Hydrol. Earth Syst. Sci., 15, 3701–3713, 2011
www.hydrol-earth-syst-sci.net/15/3701/2011/
doi:10.5194/hess-15-3701-2011

Hydrology and Earth System Sciences

DREAM$_{(D)}$: an adaptive Markov Chain Monte Carlo simulation algorithm to solve discrete, noncontinuous, and combinatorial posterior parameter estimation problems

J. A. Vrugt[1,2] **and C. J. F. Ter Braak**[3]

[1]Department of Civil and Environmental Engineering, University of California, Irvine, 4130 Engineering Gateway, Irvine, CA 92697-2175, USA
[2]Institute for Biodiversity and Ecosystem Dynamics, University of Amsterdam, Amsterdam, The Netherlands
[3]Biometris, Wageningen University and Research Centre, P.O. Box 100, 6700 AC, Wageningen, The Netherlands

Received: 19 April 2011 – Published in Hydrol. Earth Syst. Sci. Discuss.: 26 April 2011
Revised: 27 November 2011 – Accepted: 3 December 2011 – Published: 13 December 2011

Abstract. Formal and informal Bayesian approaches have found widespread implementation and use in environmental modeling to summarize parameter and predictive uncertainty. Successful implementation of these methods relies heavily on the availability of efficient sampling methods that approximate, as closely and consistently as possible the (evolving) posterior target distribution. Much of this work has focused on continuous variables that can take on any value within their prior defined ranges. Here, we introduce theory and concepts of a discrete sampling method that resolves the parameter space at fixed points. This new code, entitled DREAM$_{(D)}$ uses the recently developed DREAM algorithm (Vrugt et al., 2008, 2009a,b) as its main building block but implements two novel proposal distributions to help solve discrete and combinatorial optimization problems. This novel MCMC sampler maintains detailed balance and ergodicity, and is especially designed to resolve the emerging class of optimal experimental design problems. Three different case studies involving a Sudoku puzzle, soil water retention curve, and rainfall – runoff model calibration problem are used to benchmark the performance of DREAM$_{(D)}$. The theory and concepts developed herein can be easily integrated into other (adaptive) MCMC algorithms.

Correspondence to: J. A. Vrugt
(jasper@uci.edu)

1 Introduction

Formal and informal Bayesian methods have found widespread application and use to summarize parameter and model predictive uncertainty in hydrologic modeling. These parameters generally represent model dynamics, but could also include rainfall multipliers (Kavetski et al., 2006; Kuczera et al., 2006; Vrugt et al., 2008), error model variables (Smith et al., 2008; Schoups and Vrugt, 2010), and calibration data measurement errors (Sorooshian and Dracup, 1980; Schaefli et al., 2007; Vrugt et al., 2008). Monte Carlo methods are admirably suited to generate samples from the posterior parameter distribution, but generally inefficient when confronted with complex, multimodal, and high-dimensional model-data synthesis problems. This has stimulated the development of Markov Chain Monte Carlo (MCMC) methods that generate a random walk through the search (parameter) space and iteratively visit solutions with stable frequencies stemming from an invariant probability distribution. If well designed, such MCMC methods should be more efficient than brute force Monte Carlo or importance sampling methods.

To visit configurations with a stable frequency, an MCMC algorithm generates trial moves from the current position of the Markov chain $\mathbf{x}_t$ to a new state $\mathbf{z}$. The earliest MCMC approach is the random walk Metropolis (RWM) algorithm (Metropolis et al., 1953). Assume that we have already sampled points $\{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$ this algorithms proceeds in the following three steps. First, a candidate point $\mathbf{z}$ is sampled from a proposal distribution $q(\cdot)$ that depends on the present location $\mathbf{x}_t$. Next, the candidate point is accepted with acceptance probability, $\alpha(\mathbf{z},\mathbf{x}_t)$ (Metropolis et al., 1953; Hastings, 1970):

$$\alpha(\mathbf{z},\mathbf{x}_t) = 1 \wedge \frac{p(\mathbf{z})}{p(\mathbf{x}_t)} \frac{q(\mathbf{z} \rightarrow \mathbf{x}_t)}{q(\mathbf{x}_t \rightarrow \mathbf{z})}, \tag{1}$$

where $p(\cdot)$ represents the posterior density, and $q(\mathbf{x}_t \rightarrow \mathbf{z})$ ($q(\mathbf{z} \rightarrow \mathbf{x}_t)$) denotes the conditional probability of the forward (backward) jump. This last ratio cancels out if a symmetric proposal distribution is used. Finally, if the proposal is accepted the chain moves to $\mathbf{z}$, otherwise the chain remains at its current location $\mathbf{x}_t$. Following a so called burn-in period (of say, l steps), the chain approaches its stationary distribution and the vector $\{\mathbf{x}_{l+1},\ldots,\mathbf{x}_{l+m}\}$ contains samples from $\pi(\cdot)$. The desired summary of the posterior distribution, $\pi(\mathbf{x})$ is then obtained from this sample of m points. In Bayesian applications, $\pi(\cdot)$ is the distribution of partially unknown parameters given the data at hand, and is obtained by combining the prior distribution and the data likelihood. The dependence of $\pi(\cdot)$ on any fixed data is assumed throughout.

The standard RWM algorithm has been designed to maintain detailed balance with respect to $\pi(\cdot)$ at each individual step in the chain:

$$\pi(\mathbf{x}_t)p(\mathbf{x}_t \rightarrow \mathbf{z}) = \pi(\mathbf{z})p(\mathbf{z} \rightarrow \mathbf{x}_t) \tag{2}$$

where $\pi(\mathbf{x}_t)$ ($\pi(\mathbf{z})$) denotes the probability of finding the system in state $\mathbf{x}_t(\mathbf{z})$, and $p(\mathbf{x}_t \rightarrow \mathbf{z})$ ($p(\mathbf{z} \rightarrow \mathbf{x}_t)$) denotes the conditional probability to perform a trial move from $\mathbf{x}_t$ to $\mathbf{z}$ ($\mathbf{z}$ to $\mathbf{x}_t$). The detailed balance condition essentially ensures that the samples of $\{\mathbf{x}_{l+1},\ldots,\mathbf{x}_{l+m}\}$ are exactly distributed according to the target distribution, $\pi(\mathbf{x})$.

Existing theory and experiments prove convergence of well-constructed MCMC schemes to the appropriate limiting distribution under a variety of different conditions. In practice, this convergence is often observed to be impractically slow. This deficiency is frequently caused by an inappropriate selection of $q(\cdot)$ used to generate trial moves in the Markov Chain. This inspired Vrugt et al. (2008, 2009a,b) to develop a simple adaptive RWM algorithm called Differential Evolution Adaptive Metropolis (DREAM) that runs multiple chains simultaneously for global exploration, and automatically tunes the scale and orientation of the proposal distribution during the evolution to the posterior distribution. This scheme is an adaptation of the Shuffled Complex Evolution Metropolis (Vrugt et al., 2003) global optimization algorithm and has the advantage of maintaining detailed balance and ergodicity while showing excellent efficiency on complex, highly nonlinear, and multimodal target distributions (Vrugt et al., 2008, 2009a,b).

In DREAM, N different Markov Chains are run simultaneously in parallel. If the state of a single chain is given by a single d-dimensional vector $\mathbf{x}$, then at each generation the N chains in DREAM define a population $\mathbf{X}$, which corresponds to an N x d matrix, with each chain as a row. Jumps in each chain $i = \{1,\ldots,N\}$ are generated by adding a multiple of the difference of the states of randomly chosen pairs of chains of $\mathbf{X}$ to the current state $\mathbf{x}^i$:

$$\mathbf{z}^i = \mathbf{x}^i + (\mathbf{1}_d + \mathbf{e}_d)\gamma(\delta,d')\left(\sum_{j=1}^{\delta}\mathbf{x}^{r_1(j)} - \sum_{n=1}^{\delta}\mathbf{x}^{r_2(n)}\right) + \epsilon_d \tag{3}$$

where δ signifies the number of pairs used to generate the proposal, $\mathbf{x}^{r_1(j)}$ and $\mathbf{x}^{r_2(n)}$ are randomly selected without replacement from the population $\mathbf{X}_t^{-i}$ (the population without $\mathbf{x}_t^i$); $r_1(j), r_2(n) \in \{1,\ldots,N\}$ and $r_1(j) \neq r_2(n)$. The values of $\mathbf{e}_d$ and ϵ_d are drawn independently from $U_d(-b,b)$ and $N_d(0,b^*)$ with, typically, $b = 0.1$ and b^* small compared to the width of the target distribution, $b^* = 10^{-12}$ say. Not all dimensions of $\mathbf{x}^i$ need to be updated in each step, so some dimensions of $\mathbf{z}^i$ may be reset to those of $\mathbf{x}^i$. The value of the jump-size, γ, depends on δ and d', the number of dimensions updated jointly in the step. By comparison with RWM, a good choice for $\gamma = 2.4/\sqrt{2\delta d'}$ (Roberts and Rosenthal, 2001; Ter Braak, 2006). This choice is expected, for Gaussian and Student target distributions, to yield an acceptance probability of 0.44 for d' = 1, 0.28 for d' = 5 and 0.23 for large d'. Every 5th generation $\gamma = 1.0$ to facilitate jumping between disconnected posterior modes (Vrugt et al., 2008).

