LA-UR- 08-7126

Title:

Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling

Author(s):

Jasper A. Vrugt
Cajo J.F. ter Braak
Cees G.H. Diks
James M. Hyman
Bruce A. Robinson
Dave Higdon

Submitted to:

International Journal of Nonlinear Sciences and Numerical Simulation

Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling

Jasper A. Vrugt[1]*, C.J.F. ter Braak[2]*, C.G.H. Diks[3], Bruce A. Robinson[4],
James M. Hyman[5] and Dave Higdon[6]

Abstract

Markov chain Monte Carlo (MCMC) methods have found widespread use in many fields of study to estimate the average properties of complex systems, and for posterior inference in a Bayesian framework. Existing theory and experiments prove convergence of well constructed MCMC schemes to the appropriate limiting distribution under a variety of different conditions. In practice, however this convergence is often observed to be disturbingly slow. This is frequently caused by an inappropriate selection of the proposal distribution used to generate trial moves in the Markov Chain. Here we show that significant improvements to the efficiency of MCMC simulation can be made by using a self-adaptive Differential Evolution learning strategy within a population-based evolutionary framework. This scheme, entitled DiffeRential Evolution Adaptive Metropolis or DREAM, runs multiple different chains simultaneously for global exploration, and automatically tunes the scale and orientation of the proposal distribution in randomized subspaces during the search. Ergodicity of the algorithm is proved, and various examples involving nonlinearity, high-dimensionality, and multimodality show that DREAM is generally superior to other adaptive MCMC sampling approaches. The DREAM scheme significantly enhances the applicability of MCMC simulation to complex, multi-modal search problems.

Abbreviations: MCMC, Markov chain Monte Carlo; RWM, random walk metropolis; DE-MC, differential evolution Markov chain; DRAM, delayed rejection adaptive Metropolis; DREAM, differential evolution adaptive metropolis; SCE-UA, shuffled complex evolution - university of Arizona

[1] Center for NonLinear Studies (CNLS), Los Alamos National Laboratory, Los Alamos, NM 87545, USA; E-mail: vrugt@lanl.gov

[2] Biometris, Wageningen University and Research Centre, 6700 AC, Wageningen, The Netherlands

[3] Center for Nonlinear Dynamics in Economics and Finance, University of Amsterdam, Amsterdam, The Netherlands

[4] Civilian Nuclear Program Office (SPO-CNP), Los Alamos National Laboratory, Los Alamos, NM 87545, USA

[5] Mathematical Modeling and Analysis group (T-7), Los Alamos National Laboratory, Los Alamos, NM 87545, USA

[6] Statistical Sciences (CCS-6), Los Alamos National Laboratory, Los Alamos, NM 87545, USA

* These authors contributed equally to this work.

Introduction

In 1953, Metropolis et al. [1] introduced the Markov chain Monte Carlo (MCMC) scheme to estimate $E_\pi f(\mathbf{x})$, the expectation of a function f with respect to a distribution π. The basis of this method is a Markov chain that generates a random walk through the search space and successively visits solutions with stable frequencies stemming from distribution π. The MCMC estimator is approximated as the unweighted mean of f along the last M elements of the realized path of the chain, $\frac{1}{M}\sum_{i=1}^{M} f(\mathbf{x}_i)$, that is, after a burn-in period to allow the chain to explore the search space and reach its stationary regime. This algorithm has been used extensively in statistical physics, and appeared also in spatial statistics and statistical image analysis. In [2] the MCMC method was extended for posterior inference in a Bayesian framework. Ever since, the method has found widespread use for sampling from posterior distributions in many fields of study ranging from physics and chemistry, to finance, economics, genetics, statistical inference, hydrology, biology and bioinformatics [3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

To visit configurations with a stable frequency, an MCMC algorithm generates trial moves from the current ("*old*") position of the Markov chain $\mathbf{x}_{t-1}$ to a new state $\mathbf{z}$. The earliest and most general MCMC approach is the random walk Metropolis (RWM) algorithm. Assume that we have already sampled points $\{\mathbf{x}_0,\ldots,\mathbf{x}_{t-1}\}$ this algorithms proceeds in the following three steps. First, a candidate point $\mathbf{z}$ is sampled from a proposal distribution $q(\cdot|\cdot)$ that depends on the present location, $\mathbf{x}_{t-1}$ and is symmetric, $q(\mathbf{x}_{t-1},\mathbf{z}) = q(\mathbf{z},\mathbf{x}_{t-1})$. Next, the candidate point is either accepted or rejected using the Metropolis acceptance probability:

$$\alpha(\mathbf{x}_{t-1},\mathbf{z}) = \begin{cases} \min\left[\dfrac{\pi(\mathbf{z})}{\pi(\mathbf{x}_{t-1})},1\right] & \text{if } \pi(\mathbf{x}_{t-1}) > 0 \\ 1 & \text{if } \pi(\mathbf{x}_{t-1}) = 0 \end{cases} \tag{1}$$

where $\pi(\cdot)$ denotes the probability density function (pdf) of the target distribution. Finally, if the proposal is accepted the chain moves to $\mathbf{z}$, otherwise the chain remains at its current location $\mathbf{x}_{t-1}$.

The original RWM scheme is constructed to maintain detailed balance with respect to $\pi(\cdot)$ at each step in the chain:

$$p(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \to \mathbf{z}) = p(\mathbf{z})p(\mathbf{z} \to \mathbf{x}_{t-1}) \tag{2}$$

where $p(\mathbf{x}_{t-1})$ ($p(\mathbf{z})$) denotes the probability of finding the system in the state $\mathbf{x}_{t-1}$ ($\mathbf{z}$), and $p(\mathbf{x}_{t-1} \to \mathbf{z})$ ($p(\mathbf{z} \to \mathbf{x}_{t-1})$) denotes the conditional probability to perform a trial move from $\mathbf{x}_{t-1}$ to $\mathbf{z}$ ($\mathbf{z}$ to $\mathbf{x}_{t-1}$). The result is a Markov chain which under some regularity conditions has a unique stationary distribution with pdf $\pi(\cdot)$. Hastings [13] extended Eq. (1) to include non-symmetrical proposal distributions, i.e. $q(\mathbf{x}_{t-1},\mathbf{z}) \neq q(\mathbf{z},\mathbf{x}_{t-1})$ in which a proposal jump to $\mathbf{z}$ and the reverse jump do not have equal probability. This extension is called the Metropolis Hastings algorithm (MH), and has become the basic building block of many existing MCMC sampling schemes.

The simplicity of the original MH algorithm and theoretically sound statistical basis of the method has lead to widespread implementation and use. However, in practice the MH algorithm requires tuning of some internal variables before the MCMC simulator works effectively in practice. The efficiency of the method is affected by the scale and orientation of the proposal distribution, $q(\mathbf{x}_{t-1},\cdot)$ used to generate trial moves (transitions) in the Markov Chain. When the proposal distribution is too wide, too many candidate points are rejected, and therefore the chain will not mix efficiently and only converge slowly to the target distribution. On the other hand, when the proposal distribution is too narrow, nearly all candidate points are accepted, but the distance moved is so small that it will take a prohibitively large number of updates before the sampler will find solutions that belong to the target distribution. The choice of the proposal distribution is therefore crucial and determines the practical applicability of MCMC simulation in many fields of study [14].

Automatic tuning of the proposal distribution is a powerful remedy to overcome many of the difficulties associated with the selection of an appropriate jumping distribution. This approach uses the information from the sampling history to continuously adapt the shape and size of the proposal distribution and evolve the sampler efficiently towards a limiting distribution. Examples of such methods are the Adaptive Proposal (AP) and Adaptive Metropolis (AM) schemes presented in [15,16], respectively. These algorithms utilize a single Markov chain and continuously adapt the covariance, $\mathbf{C}_t$ of a Gaussian proposal distribution, $q_t(\mathbf{x}_{t-1},\cdot) = N_d(\mathbf{x}_{t-1},\mathbf{C}_t)$ using the information contained in the sample path of the chain, $\mathbf{C}_t = s_d\text{Cov}(\mathbf{x}_0,\ldots,\mathbf{x}_{t-1}\} + s_d\varepsilon\mathbf{I}_d$. Here, s_d represents a scaling factor that depends only on the dimensionality d of the problem, $\mathbf{I}_d$ signifies the d-dimensional identity matrix, and ε is a constant with a small value compared to that of the target so that the whole parameter space can be reached. As a basic choice, the scaling factor is chosen to be $s_d = 2.4^2/d$ which is optimal for Gaussian target and proposal distributions [17]. To improve efficiency for higher dimensional problems, [18] has extended the AM method to componentwise updating, where $\mathbf{x}$ is sampled component by component.

