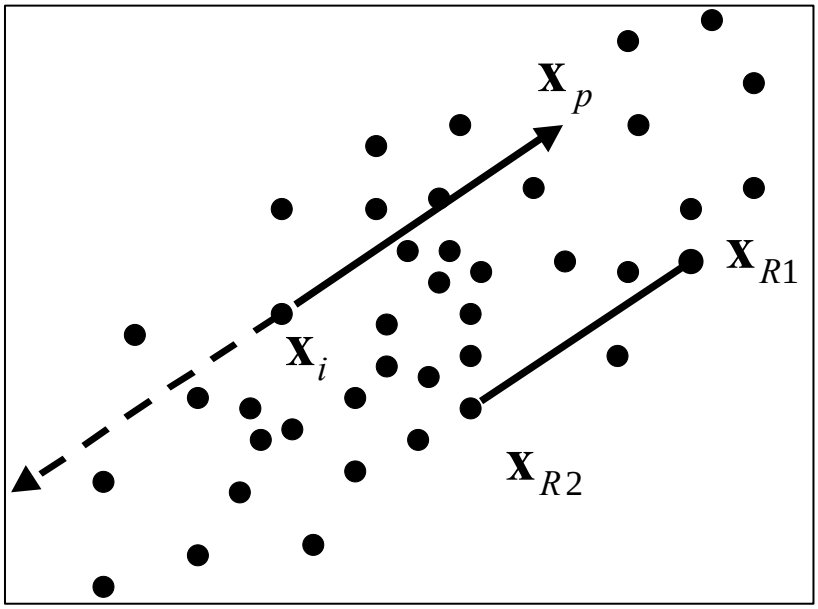
WAGENINGEN UR
For quality of life

Genetic algorithms and Markov Chain Monte Carlo: Differential Evolution Markov Chain makes Bayesian computing easy

C.J.F. ter Braak



August 2005

Biometris
quantitative methods in life and earth sciences

Biometris is the integration of the Centre for Biometry of Plant Research International and the Department of Mathematical and Statistical Methods of Wageningen University. Biometris, part of Wageningen University and Research center (Wageningen UR), was established on 20th June 2001.

For more information please visit the website http://www.biometris.nl or contact:

Biometris, Wageningen UR
P.O. Box 100
6700 AC Wageningen, The Netherlands
Phone: +31 (0)317 484085; Fax: +31 (0)317 483554; E-mail: biometris@wur.nl

Genetic algorithms and Markov Chain Monte Carlo: Differential Evolution Markov Chain makes Bayesian computing easy.

Revised version of Biometris report 010404

Cajo J. F. Ter Braak
Biometris, Wageningen University and Research Centre, Wageningen, The Netherlands

Address for correspondence: Cajo J. F. ter Braak, Biometris, Wageningen University and Research Centre, Box 100, 6700 AC Wageningen, The Netherlands.
E-mail: Cajo.terbraak@wur.nl

Differential Evolution (DE) is a simple genetic algorithm for numerical optimization in real parameter spaces. In a statistical context one would not just want the optimum but also its uncertainty. The uncertainty distribution can be obtained by a Bayesian analysis (after specifying prior and likelihood) using Markov Chain Monte Carlo (MCMC) simulation. This paper integrates the essential ideas of DE and MCMC, resulting in Differential Evolution Markov Chain (DE-MC). DE-MC is a population MCMC algorithm, in which multiple chains are run in parallel. DE-MC solves an important problem in MCMC, namely that of choosing an appropriate scale and orientation for the jumping distribution. In DE-MC the jumps are simply a fixed multiple of the differences of two random parameter vectors that are currently in the population. The selection process of DE-MC works via the usual Metropolis ratio which defines the probability with which a proposal is accepted. In tests with known uncertainty distributions, the efficiency of DE-MC with respect to random walk Metropolis with optimal multivariate Normal jumps ranged from 68% for small population sizes to 100% for large population sizes and even to 500% for the 97.5% point of a variable from a 50-dimensional Student distribution. Two Bayesian examples illustrate the potential of DE-MC in practice. DE-MC is shown to facilitate multidimensional updates in a multi-chain "Metropolis-within-Gibbs" sampling approach. The advantage of DE-MC over conventional MCMC are simplicity, speed of calculation and convergence, even for nearly collinear parameters and multimodal densities.