The difference vector in Eq. (3) contains the desired information about the scale and orientation of the target distribution, $\pi(\mathbf{x})$. By accepting each jump with the Metropolis ratio $\min\{\pi(\mathbf{z}^i)/\pi(\mathbf{x}^i),1\}$, a Markov chain is obtained, the stationary or limiting distribution of which is the posterior distribution. The proof of this is given in Ter Braak and Vrugt (2008) and Vrugt et al. (2008, 2009a,b). Because the joint pdf of the N chains factorizes to $\pi(\mathbf{x}^1) \times \ldots \times \pi(\mathbf{x}^N)$, the states $\mathbf{x}^1 \ldots \mathbf{x}^N$ of the individual chains are independent at any generation after DREAM has become independent of its initial value. After this burn-in period, the convergence of DREAM can thus be monitored with the $\hat{R}$-statistic of Gelman and Rubin (1992). This convergence diagnostic compares the within and in-between variances of the N different chains.

Although much progress has been made in the past decade towards the application of Bayesian methods to quantify and analyze parameter, model structural, forcing, and calibration data uncertainty, virtually all this research involved continuous parameters that can take on any value within their prior defined ranges. Much less attention has been given hitherto to posterior sampling problems involving discrete variables. Such non-continuous parameter estimation problems are not only of considerable theoretical interest, but also of practical significance. For instance, contributions are beginning to appear in the hydrologic literature that attempt to discern optimal experimental design strategies that minimize cost, parameter and model predictive uncertainty, or combinations thereof. This involves selecting one or multiple different measurement locations in time and/or space amongst a discrete set of possibilities. Various algorithms have been proposed in the applied mathematics and computer science literature that solve problems of this kind, yet their main focus is on finding the optimal solution, without recourse to estimating the underlying posterior uncertainty. This is of particular importance in experimental design problems, where one cannot convincingly claim that one set of measurements is significantly better than another plausible combination of observations.

In this paper we present a discrete implementation of DREAM, that is especially designed to efficiently retrieve the posterior distribution of noncontinuous and combinatorial search and optimization problems. This new code, hereafter referred to as $\text{DREAM}_{(\text{D})}$ uses DREAM as its main building block, and implements three novel proposal distributions to explicitly recognize differences in topology between discrete and Euclidean search spaces. This sampling method maintains detailed balance and ergodicity, and provides explicit information about the posterior uncertainty of the optimal solution. Within the context of optimal experimental design problems this would provide very valuable information about which measurements are absolutely required, and which data are of less importance. The width of the marginal posterior distribution essentially conveys this information.

The remainder of this paper is organized as follows. Section 2 presents a short introduction to discrete optimization problems, followed by a detailed description of $\text{DREAM}_{(\text{D})}$ in Sect. 3. Section 4 demonstrates the performance of $\text{DREAM}_{(\text{D})}$ using three different case studies involving a Sudoku puzzle, water retention curve and rainfall-runoff model calibration problem. These results illustrate the ability of $\text{DREAM}_{(\text{D})}$ to solve search and optimization problems involving discrete, combinatorial, and experimental design problems. Finally, in Sect. 5 we summarize the theory, concepts and applications presented herein.

2 Nonlinear optimization involving discrete variables

Discrete optimization problems are abundant in many fields of study, and have also started to begin to appear in the hydrologic literature (Furman et al., 2004; Harmancioglu et al., 2004; Perrin et al., 2008; Kleidorfer et al., 2008; Neuman et al., 2011). Figure 1 illustrates a simple discrete parameter estimation problem involving a tile puzzle. Each tile contains a different letter of the alphabet. The goal is to list the letters in the appropriate order. The solution to this problem is obvious to a human, but not immediately clear to a computer. A search algorithm is therefore required to solve this problem.

If we assign numbers to each letter, A = 1, B = 2 and so forth, we could measure the distance from our initial guess to the actual solution, and iteratively refine this solution by sampling from a (discrete) proposal distribution. Many algorithms have been developed in the past decades to efficiently resolve problems of this kind. Yet, the main trust of these algorithms is on finding a single optimum solution, without recourse to estimating the underlying posterior uncertainty. For example, within the context of the traveling salesman problem many different routes will exist that only deviate marginally from the optimal solution. Brute force sampling of the search space can be used to assess all these plausible routes, but this seems rather inefficient. Instead, a more intelligent search procedure is warranted that more efficiently samples the space of possible solutions. The goal of this paper therefore is to introduce theory and concepts of $\text{DREAM}_{(\text{D})}$, a MCMC simulation algorithm that is especially designed to efficiently retrieve the posterior distribution of discrete and combinatorial search problems.

Figure 1a illustrates the application of $\text{DREAM}_{(\text{D})}$ to the tile puzzle considered herein. The left panel depicts the initial guess, whereas the remaining three panels (Fig. 1b–d) illustrate how $\text{DREAM}_{(\text{D})}$ translates this random starting point into the final position of the letters. The next sections describe the theory and concepts of $\text{DREAM}_{(\text{D})}$ and present three different case studies.

3 $\text{DREAM}_{(\mathbf{D})}$ ⇒ DiffeRential Evolution Adaptive Metropolis with Discrete Sampling

We now describe our new code, entitled $\text{DREAM}_{(\text{D})}$, which uses DREAM as main building block. The algorithmic parameters (with defaults in parentheses) are N (d) the number of chains, δ (1) the number of pairs used to generate the proposal, CR the crossover probability (see later), K (10) the thinning rate in storing samples and $T_{\max}$ (10^6) the maximum number of generations, typically so large that $\text{DREAM}_{(\text{D})}$ automatically stops after convergence has been achieved.

Let $\mathbf{X}$ be a $N \times d$-matrix with rows $\mathbf{x}^i$, $i = 1,\ldots,N$, that contain the current states of the N different Markov chains and that are iteratively updated in turn in the algorithm. The initial $\mathbf{X} = [\mathbf{X}_t; t = 0]$ is obtained by drawing N samples from $p_d(\mathbf{x})$, the prior distribution. Also, let $\mathbf{S}$ be an external archive that stores the elements of $\mathbf{X}$ at regular intervals. The initial population $[\mathbf{X}_t; t = 0]$ is translated into a sample from the posterior target distribution using the following pseudo code:

1. Set $t \leftarrow 1$
   REPEAT K times (POPULATION EVOLUTION)
   FOR $i \leftarrow 1,\ldots,N$ DO (CHAIN EVOLUTION)

   (a) Draw the number of dimensions to be updated, d', from the binomial distribution with total d and probability CR. If $d' = 0$, set $d' \leftarrow 1$.

   (b) Generate a new proposal, $\mathbf{z}^i$, for chain i,

   $$\mathbf{z}^i = \mathbf{x}^i + \left\| (\mathbf{1}_d + \mathbf{e}_d)\gamma(\delta, d') \left(\sum_{j=1}^{\delta} \mathbf{X}^{r_1(j)} - \sum_{n=1}^{\delta} \mathbf{X}^{r_2(n)} \right) + \epsilon_d \right\|_d \tag{4}$$

   where the function $\|\cdot\|_d$ rounds each element $j = 1,\ldots,d$ of the jump vector to the nearest integer. The various symbols have been defined after Eq. (3).

   (c) Sample without replacement $d - d'$ numbers from $1,\ldots,d$. Denote each selected number generically by j

(A)

H	B	P	N
G	M	O	D
I	E	C	L
K	A	F	J

(B)

E	B	C	D
F	A	G	H
I	J	M	L
K	O	N	P

(C)

A	B	C	D
F	E	G	H
I	J	K	L
M	O	P	N

(D)

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Fig. 1. A 4×4 square tile puzzle with 16 different letters. The goal is to list the letters in order of the alphabet. Each letter is assigned a different integer value, and the resulting inverse problem is solved using DREAM$_{(\mathrm{D})}$. The left panel **(a)** presents the initial state, whereas the remaining three graphs **(b–d)** demonstrate how DREAM$_{(\mathrm{D})}$ iteratively solves this integer sampling problem.

and replace the corresponding proposal element z_j^i by x_j^i. The proposal $\mathbf{z}^i$ thus updates d' elements of $\mathbf{x}^i$ only.