Also relevant to the work presented here is the Delayed Rejection Adaptive Metropolis (DRAM) algorithm presented in [19]. This MCMC scheme combines Adaptive Metropolis and Delayed Rejection [20] to continuously alternate between larger and smaller steps in the Markov chain. Upon rejection, instead of retaining the same position, a second stage trial is proposed using a deflated proposal distribution. The acceptance probability of this second candidate point is computed so that reversibility of the Markov chain is preserved. Various published studies have shown that DR significantly enhances the acceptance rate of MCMC simulation. Care must be taken, however that adaptation does not destroy the detailed balance and overall ergodicity of the Markov chain.

The covariance adaptation strategy presented in [15] and used in the AM, AP, DRAM and SCAM schemes works well for relatively simple inference problems, but is inefficient and unreliable when confronted with posteriors with very heavy tails and, possibly, infinite first and second moments, and with complex posterior surfaces that contain multiple regions of attraction and numerous local optima. A single chain is unable to efficiently cope with the latter difficulties, and traverse well in pursuit of sampling the target distribution. This will be demonstrated in this paper. Other adaptive MCMC approaches proposed in the literature include the work of [21], [22], and [23] using multiple different chains to explore the search space, and the regeneration and self-regenerative algorithms of [24], [25] and [26] that approximate the proposal distribution using mixture components. Yet, these schemes have limited practical applicability in higher dimensions, either requiring significant memory storage or complicated fitting of high component and dimensional mixture distributions.

Recognizing these limitations of existing MCMC schemes, ter Braak [27] recently presented the Differential Evolution-Markov Chain (DE-MC) method. In DE-MC, N different Markov Chains $\{\mathbf{x}_t^i, i = 1,..., N\}$ are run simultaneously in parallel. At the current time they form a population, conveniently stored as an $N \times d$ matrix $\mathbf{X}$, with d the dimension of π. Jumps in each chain $i = 1,\ldots,N$ are generated by taking a fixed multiple of the difference of two randomly chosen members (chains) of $\mathbf{X}$ (without replacement):

$$\mathbf{z}^i = \mathbf{x}_{t-1}^i + \gamma(\mathbf{x}_{t-1}^{r1} - \mathbf{x}_{t-1}^{r2}) + \mathbf{e} \qquad r_1 \neq r_2 \neq i \tag{3}$$

The Metropolis ratio is used to decide whether to accept candidate points or not. From the guidelines of s_d in RWM, the optimal choice of $\gamma = 2.38/\sqrt{(2d)}$. Every 10th generation, γ = 1.0 to allow direct jumps between two modes [27]. The DE-MC approach solves two important problems in MCMC sampling. First, DE-MC automatically selects an appropriate scale and orientation of the proposal distribution en route to the target distribution. Second, heavy-tailed and multimodal target distributions are efficiently accommodated, as DE-MC directly uses the current location of the chains (stored in $\mathbf{X}$), instead of Cov($\mathbf{X}$) or the adaptive covariance, to generate candidate points, allowing the possibility of direct jumps between one mode of $\mathbf{X}$ and another.

In this paper, we show that the efficiency of DE-MC can be enhanced, sometimes dramatically, using a similar evolution scheme, but with self-adaptive randomized subspace sampling. This novel method maintains detailed balance and ergodicity and provides new opportunities for solving previously intractable sampling problems in many fields of study.

Materials and Methods

The MCMC algorithm developed in the present study is entitled, DiffeRential Evolution Adaptive Metropolis, or DREAM, and uses DE-MC as its main building block. We drop the subscript for generation time t. In the following, the current state of the i^{th} chain is given by a d-dimensional vector $\mathbf{x}^i$ (i =1,…,N) and its j^{th} element by x_j^i. The DREAM algorithm is as follows:

1. Draw an initial population $\{\mathbf{x}^i, i = 1,\ldots,N\}$ using the prior distribution.
2. Compute the density $\pi(\mathbf{x}^i)$ for $i = 1,\ldots,N$.

   FOR $i \leftarrow 1,\ldots,N$ DO (CHAIN EVOLUTION)

3. Generate a candidate point, $\mathbf{z}^i$ in chain i.

$$\mathbf{z}^i = \mathbf{x}^i + (\mathbf{1}_d + \mathbf{e})\gamma(\delta,d')\left[\sum_{j=1}^{\delta}\mathbf{x}^{r_1(j)} - \sum_{n=1}^{\delta}\mathbf{x}^{r_2(n)}\right] + \varepsilon \qquad (4)$$

   where δ signifies the number of pairs used to generate the proposal, and $r_1(j)$, $r_2(n) \in \{1,\ldots,N\}$; $r_1(j) \neq r_2(n) \neq i$ for $j = 1, \ldots, \delta$ and $n = 1, \ldots, \delta$. The values of $\mathbf{e}$ and ε are drawn from $U_d(-b,b)$ and $N_d(0,b^*)$ with b and b^* small compared to the width of the target distribution, respectively, and the value of jump-size γ depends on δ and d', the number of dimensions that will be updated jointly (see next step and Eq. (5)).
4. Replace each element ($j = 1,\ldots,d$) of the proposal z^i_j with x^i_j using a binomial scheme with probability $1\text{-}CR$, where CR is the crossover probability. With $CR = 1$, all dimensions are updated jointly and $d' = d$ (see Eq. (5)).
5. Compute $\pi(\mathbf{z}^i)$ and $\alpha(\mathbf{x}^i,\mathbf{z}^i)$ of the candidate point.
6. If accepted, $\mathbf{x}^i = \mathbf{z}^i$, otherwise remain at $\mathbf{x}^i$.

   END FOR (CHAIN EVOLUTION)

7. Remove outlier chains using the Inter-Quartile-Range (IQR) statistic. This is done during burn-in and discussed below.
8. Compute the Gelman-Rubin $\hat{R}_j$ convergence diagnostic [28] for each dimension $j = 1,\ldots,d$ using the last 50% of the samples in each chain.
9. If $\hat{R}_j < 1.2$ for all j, stop, otherwise go to CHAIN EVOLUTION.

This algorithm contains four extensions of DE-MC to improve search efficiency:

1. The original DE-MC scheme uses only two members of $\mathbf{X}$ to generate candidate points. This might downplay variation in the proposals, especially when the number of members of $\mathbf{X}$ is small. To increase diversity, DREAM generates proposals, $\mathbf{z}^i$ in Eq. (4) with DE that also includes higher-order pairs. Similar to DE-MC, a good choice for $\gamma(\delta,d') = 2.38/\sqrt{2\delta d'}$ with $d' = d$, but potentially decreased in the next step. This is expected to yield an acceptance probability of 0.44 for $d' = 1$, 0.28 for $d' = 5$ and 0.23 for large d'. To enable jumping between different modes of the posterior [27], we set $\gamma(\delta,d') = 1.0$ at every 5th generation.
2. In higher dimensions it is often not optimal to sample all d elements $\{z^i_j, j = 1,\ldots,d\}$ of $\mathbf{z}^i$ simultaneously. Therefore, DREAM implements a

randomized subspace sampling strategy that modifies each dimension with probability *CR* each time a candidate point is generated and decreases d' (from its initial value $d' = d$) accordingly:

$$z_j^i = \begin{cases} x_j^i & \text{if } U \leq 1 - CR,\ d' = d' - 1 \\ z_j^i & \text{otherwise} \end{cases} \quad j = 1,\ldots,d \tag{5}$$

where *CR* denotes the crossover probability, and $U \in [0,1]$ is a draw from a uniform distribution. Note that this subspace sampling strategy, activated when $CR < 1$, constantly introduces new directions that chains can take outside the subspace spanned by their current positions. In principle, this allows using $N < d$ in DREAM, an important advantage over DE-MC that requires $N = 2d$ chains to be run in parallel [27].

3. Outlier chains can significantly deteriorate the performance of MCMC samplers, and need to be removed to facilitate convergence to a limiting distribution. To detect aberrant trajectories, DREAM stores in Ω the mean of the logarithm of the posterior densities of the last 50% of the samples in each chain. From these, the Inter Quartile-Range statistic, $IQR = Q_3 - Q_1$ is computed, in which Q_1 and Q_3 denote the lower and upper quartile of the N different chains. Chains with $\Omega < Q_1 - 2 \cdot IQR$ are considered outliers, and are moved to the current best member of $\mathbf{X}$. This step does not maintain detailed balance and can therefore only be used during burn-in. If an outlier chain is being detected we apply another burn-in period before summarizing the posterior moments.
4. To speed up convergence to the target distribution, DREAM estimates a distribution of crossover probabilities during burn-in that favors large jumps over smaller ones in each of the N chains. Normalization is required to ensure that all d dimensions contribute equally to Δ. In practice, we generate a discrete number of candidate points for each crossover value $\{m/n_{CR} \mid m = 1,\ldots,n_{CR}\}$. Here n_{CR} is a user-defined parameter. We use the following algorithm for adaptation of p_m, the probability of each individual *CR* value:

   (a) Set $t = 1$, $L_m = 0$, $p_m = 1/n_{CR}$, $m = 1,\ldots,n_{CR}$.