KEY WORDS: Block updating; Evolutionary Monte Carlo; Metropolis algorithm; Population Markov Chain Monte Carlo; Simulated Annealing; Simulated Tempering; Theophylline Kinetics

1 Introduction

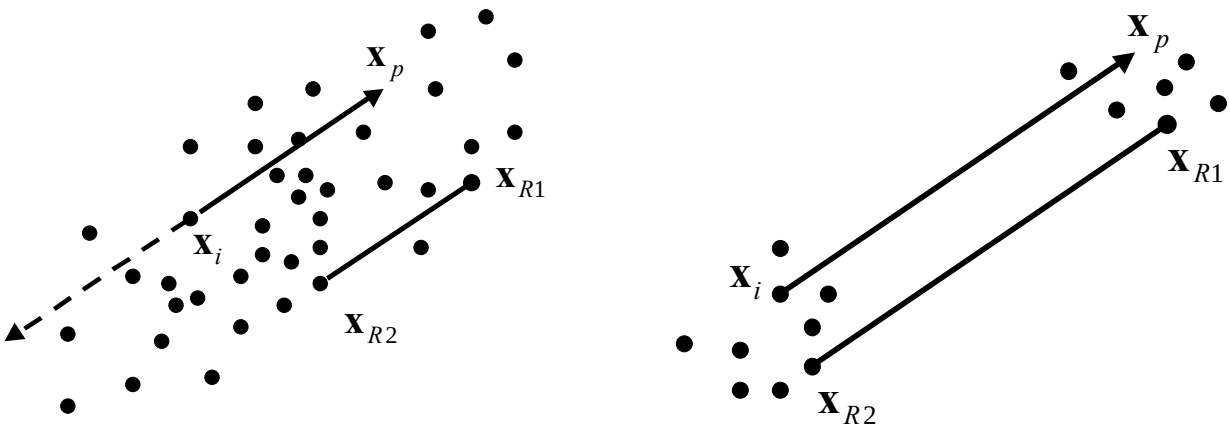
This paper combines the genetic algorithm called Differential Evolution (DE) (Price and Storn 1997, Storn and Price 1997, Price 1999) for global optimization over real

parameter space with Markov Chain Monte Carlo (MCMC)(Gilks *et al.* 1996) so as to generate a sample from a target distribution. In Bayesian analysis the target distribution is typically a high dimensional posterior distribution. Both DE and MCMC are enormously popular in a variety of scientific fields for their power and general applicability. Lampinen (2001) provides a bibliography of DE and Gelman *et al.* (2004) and Robert and Casella (2004) provide introductions to MCMC. In our combination we run multiple Markov chains, which are initialized from overdispersed states, in parallel and let the chains learn from each other - instead of running the chains independently as a way to check convergence (Gelman *et al.* 2004) and as carried out in WinBUGS (Lunn *et al.* 2000). The idea of combining genetic or evolutionary algorithms with MCMC is explored, among others, by Liang and Wong (2001), Liang (2002) and Laskey and Myers (2003) and is closely related to work in the 1990's on parallel tempering and adaptive direction sampling (Gilks and Roberts 1996). The combination of DE and MCMC solves an important problem in MCMC in real parameter spaces, namely that of choosing an appropriate scale and orientation for the jumping distribution. Note that adaptive direction sampling solves the orientation problem but not the scale problem.

A commonly used jumping distribution for MCMC in a d-dimensional real parameter space is the multivariate normal distribution (Gelman *et al.* 2004). The problem then lies in specifying the covariance matrix of this distribution. The d variances and the $d(d\text{-}1)/2$ covariances need to be chosen in such a way so as to balance progress in each step and a reasonable acceptance rate (the square-root of the variance relates to the relevant scale of each parameter and the correlations relate to the orientation). Traditionally, all these are estimated from a trial run and much recent research is devoted to ways of doing that efficiently and/or adaptively (Haario *et al.* 2001). If parameters are highly correlated, special precautions must be taken to avoid singularity of the estimated covariances matrix. In this paper, N chains are run in parallel and the jumps for a current chain are derived from the remaining N-1 chains. The simplest strategy, which balances exploration and exploitation of the space, takes the difference of vectors of two randomly chosen chains, multiplies the difference with a factor γ and adds the result to the vector of the current chain (Figure 1). The difference vector contains the required information on scale and orientation. Each

Fig. 1. Differential Evolution in two dimensions with 40 (a) and 15 (b) members in the population (d = 2, N = 40 and 15). The proposal vector $\mathbf{x}_p$ to update the ith member is generated from $\mathbf{x}_i$ and the randomly drawn members $\mathbf{x}_{R1}$ and $\mathbf{x}_{R2}$ by (2) with $\gamma = 2.4/(2\times2)^{1/2} = 1.2$ in (a) and $\gamma = 1.0$ in (b) and $\mathbf{e}$ = (0,0) in both. The dashed arrow in (a) points to the proposal when $\mathbf{x}_{R1}$ would have been drawn after $\mathbf{x}_{R2}$. The reverse jump from $\mathbf{x}_p$ to $\mathbf{x}_i$ is obtained by translating the dashed arrow to $\mathbf{x}_p$.