(d) Compute $p(\mathbf{z}^i)$ and accept the candidate points with Metropolis acceptance probability, $\alpha(\mathbf{x}^i, \mathbf{z}^i)$,

$$\alpha(\mathbf{z}^i, \mathbf{x}^i) = 1 \wedge \frac{p(\mathbf{z}^i)}{p(\mathbf{x}^i)} \tag{5}$$

(e) If accepted, move the chain to the candidate point, $\mathbf{x}^i = \mathbf{z}^i$, otherwise remain at the old location, $\mathbf{x}^i$.

END FOR (CHAIN EVOLUTION)
END REPEAT (POPULATION EVOLUTION)

2. Append $\mathbf{X}$ to $\mathbf{S}$.
3. Compute the Gelman-Rubin convergence diagnostic, $\hat{R}_j$ (Gelman and Rubin, 1992) for each dimension $j = 1, \ldots, d$ using the last 50 % of the samples of $\mathbf{S}$.
4. If $\hat{R}_j \leq 1.2$ for $j = 1, \ldots, d$ or $t > T_{\max}$, stop and go to step 5, otherwise set $t \leftarrow t + K$ and go to POPULATION EVOLUTION.
5. Summarize the posterior pdf using $\mathbf{S}$ after discarding the initial and burn-in samples.

In words, DREAM$_{(\mathrm{D})}$ runs multiple different chains in parallel, and generates jumps in each individual chain using information from the location of the other $N-1$ chains. The rounding operator, $\|\cdot\|_d$ in Eq. (4) is used to enforce integer values and accommodate discrete search problems. To speed up convergence to the target distribution, DREAM$_{(\mathrm{D})}$ estimates a distribution of CR values during burn-in that favors large jumps over smaller ones in each of the N chains. Details can be found in previous studies (Vrugt et al., 2008, 2009a,b, 2011b).

The transition kernel of DREAM$_{(\mathrm{D})}$ is especially designed for integer variables. This appears to be a major limitation as many discrete search problems involve non-integer variables. For instance, consider the discrete variable $x_1 \in [0, 0.2, 0.4, \ldots, 5]$. We cannot sample this parameter with the proposal distribution presented in Eq. (4). A simple linear transformation of the search space, $x_1 = 0.2j$; $j \in \{0, \ldots, 25\}$ suffices to facilitate inference with DREAM$_{(\mathrm{D})}$. If appropriate, a different transformation can be used for each individual parameter. An extreme case would simultaneously involve discrete and continuous variables. This would constitute some combination of jumps generated with DREAM and DREAM$_{(\mathrm{D})}$.

3.1 DREAM$_{(\mathrm{D})}$ ⇒ detailed balance?

We are now left with a proof that DREAM$_{(\mathrm{D})}$ yields an invariant distribution that is identical to the posterior target of interest. For this we need to demonstrate that the transition kernel of DREAM maintains detailed balance, and thus results in a reversible Markov chain. In other words, the forward $p(\mathbf{x}^i \rightarrow \mathbf{z}^i)$ and backward $p(\mathbf{z}^i \rightarrow \mathbf{x}^i)$ jump should have equal probability at every single step in the chain. This is easy to proof for a standard RWM algorithm that uses a fixed proposal distribution. Hence, the forward and backward jump will exhibit equal probability. Yet, the jump distribution used in DREAM$_{(\mathrm{D})}$ continuously changes scale and orientation en route to the posterior target distribution. This adaptation significantly enhances search efficiency, but does this also ensure reversibility of the sampled Markov chains?

Previous manuscripts have provided formal proofs of convergence of DREAM (Vrugt et al., 2008), DREAM$_{(\mathrm{ZS})}$ (Vrugt et al., 2011b), and MT-DREAM$_{(\mathrm{ZS})}$ (Laloy and Vrugt, 2011) to the appropriate limiting distribution. These proofs appear rather simple, but might not be easy to understand for those that are not directly familiar with the underlying statistics and mathematics. We therefore resort to a simple hypothetical example. Lets consider a two-dimensional discrete sampling problem with $N = 5$ different chains, and thus $\gamma = 2.4/\sqrt{2\delta d'} = 2.4/\sqrt{2 \times 1 \times 2} = 1.2$. Without loss of generality, lets further assume that $\delta = 1$, and $e_j = \epsilon_j = 0$; $j = 1, \ldots, 2$. We randomly sample each individual chain from the prior parameter distribution, and color code their initial position in Fig. 2. We now use the transition kernel of DREAM$_{(\mathrm{ZS})}$ to generate candidate points. Lets assume that we select the blue and green chain as r_1 and r_2 respectively.

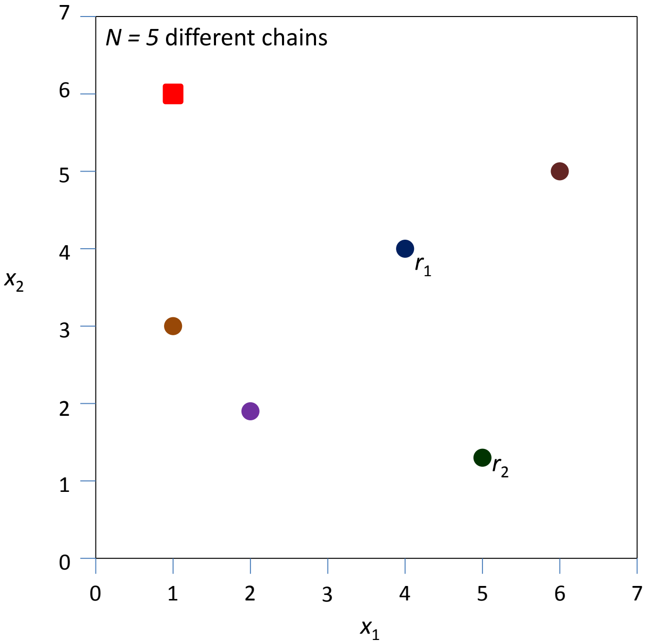

Fig. 2. Illustration of detailed balance: two-dimensional parameter estimation problem with $x_1 \in [0,7]$ and $x_2 \in [0,7]$. The color dots denote the starting points of the $N=5$ different chains. The red square signifies the candidate point of the first chain and is created by selecting the blue chain as r_1 and green chain as r_2. One step later, after moving to the red square, the backward jump has equal probability, as there is an equal chance of drawing r_1 (green) and r_2 (blue) in reversed order.

The candidate point of the first (purple) chain thus becomes:

$$\mathbf{z}^i = [2\,2] + \left\| 1.2([4\,4]-[5\,1]) \right\| = [2\,2] + \left\| [-1.2\,3.6] \right\| = [1\,6] \quad (6)$$

This proposal point is indicated in Fig. 2 with the red square. For the sake of this illustration, lets now assume that we accept this candidate point, and transition the first chain to this new state.

We now need to demonstrate that the reverse jump has equal probability. This backward jump can be obtained by selecting the blue and green chain in the opposite order from the forward jump:

$$\mathbf{z}^i = [1\,6] + \left\| 1.2([5\,1]-[4\,4]) \right\| = [1\,6] + \left\| [1.2\ -3.6] \right\| = [2\,2] \quad (7)$$

This point is exactly similar to the initial state of chain one prior to the forward jump. Apparently, the rounding operator used to sample integer values does not destroy detailed balance. By sampling r_1 and r_2 from the remaining $N-1$ chains with a uniform random number generator enforces symmetry of the proposal distribution. Indeed, the chance to select r_1 as chain two, and r_2 as chain four is equal to drawing $r_2 = 4$, and $r_2 = 2$ in the reverse order. This also holds when $\mathbf{e}_d$, and ϵ_d are drawn from their respective symmetric probability distributions, and when $\delta > 1$ and $d > 2$. We leave this up to the reader. This concludes our proof of detailed balance.

