@gmail.com

E-mail address: clara.grazian@gmail.com

E-mail address: anthony.lee@bristol.ac.uk

E-mail address: xian@ceremade.dauphine.fr