   FOR $i \leftarrow 1,\ldots,N$ DO (IMPROVE DISTRIBUTION)

   (b) Sample m from the numbers $1,\ldots,n_{CR}$ using the multinomial distribution $M(\cdot,p_1,\ldots,p_m)$.
   (c) Set $CR = m/n_{CR}$ and $L_m = L_m + 1$.
   (d) Create a candidate point $\mathbf{z}^i$ using Eqs (4) and (5) with crossover probability *CR*.
   (e) Accept/reject the candidate point using Eq. (1).
   (f) Compute the squared normalized jumping distance

   $$\Delta_m = \Delta_m + \sum_{j=1}^{d} (x_{j,t}^i - x_{j,t-1}^i)^2 / r_j^2$$

   where r_j denotes the current standard deviation of column j of $\mathbf{X}$.

```
    END FOR (IMPROVE DISTRIBUTION)
        (g) Update the probability of the different $CR$ values
            $p_m = tN \cdot (\Delta_m / L_m) / \sum_{j=1}^{n_{CR}} \Delta_j ,\ m = 1,...,n_{CR}$.
        (h) Set $t = t + 1$.
        (i) If $t$ in burn-in period, go to IMPROVE DISTRIBUTION, otherwise stop.
```

This tuning strategy results in a distribution over the m crossover values designed to decrease autocorrelation between two subsequent samples in each chain. A similar idea using the expected squared jumped distance is presented in [30]. Table 6 lists p_m, $m = 1,\dots,n_{CR}$ for each of the case studies considered herein.

In [19] it has been shown that delayed rejection can further improve the efficiency of MCMC simulation and provide better estimates of the various moments of the posterior pdf. To test the usefulness of DR within DREAM, we can replace step (6) in the pseudo-code of DREAM, with the following series of four steps, 6a-6d:

(6a). If accepted, $\mathbf{x}^i = \mathbf{z}^i$, and go to 7, otherwise continue with 6b.

(6b). Attempt a delayed rejection step, $\mathbf{z}^i_{DR} = N_d(\mathbf{x}^i, \kappa^{-1}\mathbf{C}_t)$ where $\mathbf{C}_t = s_d\text{Cov}(\mathbf{X}_\text{t}\} + s_d\varepsilon\mathbf{I}_d$ and κ is a scaling (deflation) factor of the second proposal.

(6c). Compute $\pi(\mathbf{z}^i_{DR})$ and calculate a modified Metropolis acceptance probability

$$\alpha(\mathbf{x}^i,\mathbf{z}^i,\mathbf{z}^i_{DR}) = \min\left(\frac{\pi(\mathbf{z}^i_{DR})q(\mathbf{z}^i_{DR},\mathbf{z}^i)q(\mathbf{z}^i_{DR},\mathbf{z}^i,\mathbf{x}^i)[1-\alpha(\mathbf{z}^i_{DR},\mathbf{z}^i)]}{\pi(\mathbf{x}^i)q(\mathbf{x}^i,\mathbf{z}^i)q(\mathbf{x}^i,\mathbf{z}^i,\mathbf{z}^i_{DR})[1-\alpha(\mathbf{x}^i,\mathbf{z}^i)]},1\right)$$

(6d). If accepted, $\mathbf{x}^i = \mathbf{z}^i_{DR}$, otherwise remain at $\mathbf{x}^i$.

So, upon rejection of the first proposal in each chain, a second trial move is proposed. The acceptance probability of this second proposal is computed so that reversibility and detailed balance of the Markov chain are preserved. Note that the delayed rejection step only considers the current location of the N chains to compute the variance-covariance matrix, $\mathbf{C}$ of the proposal distribution. In case study 1 of this paper we compare the results of DREAM, with and without delayed rejection sampling.

We now provide a formal proof of convergence of the DREAM algorithm, followed by four case studies with increasing complexity.

Theorem: DREAM yields a Markov Chain that is ergodic with unique stationary distribution with pdf $\pi(\cdot)^N$. *Proof*: The proof consists of three parts and follows [31,32].

1. Chains are updated sequentially and conditionally on the other chains. Thus DREAM is an N-component Metropolis-within-Gibbs algorithm that defines a single Markov chain on the state space S^N [32]. The conditional pdf of each component is $\pi(\cdot)$.

2. The update of the i^{th} chain uses a mixture of kernels. For $\delta = 1$, there are $\binom{N-1}{2}$ such kernels. This mixture kernel maintains detailed balance with respect to $\pi(\cdot)$, if each of its components does [31], as we show now. For the i^{th} chain, the conditional probability to jump from $\mathbf{x}^i$ to $\mathbf{z}^i$, $p(\mathbf{x}^i \rightarrow \mathbf{z}^i)$ is equal to the reverse jump $p(\mathbf{z}^i \rightarrow \mathbf{x}^i)$ for any distribution of $\mathbf{e}$, as the distribution of $\boldsymbol{\varepsilon}$ is symmetric and the pair $(\mathbf{x}^{r1}, \mathbf{x}^{r2})$ is as likely as $(\mathbf{x}^{r2}, \mathbf{x}^{r1})$. This also holds true for $\delta > 1$, when more than two members are selected to generate a proposal point, and for the binomial crossover scheme used to modify only selected dimensions of $\mathbf{x}^i$. Detailed balance is thus achieved point wise by accepting the proposal with probability $\min(\pi(\mathbf{z}^i)/\pi(\mathbf{x}^i),1)$. Detailed balance also holds in terms of arbitrary measurable sets, as the Jacobian of the transformation of Eq. (4) is 1 in absolute value. In case of a delayed rejection step, the acceptance probability of the second proposal is computed so that detailed balance of the i^{th} chain is preserved.
3. As each update maintains conditional detailed balance, the joint stationary distribution associated with DREAM is $\pi(\mathbf{x}^1,\ldots,\mathbf{x}^N) = \pi(\mathbf{x}^1) \times \ldots \times \pi(\mathbf{x}^N)$ [31,32]. This distribution is unique and must be the limiting distribution, because the chains are aperiodic, positive recurrent (not transient) and irreducible. The first two conditions are satisfied, except for trivial exceptions. The unbounded support of the distribution of $\boldsymbol{\varepsilon}$ in (3) guarantees the third condition. This concludes the ergodicity proof.

Case Studies

We conducted a wide range of numerical experiments using three known posterior target distributions and one real-world study involving the calibration of a flood forecasting model. These case studies cover a diverse set of problem features, including high-dimensionality, nonlinearity, non-convexity, multimodality, and numerous local optima. In all our calculations with DREAM, $N = d$, $b = 0.05$, $b^* = 10^{-6}$, $\gamma(\delta, d') = 2.38/\sqrt{(2\delta d')}$ and $\gamma(\delta, d') = 1.0$ temporarily at every 5th generation. To benchmark the performance of DREAM, we include comparison against the classical RWM, DRAM [35] and DE-MC [27] schemes for posterior inference. In all case studies presented herein, DRAM and RWM use a single chain (N=1) for posterior exploration whereas DE-MC uses $N = 2d$ as recommended in [27]. Unless stated otherwise, we generate proposals with RWM using $N_d(\mathbf{0},c\mathbf{I}_d)$ with c tuned to get an acceptance probability of about 0.24, which is considered optimal [33]. The statistics listed in Tables 1-4 denote averages over 100 different MCMC runs.

In this paper, we focus on single and parallel chain methods only. Our initial results show that DREAM also comparables favorably well against Sequential Monte Carlo (SMC) based sampling methods that recently have been introduced in [34]. These findings are particularly true for high-dimensional problems for which a relatively high number of particles are needed with SMC to properly sample the underlying target distribution. The resampling step in SMC can therefore benefit from the work presented herein, as it allows for a much smaller number of particles for posterior exploration. We will report our findings related to this research in due course.

Case Study 1: 100-dimensional Multivariate Normal Distribution

To test the performance of DREAM in the presence of high-dimensionality, the first case study involves a 100-dimensional multivariate normal distribution, centered at the zero vector. The covariance matrix was set such that the variance of the j^{th} variable was equal to j, and all pairwise correlations were 0.5. The initial population is drawn using either $\mathbf{X} \in [9.9,10.0]^d$ or $[-5.0,15.0]^d$ to test the performance of the four samplers when initialized with starting points from an under and mostly overdispersed prior distribution. We used N = 200 parallel chains with DE-MC and N = 100 with DREAM, whereas DRAM was ran with a non-adaptation period of 500 draws, and $\kappa = 5$.