A final remark is appropriate. Theoretically, it is possible that at least one of the d arguments of the rounding function, $\|\cdot\|_d$ has a fractional part of .5. In this case, convention dictates to round down to the nearest integer. This directed rounding introduces a possible bias in jumping direction and thus strictly speaking violates detailed balance. The chance that this happens in practice is virtually zero. If nothing else, the stochastic nature of $\mathbf{e}_d \sim U_d(-b,b)$, and $\epsilon_d \sim N_d(0,b^*)$ will eliminate this possibility. But, in the rare event that any argument of $\|\cdot\|_d$ has a factional part of .5 we implement stochastic rounding to the nearest integer. This guarantees reversibility of the Markov chains generated with DREAM$_{(\mathrm{D})}$.

3.2 Combinatorial search problems: proposal distribution using position swapping

The parallel direction update of Eq. (4) works well for a range of discrete problems but is not necessary optimal for combinatorial problems in which the values of the final solution are known a-priori but not their order in the parameter vector. The topology of such search problems differs substantially from the Euclidean search problems considered hitherto. We therefore introduce an alternative jump distribution that creates candidate points by randomly swapping two coordinates in each individual chain.

PROPOSAL DISTRIBUTION: RANDOM SWAP
FOR $i \leftarrow 1,\ldots,N$ DO (CHAIN EVOLUTION)

1. Set $\mathbf{z}^i = \mathbf{x}^i$.
2. Randomly select two numbers j and k without replacement from $\{1,\ldots,d\}$.
3. Swap the j^{th} and k^{th} coordinates of $\mathbf{z}^i$; $y^i_j = x^i_k$ and $y^i_k = x^i_j$.
4. Compute $\pi(\mathbf{z}^i)$ and accept the candidate points with Metropolis acceptance probability, $\alpha(\mathbf{x}^i, \mathbf{z}^i)$.
5. If accepted, move the chain to the candidate point, $\mathbf{x}^i = \mathbf{z}^i$, otherwise remain at the old location, $\mathbf{x}^i$.

END FOR (CHAIN EVOLUTION)
END RANDOM SWAP

In words, if the current state of the ith chain is given by $\mathbf{x}^i = \left[x^i_1,\ldots,x^i_j,\ldots,x^i_k,\ldots,x^i_d\right]$ then the candidate point becomes $\mathbf{z}^i = \left[x^i_1,\ldots,x^i_k,\ldots,x^i_j,\ldots,x^i_d\right]$ where j,k are discrete uniformly sampled from $\{1,\ldots,d\}$ without replacement. It is straightforward to see that this proposal distribution satisfies detailed balance as the forward and backward jump have equal probability.

A swap update is admirably suited to solve the tile problem considered previously in Fig. 1. Yet, the coordinate swaps are fully random and do not exploit any information about the topology of the solution encapsulated in the position of the other $N-1$ chains. Essentially, each chain evolves independently to the posterior target distribution. This appears rather inefficient, particularly for complicated search problems. We therefore introduce a second and alternative swap rule that takes explicit information from the dissimilarities in coordinates of the N chains running in parallel. We implement this idea as follows:

PROPOSAL DISTRIBUTION: DIRECTED SWAP
DO (CHAIN EVOLUTION)

1. Randomly select two chains, $\mathbf{X}^{r_1}$ and $\mathbf{X}^{r_2}$ without replacement from the population $\mathbf{X}_t^{-r_1,r_2}$.
2. Find those coordinates of $\mathbf{x}^{r_1}$ and $\mathbf{x}^{r_2}$ that are dissimilar and store these locations in ζ.
3. Randomly permute ζ; set $j=\zeta_1$ and $k=\zeta_2$.
4. Copy $\mathbf{z}^{r_1}=\mathbf{x}^{r_1}$ and $\mathbf{z}^{r_2}=\mathbf{x}^{r_2}$.
5. Swap the elements of $\mathbf{z}^{r_1}$; $z_j^{r_1}=x_k^{r_1}$ and $z_k^{r_1}=x_j^{r_1}$.
6. Swap the elements of $\mathbf{z}^{r_2}$; $z_j^{r_2}=x_k^{r_2}$ and $z_k^{r_2}=x_j^{r_2}$.
7. Compute $\pi(\mathbf{z}^{r_1})$ and $\pi(\mathbf{z}^{r_2})$.
8. Accept both candidate points with probability, $p(\mathbf{z}^{r_1},\mathbf{z}^{r_2})$ equal to the product of their respective Metropolis ratios, $p(\mathbf{z}^{r_1},\mathbf{z}^{r_2})=\alpha(\mathbf{x}^{r_1},\mathbf{z}^{r_1})\alpha(\mathbf{x}^{r_2},\mathbf{z}^{r_2})$.
9. If accepted, move both chains to their respective candidate points, $\mathbf{x}^{r_1}=\mathbf{z}^{r_1}$ and $\mathbf{x}^{r_2}=\mathbf{z}^{r_2}$; otherwise remain at their old location, $\mathbf{x}^{r_1}$ and $\mathbf{x}^{r_2}$.

END FOR (CHAIN EVOLUTION)
END DIRECTED SWAP

This proposal distribution swaps location j and k within two different chains, $r_1,r_2\in\{1,\ldots,N\}$ from the differences in their coordinates. This update needs to be carefully implemented so that all different chains are updated only once. This requires that the number of chains, $N>1$ and hence is easiest to implement if N is an even number. This approach considerably improves sampling efficiency, as will be illustrated later with a Sudoku puzzle.

The swap move is fully Markovian, that is, it uses only information from the current time for proposal generation, and retains detailed balanced with respect to $\pi(\cdot)$ because the reverse move is equally probable. If the swap is not feasible (less than two dissimilar coordinates), the current chain is simply sampled again. This is necessary to avoid complications with unequal probabilities of move types (Denison et al., 2002); the same trick is applied in reversible jump MCMC (Green, 1995). The restriction of the update to the dissimilar coordinates does not destroy detailed balance in any way; it just selects a subspace to sample on the basis of the current state. Coordinate swapping is especially powerful for combinatorial problems. Based on some preliminary studies, we use a 90/10 % mix of directed and random swaps, respectively.

4 Case studies

We now present three different case studies with increasing complexity. The first study consists of a typical integer estimation problem, and involves a Sudoku puzzle. This puzzle has become quite popular in the past 10 yr, and many newspapers, journals, and magazines around the world publish Sudokus for entertainment. This synthetic study illustrates the ability of DREAM$_{(\mathrm{D})}$ to help solve a relatively difficult and high-dimensional combinatorial optimization problem. The second study revisits a classical site characterization problem in vadose zone hydrology and involves inference of the hydraulic properties of unsaturated soils from laboratory or in situ measured water retention data. This example demonstrates how DREAM$_{(\mathrm{D})}$ can be utilized to solve experimental design problems and guide measurement collection. The third and final study resolves discrete parameters in a parsimonious lumped watershed model using observed daily discharge data from the Guadalupe River in Texas. The posterior distribution derived with DREAM$_{(\mathrm{D})}$ is compared against its counterpart derived with DREAM assuming a continuous formulation of the model calibration problem.

4.1 The daily Sudoku

The first case study considers a Sudoku puzzle, a popular and widely practised integer estimation problem. We use this example to illustrate the ability of DREAM$_{(\mathrm{D})}$ to successfully find the optimum of a discrete search problem. The objective is to fill a 9×9 grid with numbers so that each column, each row, and each of the nine 3×3 sub-grids that make up the total square contains the values of 1 to 9. The same integer may only appear once in each column and row of the 9×9 playing board including in any of the nine 3×3 subregions. Each puzzle starts with a partially completed grid, and typically has a single (unique) solution. The puzzle was popularized in 1986 by the Japanese puzzle company Nikoli, under the name Sudoku, meaning single number (Hayes, 2006). Nowadays, Sudoku puzzles are very popular, and widely practised by many millions of people throughout the world.

We consider the Sudoku puzzle in Fig. 3, taken from Wikipedia (http://en.wikipedia.org/wiki/Sudoku). The initial grid is depicted at the left-hand side, whereas the final solution is presented at the right-hand side. In this particular puzzle, the solution at 29 different cells is known, leaving us with $d=81-29=52$ parameter values to be estimated. These parameters can take on values between 1 to 9. Figure 4 illustrates the sampled values of DREAM$_{(\mathrm{D})}$ at various stages during the search. A total of $N=20$ different chains were used to search the parameter space, and the initial sample was created using random permutation of the available numbers. This ensures that each value of 1 to 9 appears only nine times in the grid. The log-likelihood function measures the sum of the horizontal, vertical and subgrid constraint violation.