Fig. 1 (a) (b)



proposal is shown to define a Metropolis step, in which each jump is equally likely as the reverse jump, given the current state of the remaining chains. The N-chain is therefore a single random walk Markov chain on an $N \times d$-dimensional space. The new method is called Differential Evolution Markov Chain (DE-MC). The core of the method can be coded in about 10 lines, requiring only a function to draw uniform random numbers and a function to calculate the fitness of each proposal vector (Figure 2). We provide some theory and intuition for why DE-MC works, which also suggests good values for N and γ, the only free parameters of the proposal scheme. We demonstrate how the method can be used for block updating in a multi-chain Gibbs sampler and provide DE-variants of simulated annealing and simulated tempering. The effectiveness of the method is demonstrated on three known distributions (Normal, Student and Normal mixtures) and on two Bayesian data analysis examples.

Fig. 2. C-style pseudocode for Differential Evolution Markov Chain and simulated tempering and annealing variants. Notation: X = $N \times d$ matrix with elements X[i][j] and X[i]= $\mathbf{x}_i$, the ith member chain of the population; x_p = proposal d-vector $\mathbf{x}_p$, and fitness(.)= π(.), c = γ. Uniform(a,b) is a function for drawing uniform random numbers between a and b. Record(X) is a function to collect the draws. The function CoolingSchedule() = 1 for DE-MC but unequal to 1 for simulated tempering and annealing versions of DE-MC.

```
for ( s=0; s<N_generation; s++ ){ /* cycle through generations */
        Temperature = CoolingSchedule(s, N_generation)
        for (i=0; i<N ; i++) { /* cycle through members of population */
        /* randomly select 2 different numbers R1 and R2 unequal to i */
                do {R1 = floor(Uniform(0,1)*N);} while(R1 ==i);
                do {R2 = floor(Uniform(0,1)*N);} while(R2 ==i||R2==R1);
        /* proposal: DE1 strategy in Storn & Price 1995 TR-95-012 */
                for (j=0;j<d; j++){
                        x_p[j]= X[i][j]+c*(X[R1][j]-X[R2][j])+Uniform(-b,b);}
                r = fitness(x_p) / fitness(X[i]);
        /* selection process:  accept if Metropolis ratio r > Uniform(0,1) */
                if ( log(r) > Temperature*log(Uniform(0,1) ) swap(X[i] , x_p);
        /* X[i] is a draw from the target density (even if x_p was rejected) */
        } /*end of cycle through members of population */
        Record(X);
} /* end cycle through generations */
/* summarize the recorded sample of draws*/
```

2 Theory

2.1 Random Walk Metropolis

The random walk Metropolis algorithm (RWM) is a generic algorithm to draw a sample from a d-dimensional target distribution with probability density function (pdf) π(.). In this paper, RWM is used with a multivariate normal jumping distribution centred at the current point and with variance $\tilde{\Sigma}$. Here $\tilde{\Sigma}$ is a variance matrix which must be chosen by the user. The algorithm works as follows. It repeatedly updates a

single d-dimensional parameter vector $\mathbf{x}$ by (a) generating a proposal $\mathbf{x}_p = \mathbf{x} + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \tilde{\Sigma})$, (b) calculating the Metropolis ratio $r = \pi(\mathbf{x}_p)/\pi(\mathbf{x})$ and (c) accepting the proposal by setting $\mathbf{x} = \mathbf{x}_p$ with probability min(1,r) and continuing with $\mathbf{x}$ otherwise. The result is a Markov chain which, under some regularity conditions, has a unique stationary distribution with pdf $\pi(.)$. In Bayesian analyses, $\pi(.) \propto$ prior $\times$ likelihood. Roberts and Rosenthal (2001) and Gelman *et al.* (2004) summarize the guidelines for the choice $\tilde{\Sigma}$. Optimally, $\tilde{\Sigma} = c^2\boldsymbol{\Sigma}$ with $\boldsymbol{\Sigma} = \mathrm{cov}_\pi(\mathbf{x})$, the covariance of the target distribution, and c such that the fraction of acceptances is, for large d, about 0.23 (0.44 for d = 1 and 0.28 for d = 5). For a multivariate Normal target, $c = 2.38/\sqrt{d}$ is optimal.