Table 1 summarizes the performance of DREAM with different values of n_{CR} and δ for the underdispersed initial distribution. The notation $\delta = \{1,2\}$ indicates that 50% of the candidate points are generated with $\delta = 1$, and the other half of the proposals using $\delta = 2$. For $\delta = \{1,2,3\}$ the probability of use of $\delta = 1$, $\delta = 2$ and $\delta = 3$ are 1/3 each. The columns list the average normalized Euclidean distance (D) to the true mean, μ_π and standard deviation σ_π of the target, the number of function evaluations needed to reach convergence (*FE*) and the acceptance rate (*AR*). These listed summary variables were obtained using a total of 1,000,000 function evaluations per individual trial, with a burn-in of 750,000 draws. The value of D is computed as:

$$D = \sqrt{\frac{1}{2d}\sum_{i=1}^{d}\left(\left[\frac{\mu_\pi^i - \hat{\mu}_\pi^i}{\sigma_\pi^i}\right]^2 + \left[\frac{(\sigma_\pi^i - \hat{\sigma}_\pi^i)}{\sigma_\pi^i}\right]^2\right)} \qquad (6)$$

To demonstrate what effect adaptation of *CR* has on the efficiency of DREAM, the second and third parts of Table 1 summarize the values of D, *FE* and *AR* obtained with fixed values $p_1 = \ldots. = p_{n_{CR}} = 1/n_{CR}$ and single fixed *CR* values throughout the entire simulation, respectively.

The results presented in Table 1 highlight several important observations. First, the estimates of the first and second order moment of the posterior target are much closer to their respective target values when subspace sampling is used ($n_{CR} \geq 2$). Second, the results appear to be fairly insensitive to the choice of δ. Larger values of δ somewhat decrease, on average, the values of D. Third, and as anticipated, delayed rejection significantly increases the acceptance rate of proposals, but at the expense of requiring many more function evaluations to converge, and deteriorating the average distance to the first and second order moments of the target. By deflating the variance-covariance structure of the second proposal in delayed rejection, the average jumping distance between is reduced, thereby requiring more function evaluations to explore the entire target distribution. Fourth, adaptation of the probability of individual CR values significantly increases convergence speed to the target distribution. About 10 – 20% less function evaluations are required for convergence when the probability of *CR* values is adapted by maximizing the average normalized jumping distance. Finally, the worst results are obtained when single fixed *CR* values such as 1/2 and 1/3 are used throughout the simulation instead of a distribution of *CR* values (bottom Table 1).

Based on these results, we fix $n_{CR} = 3$, and $\delta = \{1,2,3\}$ in DREAM and use these settings to create the results presented in Figure 1. The results for the overdispersed initial distribution were qualitatively very similar as those presented here for the underdispersed prior. On average, about 2 outlier chains were detected with DREAM for each of the settings considered in Table 1.

Figure 1 illustrates how the sample mean of x_l, standard deviations of x_1 and x_{100} and $\mathrm{Cov}(x_1,x_{100})$ evolve in time for the two initial distributions and four sampling methods. These posterior moments were computed using a sliding window containing the 100,000 most recent samples created with each method; these samples come from the last 500 and 1,000 generations with DE-MC ($N = 2d$) and DREAM ($N = d$), respectively, and from the last 100,000 iterations for RWM and DRAM using a single chain. The true values (0, 1, 5 and 10, respectively) are separately indicated in each panel with different symbols. Judged by inspection of the figures DREAM has the overall best performance. This algorithm smoothly converges to the target values in about 250,000 function evaluations for both initial distributions. The RWM scheme exhibits difficulty sampling the correct target values, and shows considerable fluctuation in the estimates of the four moments of interest along its sampling path. This is true for both initial distributions, and demonstrates a disadvantage of using a fixed proposal distribution. The DRAM method requires many more than 1 million function evaluations for both prior distributions. Finally, the DE-MC algorithm requires about 400,000 function evaluations for the overdispersed initial distribution, and about 600,000 evaluations for the underdispersed prior distribution. These results highlight the relative efficiency and power of DREAM. Further experimentation with different algorithmic settings, yielded very similar traces as those depicted in Fig. 1.

Case Study 2: 10-dimensional Twisted Gaussian Target Distribution

The second case study considers a 10-dimensional twisted Gaussian density function first introduced in [15] which is given by the unnormalized density $\pi_b(\mathbf{x}) \propto \pi(\phi_b(\mathbf{x}))$, with $\phi_b(\mathbf{x}) = (x_1, x_2 + bx_1^2 - 100b, x_3, \ldots, x_{10})$. Here, π signifies the density of a multivariate normal distribution, $N_d(0,\Sigma)$ with $\Sigma = \mathrm{diag}(100,1,\ldots,1)$, and ϕ_b is a function that is used to transform π to a twisted distribution. The initial sample was generated from a normal distribution with variance-covariance matrix $5\mathbf{I}_d$.

Table 2 lists the performance of DREAM as function of n_{CR} and δ using $b = 0.1$ (highly nonlinear target) and $b = 0.01$ (mildly nonlinear). First, the average normalized Euclidean distance to the true mean and standard deviation of the target is higher for the more nonlinear target and, second, the pattern of results for each target is very similar to that presented for the 100-dimensional normal distribution in Table 1. Based on these results we fix $N = d$, $n_{CR} = 3$, and $\delta = \{1,2,3\}$ in DREAM and use these settings in the following case studies. We have now given recommendations for all algorithmic parameters in DREAM.

Table 3 compares the performance of DREAM against the RWM, DRAM and DE-MC sampling methods. Note that the RWM scheme employed a fixed proposal distribution, whereas DRAM and DE-MC continuously updated the scale and orientation of the proposal distribution during sampling. The DRAM scheme used a non-adaptation period of 50 draws and value of $\kappa = 10$ for the scaling factor of the second proposal. RWM and DRAM with N=1 do not allow the convergence to assessed by Gelman-Rubin statistics $\hat{R}_j$ [28] as in the DE-MC and DREAM algorithms. Therefore, we actually ran RWM and DRAM with 10 independent chains and determined the number of generations so that $\hat{R}_j < 1.2$ for all dimensions [27] using the last 50% of the samples in each chain. To mimic nevertheless the $N = 1$ case, the number of generations is then the number of function evaluations (iterations) used in a single chain and thus the appropriate value of *FE*; correspondingly the samples of the first chain only are used to calculate D and AR. This approach is advantageous for RWM and DRAM as a single chain run cannot give a reliable assessment of convergence.

The DRAM and DREAM sampling schemes have a very similar performance, requiring about 50,000 function evaluations to converge to the appropriate limiting distribution. The results for DREAM with and without *CR* adaptation show again the advantage of *CR* adaptation; with CR adaptation, about 15% less function evaluations are required. The performance of RWM and DE-MC is rather poor. These MCMC schemes require many more function evaluations to converge to the target distribution, and the RWM estimates of the posterior mean and standard deviation are, on average, much further removed from their actual values.

Case Study 3:10-dimensional Bimodal Target Distribution

The third case study considers a 10-dimensional bimodal pdf with two well-separated modes. This example is taken from [27] and is given by, $\pi(\mathbf{x}) = 1/3N_d(\mathbf{-5},\mathbf{I}_d) + 2/3N_d(\mathbf{5},\mathbf{I}_d)$ where **-5** and **5** are d-dimensional vectors. This probability distribution is notoriously difficult to approximate with MCMC simulation, because the individual modes of the normal distributions are so far separated that a jump from one mode to the other has very low probability, complicating convergence to the target.

Figure 2a depicts the transitions of x_1 in a selected set of Markov chains in DREAM during their evolution to the posterior target distribution. Each individual chain is coded with a different color. The 1-D scatter plots of the sampled parameter space demonstrate that DREAM exhibits no difficulty jumping from one mode to the other, resulting in an excellent mixing of the individual paths, and therefore a relatively quick convergence. Although DREAM employs adaptive proposal updating the sampler is not inclined to converge to a single mode, provided the initial population is wide enough. The marginal posterior pdf of x_1 is shown in Fig. 2b. The sampled histogram shows an excellent match with the known target distribution indicated with the black line. Two outlier chains were reported with DREAM during burn-in.

Table 4 compares the performance of DREAM with RWM, DE-MC and DRAM for the 10-dimensional bimodal target distribution. The listed statistics demonstrate that DREAM has superior performance. It not only requires about 50% less function evaluations than DE-MC to converge to the target, but simultaneously also provides the most accurate estimates of the first and second order moments of the target. This increased efficiency of DREAM over DE-MC is explained by the smaller number of chains it uses for posterior exploration (N = $2d$ = 20 for DE-MC against $N = d = 10$ for DREAM), and because $\gamma = 1.0$ at every 10th generation in DE-MC, but $\gamma(\delta,d') = 1.0$ at every 5th generation in DREAM, increasing the chance of jumping between the two modes, and speeding up convergence. The RWM and DRAM schemes exhibit particular poor performance on this test problem, as they are unable to jump between the two (disconnected) modes sufficiently frequently to accurately assess their underlying probability mass.