The first grid, on the left-hand side of Fig. 4 depicts the starting solution of one of the different chains. This initial state is randomly drawn from the prior distribution, and appears rather poor. For example, the subregion at the bottom right hand corner contains three numbers eight, a severe

(A)

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

(B)

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Fig. 3. The daily sudoku: **(A)** initial solution, and **(B)** final solution. The black numbers were given, whereas the solution numbers are marked in red.

(A)

5	3	1	7	7	2	7	1	4
6	4	2	1	9	5	4	8	3
1	9	8	6	4	3	9	6	2
8	5	9	2	6	7	1	5	3
4	7	6	8	5	3	2	9	1
7	3	5	9	2	1	8	4	6
9	6	7	5	2	4	2	8	8
3	2	7	4	1	9	6	8	5
5	1	4	3	8	6	3	7	9

(B)

5	3	4	6	7	2	9	1	7
6	7	2	1	9	5	1	5	8
1	9	8	3	4	8	4	6	2
8	1	9	7	6	4	5	2	3
4	2	6	8	5	3	7	9	1
7	5	3	9	2	1	8	4	6
9	6	7	5	3	7	2	8	4
2	3	8	4	1	9	6	3	5
1	4	5	2	8	6	3	7	9

(C)

5	3	2	6	7	8	9	1	4
6	7	4	1	9	5	3	2	8
1	9	8	3	4	2	5	6	7
8	1	9	7	6	4	2	5	3
4	2	6	8	5	3	7	9	1
7	5	3	9	2	1	8	4	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

(D)

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	6	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Fig. 4. Sudoku puzzle: evolution of the DREAM$_{(D)}$ sampled parameter space to the posterior distribution: **(A)** typical starting solution, **(B)** intermediate solution, **(C)** nearly final solution, and **(D)** final solution.

violation of the constraints. Figure 4b displays the state of the same chain after about 25 000 function evaluations. The solution has improved considerably, but still contains several noticeable deviations from the actual solution. The third panel in Fig. 4c, derived after about 50 000 Sudoku evaluations is a further refinement of the solution presented in Fig. 4b. A few important swaps have been introduced that further reduce the constraint violation. Finally, after about 100 000 samples, Fig. 4d illustrates that the puzzle has been successfully solved.

The results of this study inspire confidence in the ability of DREAM$_{(D)}$ to successfully locate the optimum solution of a discrete search problem. This constitutes a necessary benchmark to inspire confidence in the search capabilities and convergence properties of the algorithm. A branch and bound optimization approach is about 2 orders of magnitude more efficient than DREAM$_{(D)}$ in solving the Sudoku puzzle, but this method violates detailed balance, and hence cannot be used to derive the posterior target distribution. This is the main purpose of DREAM$_{(D)}$ and this ability will be highlighted in the remaining two case studies of this paper.

If our sole interest is in finding the optimum solution of a discrete search problem, then significant efficiency gains can be achieved with DREAM$_{(D)}$ by relaxing the assumption of detailed balance of the sampled Markov chains. For instance, if we modify the directed swap in DREAM$_{(D)}$ so that each candidate point is accepted/rejected based on its own Metropolis ratio independently of the associated move of the corresponding chain, then far fewer function evaluations would be needed to solve the Sudoku puzzle. The consequence of such non-Markovian adaptation, however is that the algorithm no longer adequately samples the underlying target distribution.

4.2 Optimal experimental design for soil hydraulic characterization

The second case study considers a common problem in vadose zone hydrology and involves characterization of the hydraulic properties of variably saturated soils. This serves to demonstrate the ability of DREAM$_{(D)}$ to guide experimental data collection and help determine which soil water retention measurements to collect. We assume the capillary pressure-saturation relationship of (van Genuchten, 1980):

$$\theta(h) = \theta_r + (\theta_s - \theta_r)\left[1 + (\alpha h)^n\right]^{-m} \quad (8)$$

where θ (cm^{3}cm^{-3}) is the volumetric water content, h (cm) denotes the soil water pressure head, θ_s (θ_r) (cm^{3}cm^{-3}) signifies the saturated (residual) water content, α (cm^{-1}) and n (-) are coefficients that determine the shape of the water retention function, and $m = 1 - 1/n$. A synthetic data set of water retention observations was created for a sandy soil by evaluating Eq. (8) for a given set of pressure head values. The soil water pressure head was discretized into $M = 501$ equidistant points between $h = 0$ (saturation) and $h = -1000$ (cm), and the corresponding $\theta(h)$ curve is plotted in Fig. 5 with a solid black line. Then, the soil water pressure head observations are corrupted with a normally distributed error, $h \leftarrow N(h, 2)$ to represent the combined effect of measurement error and soil inhomogeneity. The final data set is depicted with the circles in Fig. 5.

From this data set of $M = 501$ water retention measurements we now use DREAM$_{(D)}$ to find those four $(h, \theta(h))$ observations that are most informative and best constrain the hydraulic properties, $\mathbf{x} = [\theta_s, \theta_r, \alpha, n]$ of the sandy soil under consideration. For each selected combination of four different $(h, \theta(h))$ measurements in DREAM$_{(D)}$ we estimate the corresponding values of $\mathbf{x}$ by nonlinear minimization using the Nelder-Mead Simplex algorithm. The hydraulic parameters obtained this way are then used to predict the water retention curve for the entire range of pressure head values considered in Fig. 5. A standard squared-deviation likelihood function is subsequently used to calculate $\pi(\mathbf{x})$ in step (d) of the pseudo code of DREAM$_{(D)}$ and summarize the information content for each selected set of four $(h, \theta(h))$ observations. We use the standard settings of DREAM$_{(D)}$ and run

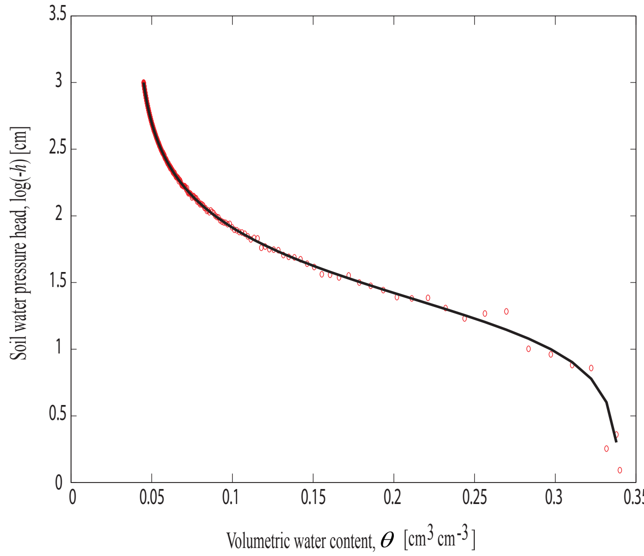

Fig. 5. Synthetic soil water retention curve of the sandy soil, and the effect of data error.

$N = 10$ different Markov chains in parallel to search the discrete measurement space. About 10 000 function evaluations were required to reach convergence to a limiting distribution.

Figure 6 presents the results of our analysis and plots histograms of the DREAM$_{(D)}$ derived posterior measurement samples after sorting each combination in increasing order. The marginal posterior distributions appear clustered around distinctly different soil water pressure head values. This includes soil water pressure head measurements around (Fig. 6a) $h = 0$, (Fig. 6b) $h = -40$, (Fig. 6c) $h = -220$, and (Fig. 6d) $h = -1000$ cm respectively. Indeed, the most informative $(h, \theta(h))$ observations are found close to saturation, around the air-entry value of the sandy soil, at the inflection point of the water retention function, and in the very dry moisture range. The location of these most informative $\theta(h)$ measurements matches very well with our previous findings (Vrugt et al., 2002) and are in agreement with the dynamic behavior of the marginal sensitivity coefficients, $\partial\theta/\partial\theta_s$, $\partial\theta/\partial\alpha$, $\partial\theta/\partial n$, and $\partial\theta/\partial\theta_r$ derived from the partial derivatives of Eq. (8). A similar pattern is observed for loamy and clayey soils (not shown herein), but the second, third and last observation depicted in Fig. 4b–d move towards lower soil water pressure head values, consistent with the increased water holding capacity of these more fine textured soils. An accurate soil hydraulic characterization thus requires water retention measurements that span the entire range of soil moisture values.