2.2 Genetic Algorithms and Differential Evolution

In genetic algorithms (Schmitt 2004) and population MCMC (Laskey and Myers 2003) several (Markov) chains are simulated in parallel. Where the state of a single chain is given by a single d-dimensional vector $\mathbf{x}$, there are now N such vectors $\mathbf{x}_1 \ldots \mathbf{x}_N$. Here these vectors are called members of population $\mathbf{X}$, an $N \times d$ matrix, with members in rows. In a Bayesian analysis the initial population could be drawn from the prior distribution of the parameters.
Differential evolution (DE) (Price and Storn 1997) is a particularly simple genetic algorithm designed for optimization in real parameter spaces. Assuming $N > 4$, the default proposal for ith member $\mathbf{x}_i$ in DE is (Storn and Price 1997)

$$\mathbf{x}_p = \mathbf{x}_{R0} + \gamma(\mathbf{x}_{R1} - \mathbf{x}_{R2}) \qquad (1)$$

where $\mathbf{x}_{R0}$, $\mathbf{x}_{R1}$ and $\mathbf{x}_{R2}$ are randomly selected without replacement from the population $\mathbf{X}_{-i}$ (the population without $\mathbf{x}_i$). Crossover to further modify the proposal is introduced and discussed in section 5.1. The proposal vector is retained if the fitness of $\mathbf{x}_p$ is higher than the fitness of $\mathbf{x}_i$. If the fitness function is $\pi(.)$ then the proposal is thus accepted if $r = \pi(\mathbf{x}_p)/\pi(\mathbf{x}_i) > 1$. Typical values of γ are between 0.4 and 1. Proposal (1) is just one of a family of proposal schemes (Storn and Price 1997).

2.3 Differential Evolution Markov Chain

In order to turn DE into a Markov chain for drawing samples from a target distribution, the proposal and acceptance scheme must be such that there is detailed balance with respect to $\pi(.)$ (Waagepetersen and Sorensen 2001, Gelman *et al.* 2004, Robert and Casella 2004) This appears impossible with proposal scheme (1). More promise has scheme DE1, the first one considered in Storn and Price (1995) in which $\mathbf{x}_{R0}$ in (1) is replaced by $\mathbf{x}_i$ (Figure 1). To ensure that the whole parameter space can be reached, scheme DE1 is modified to

$$\mathbf{x}_p = \mathbf{x}_i + \gamma(\mathbf{x}_{R1} - \mathbf{x}_{R2}) + \mathbf{e} \qquad (2)$$

where $\mathbf{e}$ is drawn from a symmetric distribution with a small variance compared to that of the target, but with unbounded support, *e.g.* $\mathbf{e} \sim \mathrm{N}(0,b)^d$ with b small. The key of this paper is to introduce a probabilistic acceptance rule in DE: proposal (2) is

accepted with probability min(1,r) where $r = \pi(\mathbf{x}_p)/\pi(\mathbf{x}_i)$. The resulting algorithm is called Differential Evolution Markov Chain (DE-MC). The simplicity of DE-MC is best appreciated from the pseudocode in Fig. 2.

Theorem. DE-MC yields a Markov chain of which the unique stationary distribution has pdf $\pi(.)^N$.

Proof. The proof consists of two parts.
(a) $\pi(.)$ is a stationary distribution of the ith chain, because the chain is reversible. This holds true because the jumps in each member chain satisfy detailed balance with respect to $\pi(.)$ at each step. This can be proven as follows. For the ith member, the probability from the jump of $\mathbf{x}_i$ to $\mathbf{x}_p$ is equal to the reverse jump, as we can see from

$$\mathbf{x}_i = \mathbf{x}_p - \gamma(\mathbf{x}_{R1} - \mathbf{x}_{R2}) - \mathbf{e} = \mathbf{x}_p + \gamma(\mathbf{x}_{R2} - \mathbf{x}_{R1}) - \mathbf{e}$$