Case Study 4: Flood Forecasting with the Sacramento Soil Moisture Accounting Model

The final case study considers application of MCMC simulation to parameter inference in the Sacramento Soil Moisture Accounting (SAC-SMA) model. The SAC-SMA model is a lumped conceptual watershed model that describes the transformation from rainfall into basin runoff using six state variable reservoirs. A unit hydrograph is commonly used to rout channel inflow downstream and compute streamflow at the gauging point. This model is extensively used by the National Weather Service for flood forecasting throughout the United States, and has 13 user-specifiable (and 3 fixed) parameters, which are listed in Table 5. Inputs to the model include mean areal precipitation (MAP) and potential evapotranspiration (PET) while the outputs are estimated evapotranspiration and channel inflow. Various studies have demonstrated that calibration of the SAC-SMA model is very difficult due to the presence of numerous local optima in the parameter space with both small and large domains of attraction, discontinuous first derivatives, and curving multidimensional ridges [35]. This real-world study therefore poses an interesting challenge for MCMC samplers.

We estimate the posterior pdf of the SAC-SMA parameters using 2 years of historical daily streamflow data from the Leaf River watershed. This humid basin of approximately 1,950 km^2 is located north of Collins, Mississippi, USA. We use a squared deviation likelihood function [36, 37]

$$\pi(\mathbf{x}) = \left[\sum_{i=1}^{T} (S_i(\mathbf{x}) - \hat{S}_i)^2 \right]^{-\frac{1}{2}T} \tag{7}$$

where T denotes the number of streamflow observations, and $S(\mathbf{x})$ and $\hat{S}$ are the simulated and observed data, respectively. A uniform prior for each parameter is used with ranges specified in Table 5. Consistent with our recommendations and previous settings, we used N = 26 parallel chains with DE-MC, and N = 13 parallel chains with DREAM, whereas N = 5 independent chains where generated with RWM and DRAM. As proposal distribution in RWM, we used a multiple of the covariance matrix of the target derived with DREAM.

Figure 3 illustrates the evolution of the sampled values in millimeter of the upper zone tension water maximum storage (UZTWM) parameter as function of the number of SAC-SMA model evolutions using the a) RWM, b) DRAM (c), DE-MC, (d) DREAM, and (e) Shuffled Complex Evolution (SCE-UA) global optimization algorithm. The SCE-UA method was developed by [35] in the early 1990s to find the global optimum for highly nonlinear, non-convex, and non-continuous d-dimensional parameter spaces of typical watershed models. The bottom panel (Fig. 3e) depicts the evolution of the mean RMSE value derived with the RWM (purple), DRAM (cyan), DE-MC (blue), DREAM (red) and SCE-UA (green) methods. The results presented in this figure highlight two important observations.

First, DREAM exhibits superior performance. The N = 13 chains have converged to a limiting distribution within 40,000 SAC-SMA model evaluations with a good mixing of the individual trajectories. The DE-MC exhibits a decent performance as well, but suggests bimodality of the target with a number of paths that maintain their presence at the upper bound of UZTWM. Long-run performance however, shows that these respective Markov chains are stuck in an area with negligible probability mass [38]. This prohibits convergence of DE-MC [32]. The performance of the RWM sampler is rather disappointing. All N = 5 independent chains have converged to the wrong limiting distribution, even though the proposal distribution has been chosen optimal. A similar problem is observed with DRAM, where the different chains have been captured in different basins of attraction with RMSE values ranging between 14.0 and 14.5 (m^3/s). These chains are unable to traverse the target. These results clearly demonstrate the need for explicit handling of aberrant trajectories.

Second, it is interesting to observe that DREAM is not only superior to the RWM, DRAM and DE-MC sampling schemes, but also outperforms the widely used SCE-UA global optimization algorithm. The bottom panel clearly shows that DREAM not only exhibits the fastest decline of the objective function, but also finds the minimum overall RMSE value of about 13.25 (m^3/s). This value is significantly lower than its counterpart of approximately 13.70 (m^3/s) separately derived with SCE-UA. This shows that SCE-UA has converged prematurely to a sub-optimal region in the parameter space, contradicting many published studies in the literature that have shown that SCE-UA is a reliable and efficient optimizer of nonlinear watershed models. Using different settings for the algorithmic parameters in the various methods yielded very similar traces as those depicted in Fig. 3. We conclude that DREAM enhances the efficiency of MCMC simulation, and simultaneously estimates values of the SAC-SMA model parameters that improve the reliability of flood forecasts.

Conclusions

In this paper we have shown that significant improvements to the efficiency of MCMC simulation can be made by running multiple interacting chains simultaneously using differential evolution. This method entitled, DREAM automatically tunes the scale and orientation of the proposal distribution during the search, adapts subspace sampling to maximize the average normalized jumping distance in each chain, and explicitly handles outlier chains to avoid convergence problems on difficult response surfaces with numerous local optimal solutions. Detailed balance and ergodicity of the algorithm have been proved, and various case studies covering a wide range of problem features have shown that DREAM is generally superior to existing MCMC schemes, and can efficiently handle problems involving high-dimensionality, multimodality, nonlinearity, and local optimas.

Acknowledgments

The first author is supported by a J. Robert Oppenheimer Fellowship of the Los Alamos Postdoctoral Program. The source code of DREAM is written in MATLAB and sequential and parallel implementations of this software can be obtained from the first author (vrugt@lanl.gov) upon request.

References

[1] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *J Chem Phys* 21:1087-1092.

[2] Gelfand AE, Smith AF (1990) Sampling based approaches to calculating marginal densities. *J Amer Stat Assoc* 85: 398409.

[3] Frenkel D (2004) Speedup of Monte Carlo simulations by sampling of rejected states. *Proc Natl Acad Sci USA* 101:17571-17575.

[4] Frenkel D, Smit B (2002) Understanding Molecular Simulations: From Algorithms to Applications (Academic, San Diego) 2nd ed.

[5] Tse YK, Zhang, XB, Yu J (2004) Estimation of hyperbolic diffusion using the Markov chain monte Carlo method. *Quan Finan* 4:158-169.

[6] Chib S, Greenberg E (1996) Markov chain Monte Carlo simulation methods in econometrics. *Econ Theor* 12:409-431.

[7] Nielsen R,Wakeley J (2001) Distinguishing migration from isolation: a Markov chain Monte Carlo approach. *Genetics* 158:885-896.

[8] Tierney L (1994) Markov chains for exploring posterior distributions. *Ann Stat* 22:1701--1728.

[9] Beerli P, Felsenstein J (2001) Maximum likelihood estimation of migration rates and effective population numbers in two populations using a coalescent approach. *Proc Natl Acad Sci USA* 98:4563-4568.

[10] Kuczera G, Parent E (1998) Monte Carlo assessment of parameter uncertainty in conceptual catchment models. *J Hydrol* 211:69-85.

[11] Lambert DM, Richie PA, Millar CD, Holland B, Drummond AJ, Baroni C (2002) Rates of evolution in ancient DNA from Adélie penguins. *Science* 295:2270-2273.

[12] Huelsenbeck JP, Ronquist F (2001) MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* 17:754-755.

[13] Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57:97-109.

[14] Owen AB, Tribble SD (2005) A quasi-Monte Carlo Metropolis algorithm. *Proc Natl. Acad Sci USA* 102:8844-8849.

[15] Haario H, Saksman E, Tamminen J (1999) Adaptive proposal distribution for random walk Metropolis algorithm. *Comp Stat* 14:375-395.

[16] Haario H, Saksman E, Tamminen J (2001) An adaptive Metropolis algorithm. *Bernoulli* 7:223-242.

[17] Gelman AG, Roberts GO, Gilks WR (1996) in Bayesian Statistics, Oxford University Press pp. 599-608.

[18] Haario H, Saksman E, Tamminen J (2005) Componentwise adaptation for high dimensional MCMC. *Comp Stat* 20:265-274.

[19] Haario H, Laine M, Mira A, Saksman E (2006) DRAM: Efficient adaptive MCMC. *Stat Comp* 16:339-354.

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

[21] Gilks WR, Roberts GO, George EI (1994) Adaptive direction sampling. *Statistician* 43:179-189.

[22] Warnes GR (2001) Techical Report 3, Department of Statistics, University of Washington, Seattle, WA, USA pp. 135.

[23] Chauveau D, Vandekerkhove P (2002) Improving convergence of the Hastings-Metropolis algorithm with a learning proposal. *Scan J Stat* 29:13-29.