It is interesting to observe the differences in the posterior spread of the four different histograms. Whereas the first two pressure head observations in Fig. 4b–d are very well defined with relatively little dispersion, the other two measurements are considerably less well defined symptomatic

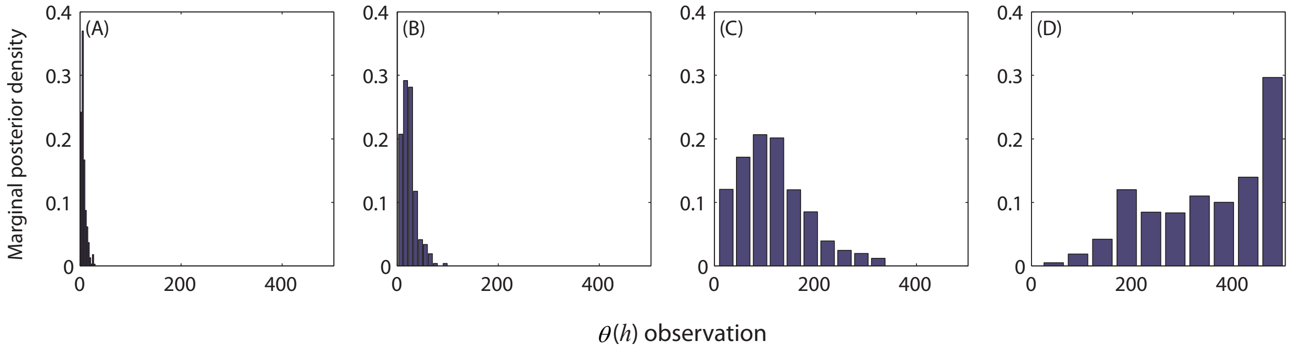

Fig. 6. Histograms of the posterior samples generated with DREAM$_{(\mathrm{D})}$. The most informative soil water pressure head measurements are somewhat clustered at different moisture values that range from saturation (left plot) to the dry end (right plot). The **x**-axis indexes the observations $[1,\ldots,501]$.

for a redundancy in information content. This is an interesting finding, and explains why the parameters n and θ_r are often found to be highly correlated in the fitting of water retention curves. The posterior spread derived with DREAM$_{(\mathrm{D})}$ is hence a useful byproduct to analyze measurement uniqueness, and determine the information content of experimental data.

4.3 Watershed model calibration using discrete parameter estimation

The third and final case study involves flood forecasting, and consists of the calibration of a mildly complex lumped watershed model using historical data from the Guadalupe River at Spring Branch, Texas. This is the driest of the 12 MOPEX basins described in the study of Duan et al. (2006). The model structure and hydrologic process representations are found in Schoups and Vrugt (2010). The model transforms rainfall into runoff at the watershed outlet using explicit process descriptions of interception, throughfall, evaporation, runoff generation, percolation, and surface and subsurface routing. Table 1 summarizes the seven different model parameters, and their prior uncertainty ranges. Each parameter is discretized equidistantly in 250 intervals with respective step size listed in the last column at the right hand side. This gridding is necessary to create a non-continuous, discrete, parameter estimation problem. Unlike the previous case study in which integer values are sampled only, this particular study (mostly) involves non-integer values.

Daily discharge, mean areal precipitation, and mean areal potential evapotranspiration were derived from Duan et al. (2006) and used for model calibration. Details about the basin, experimental data, and likelihood function can be found there, and will not be discussed herein. The same model and data was used in a previous study (Schoups and Vrugt, 2010), and used to introduce a generalized likelihood function for heteroscedastic, non-Gaussian, and correlated (streamflow) prediction errors.

Figure 7 presents histograms of the marginal distribution of a few selected hydrologic model parameters using five years of observed daily discharge data. The top panel presents the results of DREAM$_{(\mathrm{D})}$, whereas the bottom panel presents the results for a continuous parameter space. These histograms were derived by separately running DREAM for the same data set and model. Notice the close agreement between the histograms derived with both MCMC methods. This is a testament to the ability of DREAM$_{(\mathrm{D})}$ to successfully solve discrete posterior parameter estimation problems. The influence of gridding is hardly noticeable, but becomes apparent if we use at least 25 bins to represent the marginal density (not shown herein).

To better illustrate the effect of discretization, please consider Fig. 8 that presents two-dimensional scatter plots of the DREAM$_{(\mathrm{D})}$ derived posterior samples for a few selected parameter pairs. The bottom panel shows similar plots but then assuming continuity of the parameter space. The effect of gridding is immediately apparent. Whereas the original bivariate scatter plots sample the parameter space in a (bloodstain) spatter pattern, two-dimensional plots of the posterior samples derived with DREAM$_{(\mathrm{D})}$ exhibit an obvious grid pattern with horizontally and vertically aligned points. The posterior samples take on discrete values with a distance between subsequent points that is similar to the intervals listed in Table 1. Despite this difference in sampling pattern, the shape of the DREAM and DREAM$_{(\mathrm{D})}$ derived bivariate scatter plots are very similar, commensurate with the covariance structure of the posterior distribution. The results presented in Fig. 7 inspire confidence in the ability of DREAM$_{(\mathrm{D})}$ to solve noncontinuous posterior sampling problems.

The excellent correspondence of the posterior parameter distributions derived with DREAM and DREAM$_{(\mathrm{D})}$ results in marginal differences of the resulting streamflow predictions. We therefore do not show any times series plots of model predictions, and corresponding 95 % uncertainty ranges. Such plots can be found in Schoups and Vrugt (2010)

Table 1. Prior Uncertainty Ranges of Hydrologic and Error Model Parameters.

Parameter	Symbol	Minimum	Maximum	Units	Step size
Maximum interception	I_{max}	0	10	mm	0.02
Soil water storage capacity	S_{max}	10	1000	mm	1.98
Maximum percolation rate	Q_{max}	0	100	mm d^{-1}	0.20
Evaporation parameter	α_E	0	100	–	0.20
Runoff parameter	α_F	-10	10	–	0.04
Time constant, fast reservoir	K_F	0	10	days	0.02
Time constant, slow reservoir	K_S	0	150	days	0.30
Heteroscedasticity intercept	σ_0	0	1	mm d^{-1}	0.002
Heteroscedasticity slope	σ_1	0	1	–	0.002
Autocorrelation coefficient	ϕ_1	0	1	–	0.002
Kurtosis parameter	β	-1	1	–	0.004

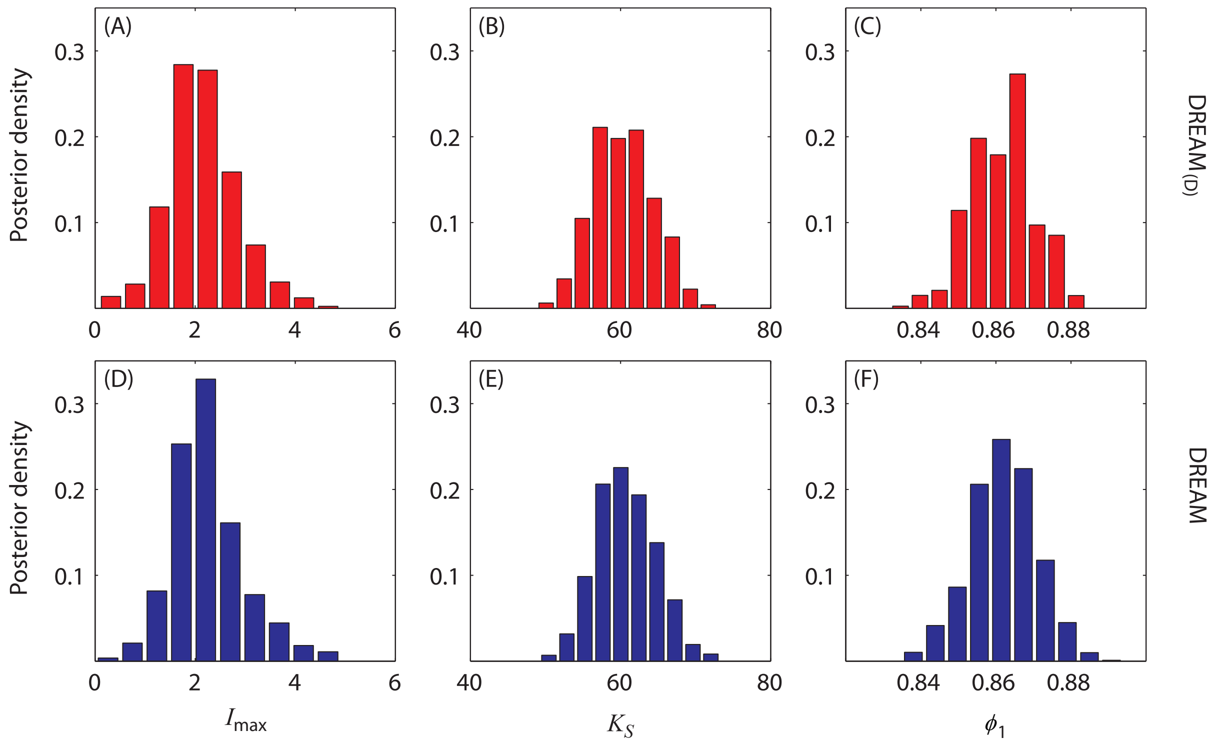

Fig. 7. Histogram of the DREAM$_{(D)}$ derived marginal posterior distributions of the **(A)** I_{max}, **(B)** K_S, and **(C)** ϕ_1 rainfall – runoff and error model parameters (in red). For convenience, only a few parameters are plotted. To benchmark the results of DREAM$_{(D)}$ the bottom panel illustrates the results for DREAM (in blue), with the common assumption of a continuous parameter space.

and details can be found in that publication. It is interesting to observe that the maximum log-likelihood value of 543 found with DREAM$_{(D)}$ is somewhat larger than its counterpart estimated with DREAM (540). This difference was consistently observed for multiple different trials with both MCMC algorithms.