[24] Mykland P, Tierney L, Yu B (1995) Regeneration in Markov chain samplers. *J Amer Stat Assoc* 90:233-241.

[25] Gilks WR, Roberts G, Sahu S (1998) Adaptive Markov chain Monte Carlo through regeneration. *J Amer Stat Assoc* 93:1045-1054.

[26] Sahu SK, Zhigljavsky AA (2003) Self regenerative Markov chain Monte Carlo with adaptation. *Bernoulli* 9:395-422.

[27] Ter Braak CJF (2006) A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. *Stat Comp* 16:239-249.

[28] Gelman AG, Rubin DB (1992) Inference from iterative simulation using multiple sequences. *Stat Sci* 7:457-472.

[29] Vrugt JA, Robinson BA (2007) Improved evolutionary optimization from genetically adaptive multimethod search. *Proc Natl Acad Sci USA* 104:708-711.

[30] Pasarica C, Gelman A (2005) Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Stat Sin* xxxx, Under Revision.

[31] Mengersen K, Robert CP (2003) in Bayesian Statistics, Oxford University Press pp. 277-292.

[32] ter Braak CJF, Vrugt JA (2008) Differential Evolution Markov Chain with snooker updater and fewer chains. *Stat Comp*, doi:10.1007/s11222-008-9104-9.

[33] Roberts GO, Gelman A, Gilks WR (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann Appl Prob* 7:110-120.

[34] Del Moral P, Doucet A, Jasra A (2006) Sequential Monte Carlo samplers. *J R Statist Soc B* 68(3):411-436.

[35] Duan Q, Gupta VK, Sorooshian S (1992) Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour Res* 28:1015-1031.

[36] Box GEP, Tiao GC (1973) Bayesian Inference in Statistical Analyses (Addison-Wesley-Longman, Reading, Mass.)

[37] Vrugt JA, Gupta HV, Bouten W, Sorooshian S (2003) A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour Res* 39:1201, doi:10.1029/2002WR001642.

[38] Vrugt JA, C.J.F. ter Braak (2008) DREAM and sampling from the past: Efficient Markov Chain Monte Carlo simulation. In Prep.

Table 1: Performance of DREAM using different combinations of n_{CR} and δ for the 100-dimensional multivariate normal distribution with an underdispersed initial distribution The variable n_{CR} determines how many different crossover values, $CR = \{1/n_{CR}, 2/n_{CR}, \ldots, 1\}$ are being used to generate candidate points, whereas δ determines the number of pairs of chains to generate proposals. We ran DREAM without (left-side) and with (right-side) delayed rejection, and with (top part) and without (bottom part) adaptation of *CR*. The column headings *D*, *FE* and *AR* denote the average normalized Euclidean distance to the true mean and standard deviation of the target distribution ($\cdot 10^{-2}$), the number of function evaluations ($\cdot 10^{5}$) needed to reach convergence ($\hat{R}_j < 1.2$ for all dimensions [27]) and the acceptance rate (%). Listed values represent averages over 100 independent trials.

	Without Delayed Rejection									With Delayed Rejection								
n_{CR}	$\delta = \{1\}$			$\delta = \{1,2\}$			$\delta = \{1,2,3\}$			$\delta = \{1\}$			$\delta = \{1,2\}$			$\delta = \{1,2,3\}$		
	D	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*
1	74.3	2.31	24.0	69.5	2.36	24.0	71.6	2.34	24.0	90.5	3.84	38.6	92.6	3.84	38.6	90.1	3.92	38.6
2	3.57	3.89	17.7	3.77	3.80	17.8	3.80	3.94	17.7	4.86	6.62	34.5	4.93	6.47	34.5	4.94	6.45	34.5
3	3.71	4.26	17.0	3.75	4.24	17.0	3.73	4.23	17.1	4.73	7.19	33.9	4.72	7.15	33.9	4.83	7.17	33.9
5	4.02	4.71	16.5	4.23	4.68	16.5	3.83	4.75	16.5	4.85	7.99	33.4	5.17	7.76	33.4	4.74	7.90	33.4
10	4.11	5.12	16.2	4.24	5.15	16.2	3.93	5.10	16.2	5.48	8.46	32.9	5.19	8.80	32.9	5.23	8.42	32.9
n_{CR}	Without CR adaptation									Without CR adaptation								
2	4.06	4.07	17.3	3.90	4.06	17.3	3.80	4.12	17.3	5.04	7.15	34.0	5.25	7.15	34.0	5.09	7.22	34.0
3	4.10	4.57	16.6	4.00	4.51	16.6	3.86	4.51	16.7	5.10	8.20	33.3	5.60	8.09	33.3	5.48	8.00	33.1
5	4.08	5.09	16.1	4.07	4.98	16.0	3.66	5.06	16.1	5.45	8.90	32.8	5.44	8.91	32.8	5.34	8.86	32.8
10	4.13	5.38	15.8	4.07	5.61	15.8	4.15	5.51	15.9	5.34	9.47	32.6	5.51	9.68	32.6	5.43	9.51	32.5
CR	Single fixed *CR* value									Single fixed *CR* value								
1/2	4.29	7.44	15.1	4.98	7.38	15.0	4.28	7.35	15.1	7.69	*N/C*	31.9	8.26	*N/C*	31.9	7.54	*N/C*	31.9
1/3	6.31	*N/C*[†]	14.5	5.47	*N/C*	14.4	5.28	*N/C*	14.6	16.9	*N/C*	31.7	17.5	*N/C*	31.7	15.7	*N/C*	31.7

[†] More than 50% of the runs did not converge within the total number of 1,000,000 function evaluations.

Table 2: Performance of DREAM using different combinations of n_{CR} and δ for the 10-dimensional nonlinear twisted Gaussian distribution with (a) $b = 0.1$, and (b) $b = 0.01$. The content of this Table is similar to Table 1, and variables have been defined there. Listed values represent averages over 100 independent trials.

	(a) $b = 0.1$; highly nonlinear ($D{\cdot}10^{-2}$; $FE{\cdot}10^{4}$)									(b) $b = 0.01$; mildly nonlinear ($D{\cdot}10^{-2}$; $FE{\cdot}10^{3}$)								
n_{CR}	$\delta = \{1\}$			$\delta = \{1,2\}$			$\delta = \{1,2,3\}$			$\delta = \{1\}$			$\delta = \{1,2\}$			$\delta = \{1,2,3\}$		
	D	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*	*D*	*FE*	*AR*
1	23.3	2.69	16.7	21.4	2.90	14.4	21.8	3.75	13.4	17.7	7.26	28.1	17.1	7.66	27.8	17.2	7.12	27.8
2	9.94	3.09	9.13	9.19	4.45	8.53	8.81	4.60	8.23	5.11	6.28	19.9	5.41	6.04	19.7	6.09	5.85	19.7
3	9.75	3.31	11.1	9.79	4.47	10.5	8.31	3.54	10.1	5.04	5.53	20.5	6.13	6.26	20.4	5.79	5.91	20.4
5	10.1	4.62	12.9	8.71	3.96	12.4	8.38	5.03	12.2	5.02	6.35	21.4	5.48	5.64	21.2	5.22	5.78	21.1
10	9.35	3.75	14.1	8.63	4.07	13.7	9.00	4.96	13.6	5.63	6.54	22.2	4.88	5.97	21.9	4.82	6.33	21.8

Table 3: Comparison of DREAM (n_{CR} = 3; δ = {1,2,3}) against the Random Walk Metropolis (RWM), Delayed Rejection Adaptive Metropolis (DRAM) and DiffeRential Evolution-Markov Chain (DE-MC) MCMC schemes for the 10-dimensional twisted Gaussian distribution with b = 0.1. N denotes the number of parallel chains; FE = $N\cdot$ (number of generations) = number of function evaluations in an N chain sampler needed to reach convergence and AR the acceptance rate (%). The reported values represent averages over 100 independent trials.

	N	D	FE $(\cdot 10^4)$	AR (%)
RWM	1	0.12	6.71	24.3
DRAM	1	0.08	4.82	51.6
DE-MC	20	0.09	9.45	6.33
DREAM†	10	0.09	4.01	9.50
DREAM	10	0.08	3.54	10.1

† n_{CR} = 3 without adaptation, $p_1 = p_2 = p_3$ = 1/3.

Table 4: Comparison of DREAM against the RWM, DRAM and DE-MC schemes for posterior exploration of the 10-dimensional bimodal mixture function. The values represent averages over 100 independent trials. For symbols see Table 3.

	N	*D*	*FE* $(\cdot 10^4)$	AR (%)
RWM	1	0.99	*N/C*†	24.6
DRAM	1	0.96	*N/C*†	57.4
DE-MC	20	0.05	4.95	12.5
DREAM	10	0.04	2.56	11.7

† None of the runs have converged within 10^6 generations.