To provide better insights into the efficiency of DREAM$_{(D)}$, Fig. 9a plots the evolution of the $\hat{R}$-statistic of Gelman and Rubin (1992) for each of the different parameters. To benchmark these results, the bottom panel (Fig. 9b) illustrates the convergence results of DREAM assuming a continuous parameter space. The presented trace plots represent an average of 25 different MCMC trials. The

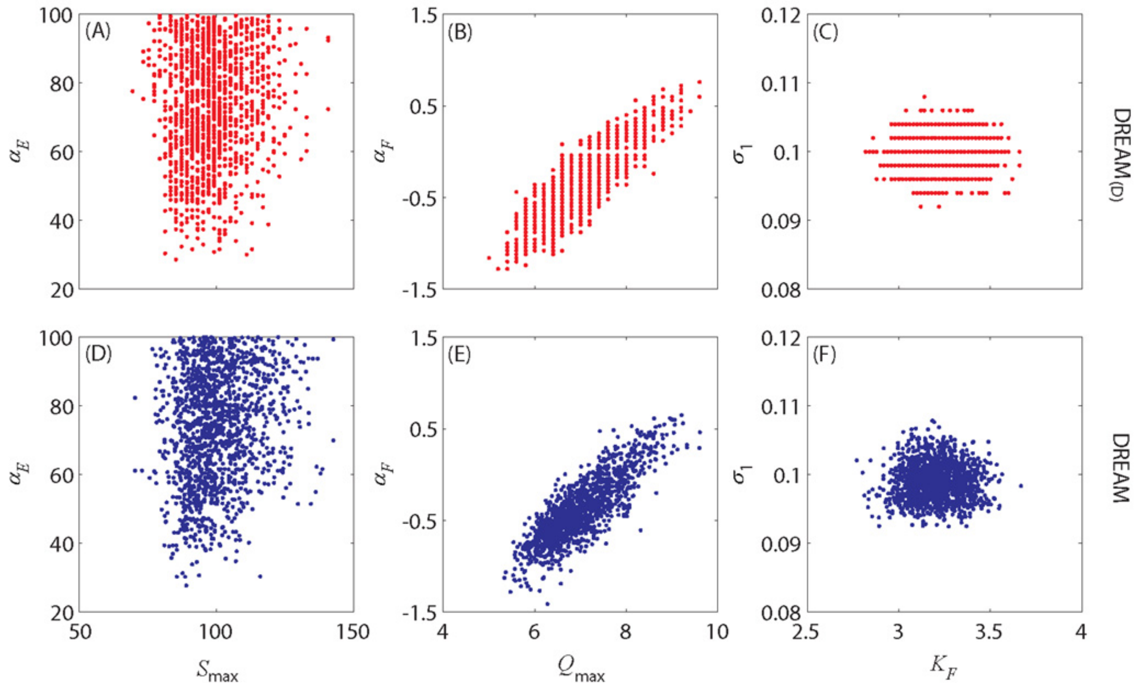

Fig. 8. Two-dimensional scatter plots of the DREAM$_{(D)}$ derived posterior samples (top panel: red), and corresponding bivariate samples estimated with DREAM (bottom panel: blue). Only a few parameter pairs are shown as this is sufficient to illustrate the similarity of the sampled distributions.

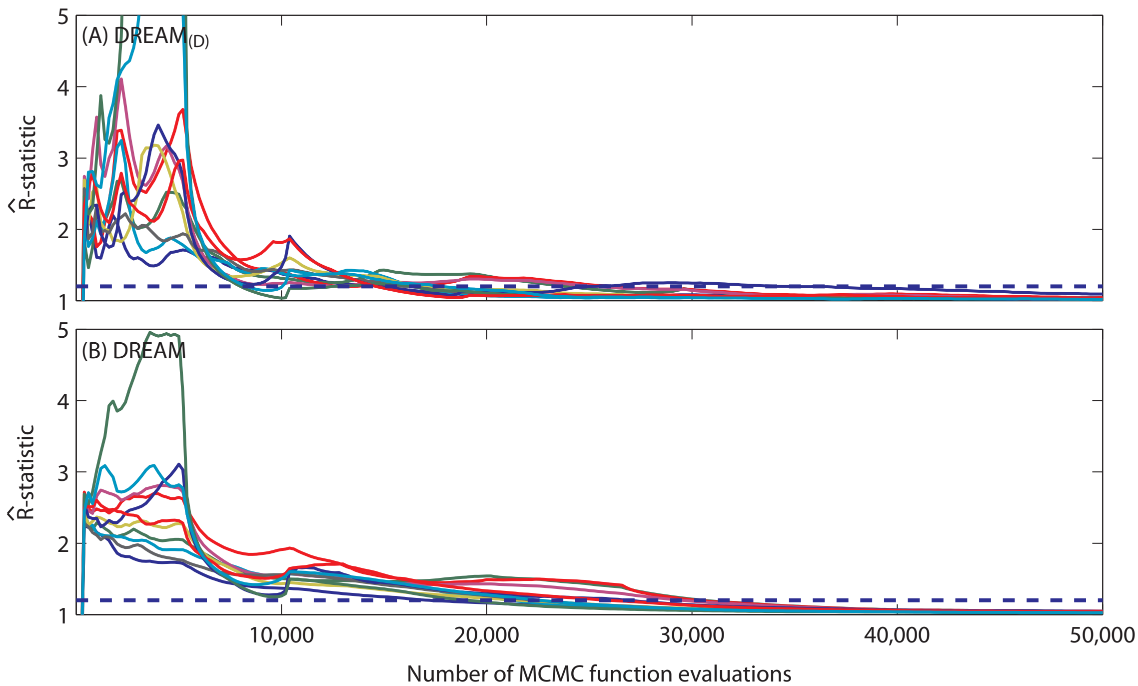

Fig. 9. Simulated traces of the $\hat{R}$-statistic of Gelman and Rubin (Gelman and Rubin, 1992) using the **(A)** DREAM$_{(D)}$ (top panel)(discrete sampling), and **(B)** DREAM (bottom panel)(continuous sampling) MCMC algorithms. Each parameter is coded with a different color. A similar convergence speed is observed for both algorithms.

convergence speed for both algorithms is strikingly similar. Both MCMC methods require about 30 000 model evaluations to converge to a limiting distribution. Although gridding significantly reduces the size of the feasible parameter space, an approximately similar number of function evaluations remains necessary to explore the posterior target distribution. Our experience with models involving a much higher parameter dimensionality demonstrate considerable enhancements in efficiency (sometimes dramatically) when sampling in the discretized rather than continuous space. Discretization might therefore provide a practical solution to speed up search efficiency for insensitive parameters. Further research on this topic is warranted.

5 Conclusions

In the past decade much progress has been made in the development of sampling algorithms for statistical inference of the posterior parameter distribution. The typical assumption in this work is that the parameters are continuous and can take on any value within their upper and lower bounds. Unfortunately, such algorithms typically do not work for discrete parameter estimation problems. Such problems are abundant in many fields of study, and therefore of considerable theoretical and practical interest. Examples include selecting among different measurement locations in the design of optimal experimental strategies, finding the best members of an ensemble of predictors, and more generally discrete model calibration problems. Here, we have introduced a discrete MCMC simulation algorithm that is especially designed to solve non-continuous and combinatorial posterior sampling problems. This method, entitled DREAM$_{(\mathrm{D})}$ uses DREAM as its main building block, yet uses a modified proposal distribution to facilitate solve discrete sampling problems. The DREAM$_{(\mathrm{D})}$ algorithm maintains detailed balance and ergodicity, and receives good performance across a range of problems involving a Sudoku puzzle, soil water retention function and discrete rainfall - runoff model calibration problem.