Table 5: Description of the SAC-SMA model parameters, including their prior and 95% posterior uncertainty intervals derived with DREAM.

Parameter	Description	Units	Prior	Posterior
	Capacity thresholds			
UZTWM	upper zone tension water maximum storage	[mm]	1.0 – 150.0	19.4 – 45.7
UZFWM	upper zone free water maximum storage	[mm]	1.0 – 150.0	16.2 – 33.8
LZTWM	lower zone tension water maximum storage	[mm]	1.0 – 500.0	224.9 – 275.3
LZFPM	lower zone free water primary maximum storage	[mm]	1.0 – 1000.0	80.7 – 127.4
LZFSM	lower zone free water supplemental maximum storage	[mm]	1.0 – 1000.0	27.7 – 88.4
ADIMP	additional impervious area	[-]	0.0 – 0.40	0.23 – 0.36
	Recession parameters			
UZK	upper zone free water lateral depletion rate	[day^{-1}]	0.1 – 0.5	0.28 – 0.49
LZPK	lower zone primary free water depletion rate	[day^{-1}]	0.0001 – 0.025	0.015 – 0.025
LZSK	lower zone supplemental free water depletion rate	[day^{-1}]	0.01 – 0.25	0.22 – 0.25
	Percolation and other			
ZPERC	maximum percolation rate	[-]	1.0 – 250.0	144.7 – 248.9
REXP	exponent of the percolation equation	[-]	0.0 – 5.0	2.91 – 4.84
PCTIM	impervious fraction of the watershed area	[-]	0.0 – 0.1	$7.7\cdot10^{-5}$ – 0.011
PFREE	fraction percolating from upper to lower zone free water storage	[-]	0.0 – 0.1	0.11 – 0.22
	Not optimized			
RIVA	riparian vegetation area	[-]	0.0	
SIDE	ratio of deep recharge to channel base flow	[-]	0.0	
RSERV	fraction of lower zone free water not transferable to tension water	[-]	0.3	

Table 6: Number of parallel chains, N, and obtained probability distributions P for the crossover probability CR in the various case studies ($P=\{p_m \mid p_m = \mathrm{Prob}(CR = 1/m), m = 1,\ldots,n_{CR}\}$ with $n_{CR} = 3$ so that $CR = \{1/3, 2/3, 1\}$).

Case Study 1: 100-d normal distribution with underdispersed initial distribution

$N = 100 \rightarrow P = \{0.28, 0.28, 0.44\}$

Case Study 1: 100-d normal distribution with overdispersed initial distribution

$N = 100 \rightarrow P = \{0.40, 0.42, 0.18\}$

Case Study 2a: 10-d twisted Gaussian with b = 0.1

$N = 10 \rightarrow P = \{0.45, 0.32, 0.23\}$

Case Study 2b: 10-d twisted Gaussian with b = 0.01

$N = 10 \rightarrow P = \{0.36, 0.37, 0.27\}$

Case Study 3: 10-d bimodal mixture distribution

$N = 10 \rightarrow P = \{0.16, 0.18, 0.66\}$

Case Study 4: 13-d SAC-SMA watershed model

$N = 13 \rightarrow P = \{0.37, 0.28, 0.35\}$

Figure Captions

Figure 1: Simulated traces for the 100 dimensional Gaussian target distribution with correlated dimensions using the (a,b) RWM, (c,d) DRAM, (e,f) DE-MC and (g,h) DREAM sampling schemes initialized with an overdispersed (left column) and underdispersed (right column) prior distribution. The blue and green line depict the evolution of the sampled standard deviations of x_{100} and x_1, respectively, whereas the red and black line denote the evolution of the Cov(x_1,x_{100}) and mean of sampled x_1 values, respectively. The true values of these four entities are separately indicated with different symbols at the right hand side in each panel.

Figure 2: Results of DREAM for the 10-dimensional bimodal mixture model. This target function is notoriously difficult to sample from using standard MCMC schemes with for d = 10 a distance of $10\sqrt{10}$ between the modes; (a) Sampled x_1 values with DREAM in the $N = d =$ 10 different chains. Each of the chains is coded with a different color, and (b) estimated marginal posterior probability distribution of x_1. The black line depicts the true bimodal target distribution.

Figure 3: Evolution of sampled values of the upper zone tension water maximum storage (UZTWM) parameter (in mm) with the (a) RWM, (b) DRAM, (c) DE-MC, (d) DREAM Markov chain Monte Carlo sampling schemes, and (e) SCE-UA global optimization algorithm. Each trajectory in panels (a)-(d) is coded with a different color and symbol. The lines in the bottom panel (f) depict the evolution of the mean Root Mean Square Error (RMSE) value derived with the RWM (purple), DRAM (cyan), DE-MC (blue), DREAM (red), and SCE-UA (green) algorithms.

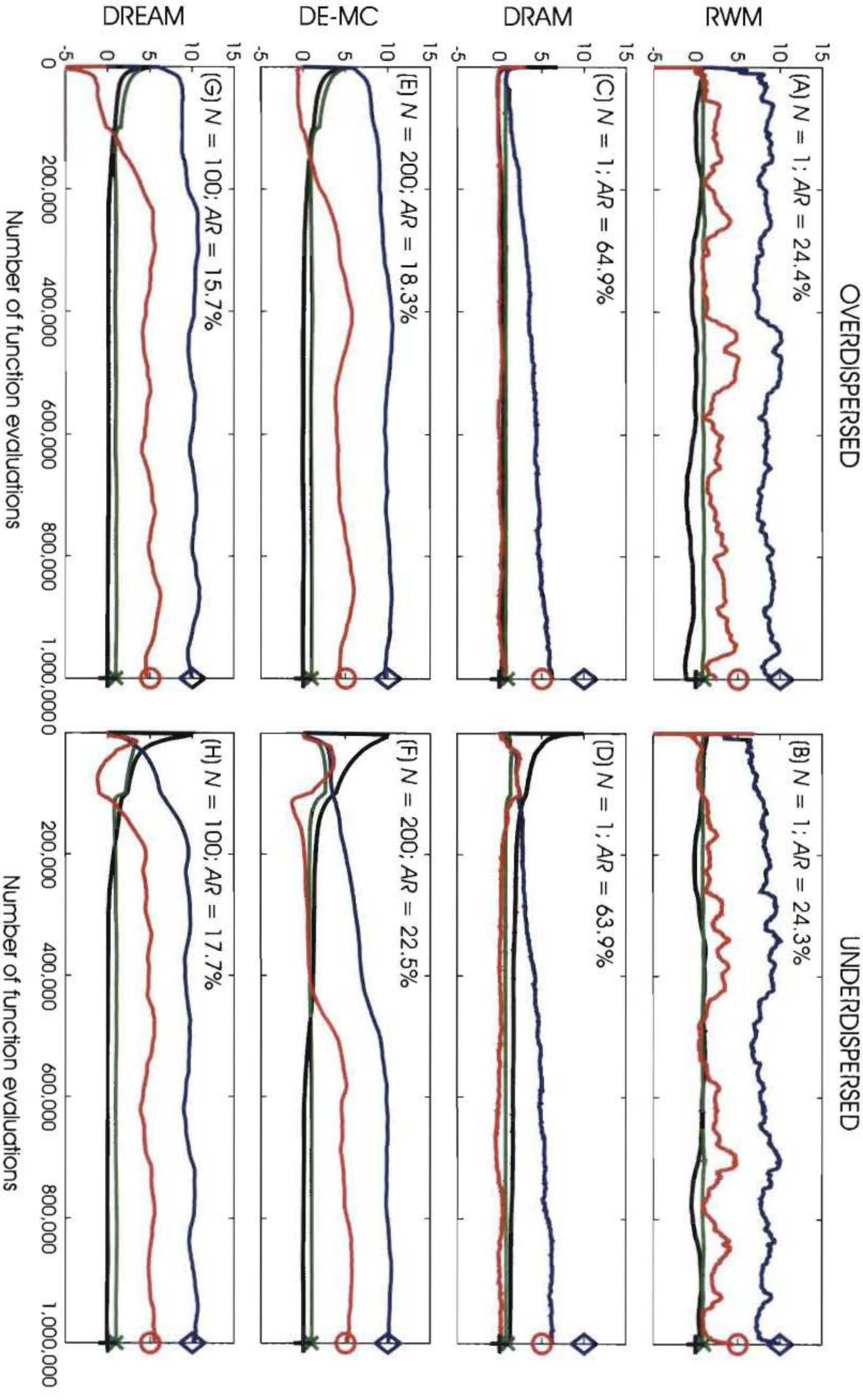
OVERDISPERSED
UNDERDISPERSED
RWM
DRAM
DE-MC
DREAM
(A) N = 1; AR = 24.4%
(B) N = 1; AR = 24.3%
(C) N = 1; AR = 64.9%
(D) N = 1; AR = 63.9%
(E) N = 200; AR = 18.3%
(F) N = 200; AR = 22.5%
(G) N = 100; AR = 15.7%
(H) N = 100; AR = 17.7%
15
10
5
0
-5
0
200,000
400,000
600,000
800,000
1,000,000
Number of function evaluations
Number of function evaluations

Figure 1.

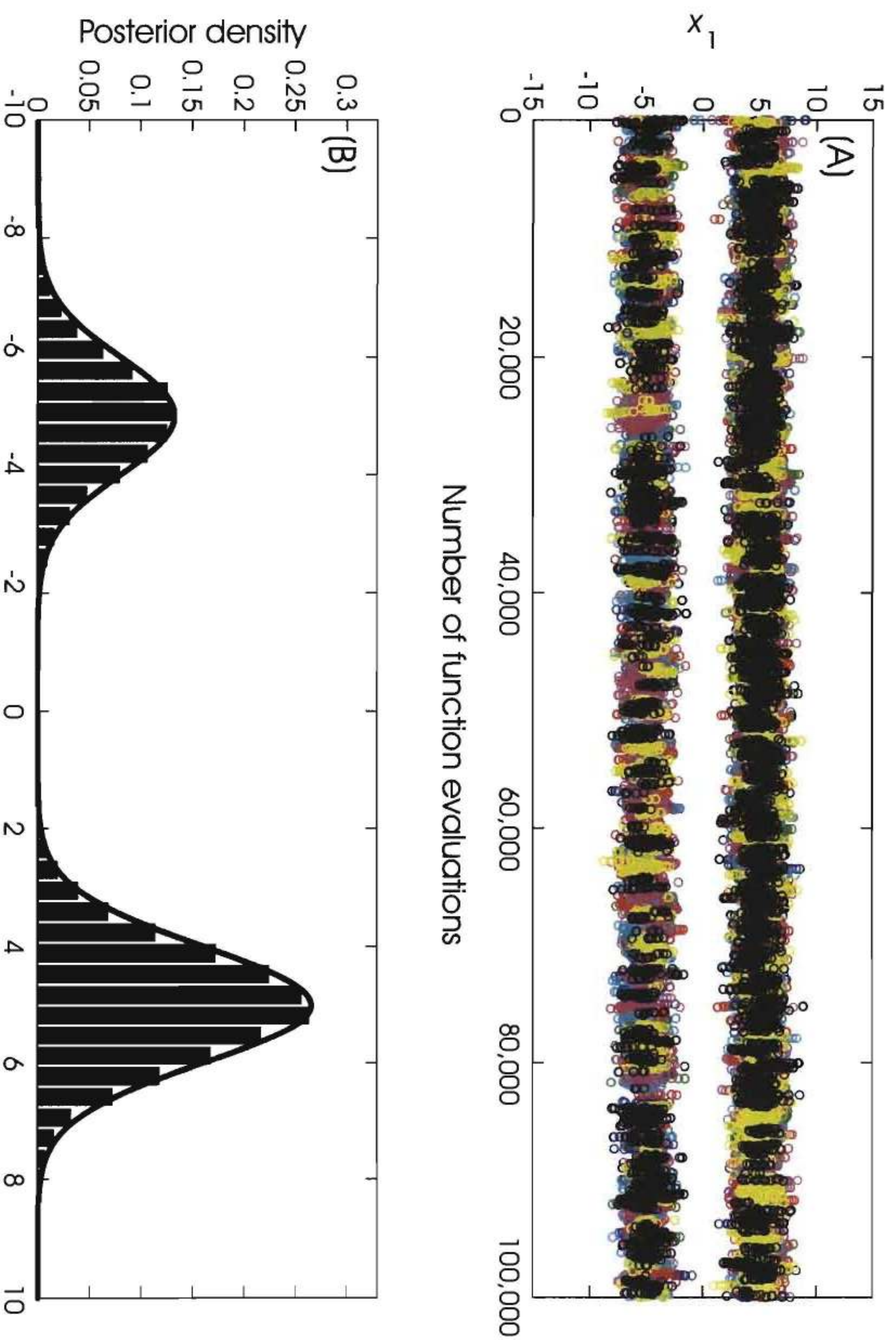
(A)
x_1
15
10
5
0
-5
-10
-15
0
20,000
40,000
60,000
80,000
100,000
Number of function evaluations
(B)
Posterior density
0.3
0.25
0.2
0.15
0.1
0.05
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
x_1

Figure 2.

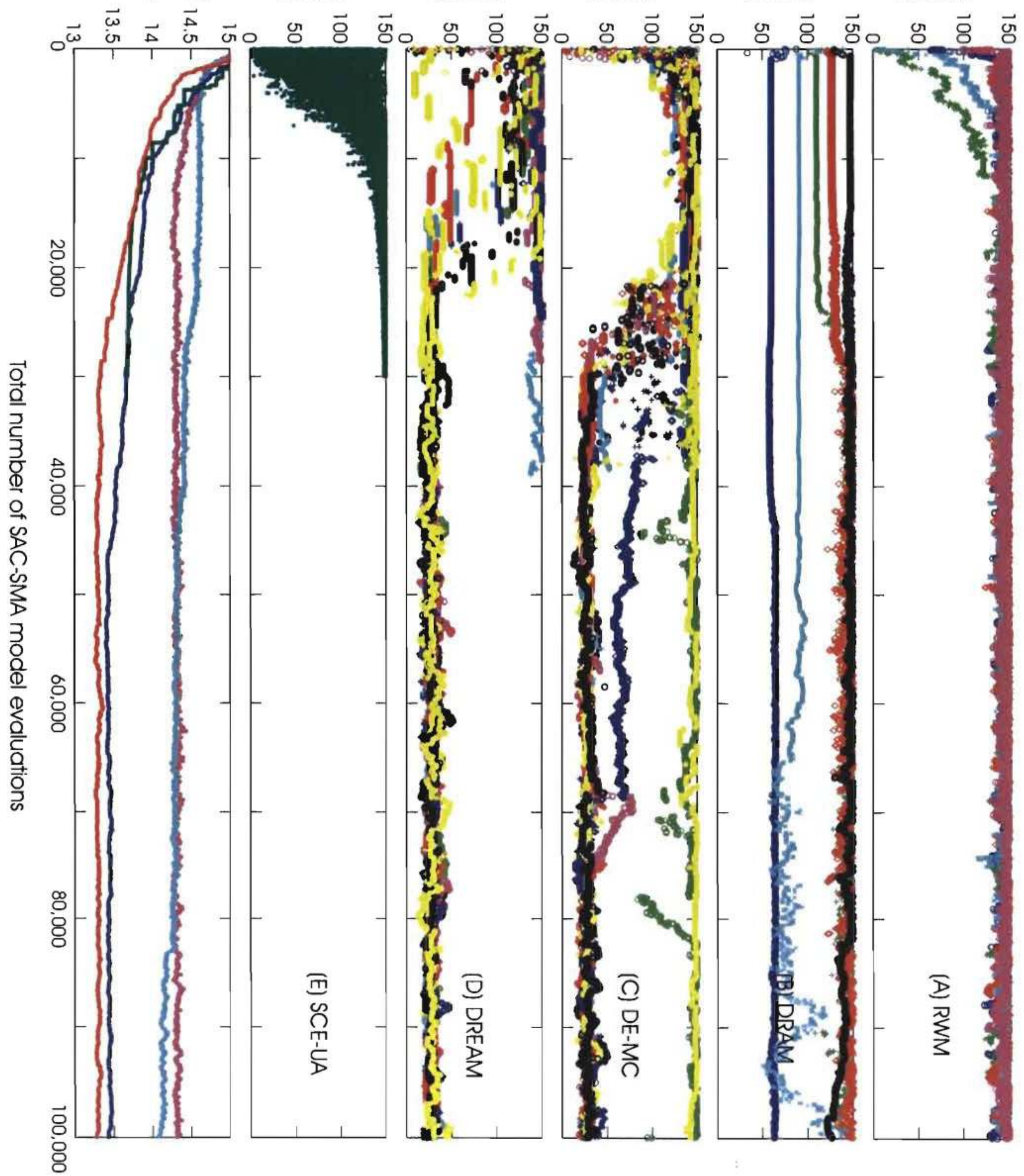
UZTWM
UZTWM
UZTWM
UZTWM
UZTWM
RMSE [m^3/s]
0
50
100
150
13
13.5
14
14.5
15
0
20,000
40,000
60,000
80,000
100,000
Total number of SAC-SMA model evaluations
(A) RWM
(B) DRAM
(C) DE-MC
(D) DREAM
(E) SCE-UA

Figure 3.