The theory developed herein is easily implementable in DREAM$_{(\mathrm{ZS})}$ (Vrugt et al., 2011b) and MT-DREAM$_{(\mathrm{ZS})}$ (Laloy and Vrugt, 2011), which provides a venue to further increase the efficiency of MCMC simulation. The DREAM$_{(\mathrm{D})}$ code is written in MATLAB and is available upon request (jasper@uci.edu).

Acknowledgements. We gratefully acknowledge the helpful comments of Christian Robert, Bettina Schaefli and one anonymous reviewer. Computer support, provided by the SARA center for parallel computing at the University of Amsterdam, The Netherlands, is highly appreciated.

Edited by: E. Morin

References

Box, G. E. P. and Tiao, G. C.: Bayesian Inference in Statistical Analyses Addison-Wesley-Longman, Reading, Massachusetts, 1973.

Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M.: Bayesian methods for nonlinear classification and regression, John Wiley & Sons, Chicester, 2002.

Duan, Q., Gupta, V. K., and Sorooshian, S.: Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resour. Res., 28, 1015–1031, 1992.

Duan, Q., Schaake, J., Andréassian, V., Franks, S., Goteti, G., Gupta, H. V., Gusev, Y. M., Habets, F., Hall, A., Hay, L., Hogue, T., Huang, M., Leavesley, G., Liang, X., Nasonova, O. N., Noilhan, J., Oudin, L., Sorooshian, S., Wagener, T., and Wood, E. F.: Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops, J. Hydrol., 320, 3–17, 2006.

Furman, A., Ferré, T. P. A., and Warrick, A. W.: Optimization of ERT surveys for monitoring transient hydrological events using perturbation sensitivity and genetic algorithms, Vadose Zone J., 3, 1230–1239, 2004.

Gelfand, A. E. and Smith, A. F.: Sampling based approaches to calculating marginal densities, J. Am. Stat. Assoc., 85, 398–409, 1990.

Gelman, A. G. and Rubin, D. B.: Inference from iterative simulation using multiple sequences, Stat. Sci., 7, 457–472, 1992.

Gilks, W. R. and Roberts, G. O.: Strategies for improving MCMC, in: Markov Chain Monte Carlo in Practice, edited by: Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., Chapman & Hall, London, 89–114, 1996.

Gilks, W. R., Roberts, G. O., and George, E. I.: Adaptive direction sampling, Statistician, 43, 179–189, 1994.

Green, P. J.: Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination, Biometrika, 82, 711–732, 1995.

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

Haario, H., Saksman, E., and Tamminen, J.: Adaptive proposal distribution for random walk Metropolis algorithm, Computation. Stat., 14, 375–395, 1999.

Haario, H., Saksman, E., and Tamminen, J.: An adaptive Metropolis algorithm, Bernoulli, 7, 223–242, 2001.

Haario, H., Saksman, E., and Tamminen, J.: Componentwise adaptation for high dimensional MCMC, Computation. Stat., 20, 265–274, 2005.

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

Harmancioglu, N. B., Icaga, Y., and Gul, A.: The use of an optimization method in assessment of water quality sampling sites, European Water, 5/6, 25–35, 2004.

Hayes, B.: Unwed numbers, American Scientist, 94, 12–15, doi:10.1511/2006.1.12, 2006.

Kavetski, D., Kuczera, G., and Franks, S. W.: Bayesian analysis of input uncertainty in hydrologic modeling: 2. Application, Water Resour. Res., 42, W03408, doi:10.1029/2005WR004376, 2006.

Kleidorfer, M., Möderl, M., Fach, S., and Rauch, W.: Optimization of measurement campaigns for calibration of a hydrological model, Proceedings of the 11th International Conference on Urban Drainage, Edinburgh, Scotland, UK, 2008.

Krink, T., Mittnik, S., and Paterlini, S.: Differential Evolution and Combinatorial Search for Constrained Index Tracking, Center for Studies in Banking and Finance, 09032, Universita di Modena e Reggio Emilia, Facolt̀ di Economia “Marco Biagi”, 2009.

Kuczera, G., Kavetski, D., Franks, F. W., and Thyer, M. T.: Towards a Bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters, J. Hydrol., 331, 161–177, 2006.

Laloy, E. and Vrugt, J. A.: High-dimensional posterior exploration of hydrologic models using multi-try DREAM$_{(ZS)}$ and high performance computing, Water Resour. Res., in review, doi:2011WR010608RR, 2011.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E.: Equation of state calculations by fast computing machines, J. Chem. Phys., 21, 1087–1092, 1953.

Neuman, S. P., Xue, L., Ye, M., and Lu, D.: Bayesian analysis of data-worth considering model and parameter uncertainties, Adv. Water Resour., doi:10.1016/j.advwatres.2011.02.007, in press, 2011.

Perrin, C., Andréassian, V., Rojas Serna, C., Mathevet, T., and Le Moine, N.: Discrete parameterization of hydrological models: Evaluating the use of parameter sets libraries over 900 catchments, Water Resour. Res., 44, W08447, doi:10.1029/2007WR006579, 2008.

Price, K. V., Storn, R. M., and Lampinen, J. A.: Differential Evolution, A Practical Approach to Global Optimization, Springer, Berlin, 2005.

Roberts, G. O. and Gilks, W. R.: Convergence of adaptive direction sampling, J. Multivariate Anal., 49, 287–298, 1994.

Roberts, G. O. and Rosenthal, J. S.: Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms, J. Appl. Probab., 44, 458–475, 2007.

Roberts, G. O. and Rosenthal, J. S.: Examples of adaptive MCMC, online manucript, available at: http://probability.ca/jeff/ftpdir/adaptex.pdf, 2008.

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

Schaefli, B., Talamba, D. B., and Musy, A.: Quantifying hydrological modeling errors through a mixture of normal distributions, J. Hydrol., 332, 303–315, 2007.

Schoups, G. and Vrugt, J. A.: A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic and non-gaussian errors, Water Resour. Res., 46, W10531, doi:10.1029/2009WR008933, 2010.

Smith, T. J. and Marshall, L. A.: Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques, Water Resour. Res., 44, W00B05, doi:10.1029/2007WR006705, 2008.

Sorooshian, S. and Dracup, J. A.: Stochastic parameter estimation procedures for hydrologic rainfall-runoff models: correlated and heteroscedastic error cases, Water Resour. Res., 16, 430–442, doi:10.1029/WR016i002p00430, 1980.

Storn, R. and Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Global Optimization, 11, pp. 341–359, 1997.

Ter Braak, C. J. F.: A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces, Stat. Comput., 16, 239–249, 2006.

Ter Braak, C. J. F. and Vrugt, J. A.: Differential evolution Markov chain with snooker updater and fewer chains, Stat. Comput., 16, 239–249, doi:10.1007/s11222-008-9104-9, 2008.

van Genuchten, M. Th.: A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, Soil Sci. Soc. Am. J., 44, 892–898, 1980.

Vrugt, J. A., Bouten, W., Gupta, H. V., and Sorooshian, S.: Toward improved identifiability of hydrologic model parameters: the information content of experimental data, Water Resour. Res., 38, 1312, doi:10.1029/2001WR001118, 2002.

Vrugt, J. A., Gupta, H. V., Bouten, W., and Sorooshian, S.: A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, Water Resour. Res., 39, 1–14, doi:10.1029/2002WR001642, 2003.

Vrugt, J. A., ter Braak, C. J. F., Clark, M. P., Hyman, J. M., and Robinson, B. A.: Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation, Water Resour. Res., 44, W00B09, doi:10.1029/2007WR006720, 2008.

Vrugt, J. A., ter Braak, C. J. F., Diks, C. G. H., Higdon, D., Robinson, B. A., and Hyman, J. M.: Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling, I. J. Nonlin. Sci. Num., 10, 271–288, 2009a.

Vrugt, J. A., ter Braak, C. J. F., Gupta, H. V., and Robinson, B. A.: Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling?, Stoch. Env. Res. Risk A., 23, 1011–1026, doi:10.1007/s00477-008-0274-y, 2009b.

Vrugt, J. A., Laloy, E., and ter Braak, C. J. F.: Differential evolution adaptive Metropolis with sampling from past states, SIAM J. Optimiz., in preparation, 2011.