and noting that the pair ($\mathbf{x}_{R1}$, $\mathbf{x}_{R2}$) is equally likely as ($\mathbf{x}_{R2}$, $\mathbf{x}_{R1}$) and that the distribution of $\mathbf{e}$ is symmetric. If $\mathbf{x}_i \sim\pi(.)$, then detailed balance is achieved point-wise by accepting the proposal with probability min(1,r) where $r = \pi(\mathbf{x}_p)/\pi(\mathbf{x}_i)$. As the Jacobian of the transformation implied by (2) is 1 in absolute value[1], detailed balance also holds in terms of arbitrary measurable sets, as required for reversibility of the Markov chain (Waagepetersen and Sorensen 2001). Conditionally on the other chains, $\pi(.)$ is therefore a stationary distribution of the ith chain. As the conditional stationary distribution does not depend on the state of the other chains and is identical for all chains, $\pi(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \pi(\mathbf{x}_1)\times\ldots\times\pi(\mathbf{x}_N)$ is a joint stationary distribution.
(b) The stationary distribution is unique, if the chain is aperiodic, not transient and irreducible (Robert and Casella 2004). The first two conditions are satisfied, except for trivial exceptions, because DE-MC generates in each member a random walk. For the third condition, it is required that any state can be reached with positive probability and this is guaranteed by the unbounded support of the distribution of $\mathbf{e}$ in (2) (Robert and Casella 2004). Each component has therefore a unique stationary distribution which, from (a), is $\pi(.)$. This concludes the proof.

Because the joint stationary 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 DE-MC has become independent of its initial value. This feature of population MCMC samplers, first noticed by Mengersen and Robert (2003), is important for monitoring the convergence of a DE-MC run with the $\hat{R}$-statistic of Gelman *et al.* (2004). This statistic compares for each scalar parameter of interest the between- and within-variance of the chains. Because of the asymptotic independence, the between-member variance and $\hat{R}$ can be estimated consistently from a single DE-MC run. Gelman *et al.* (2004) consider $\hat{R}$ below 1.2 acceptable.

[1] The Jacobian is unequal to 1 in the 'type II' geometric proposals of Strens *et al*. (2002) so that the target is not a stationary distribution of their downhill Simplex sampler, as can easily be checked by simulation.

2.4 Why does DE-MC work in practice?

Let, if they exist, $\boldsymbol{\mu} = E(\mathbf{x})$ and $\boldsymbol{\Sigma} = \text{cov}(\mathbf{x})$, the expectation and covariance of the target distribution. Then, after convergence, for each population member i and j,

$E(\mathbf{x}_i) = \boldsymbol{\mu}$ and $E[(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)] = 2\boldsymbol{\Sigma}$

with expectation across generations. Also, after burn-in the averages across the population at each generation converge for large N to the expectation and covariance of the target distribution, *i.e.*

$\text{ave}(\mathbf{x}_i) \rightarrow \boldsymbol{\mu}$ and $\text{ave}[(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)] \rightarrow 2\boldsymbol{\Sigma}$ for $N \rightarrow \infty$

with ave the average across the (pairs of) population members.

For large N and small b, the proposal (2) thus looks like $\mathbf{x}_p = \mathbf{x}_i + \gamma\,\boldsymbol{\varepsilon}$ with $E(\boldsymbol{\varepsilon})=\mathbf{0}$ and $\text{cov}(\boldsymbol{\varepsilon})= 2\boldsymbol{\Sigma}$, the covariance matrix of the target. In particular, if $\pi(.)$ is multivariate normal, then $\gamma\boldsymbol{\varepsilon} \sim N(0,2\gamma^2\boldsymbol{\Sigma})$ so that DE-MC is expected to behave like RWM. From the guidelines for c in RWM (Roberts and Rosenthal 2001) the optimal choice of γ is then $2.38/\sqrt{(2d)}$. This choice of γ is expected to give an acceptance probability of 0.44 for $d = 1$, 0.28 for $d= 5$ and 0.23 for large d. If the initial population is drawn from the prior, DE-MC translates the 'prior population' to the 'posterior population'.

What happens if $N \leq d$? Because N points lie in an N-1 dimensional space, all proposals (2) will lie in this reduced space when $\mathbf{e} = \mathbf{0}$. Therefore convergence of DE-MC would rely on $\mathbf{e}$, which would take a long time if its variance is small. For speed of computation and to stress that convergence does not depend on the unbounded support of $\mathbf{e}$, we actually used $\mathbf{e} \sim \text{Uniform}[-b,b]^d$ with $b = 10^{-4}$ in all computations (Fig. 2). In the next section the effect of N on the efficiency of DE-MC is studied via simulation for $N > d$.

3 Tests with known targets

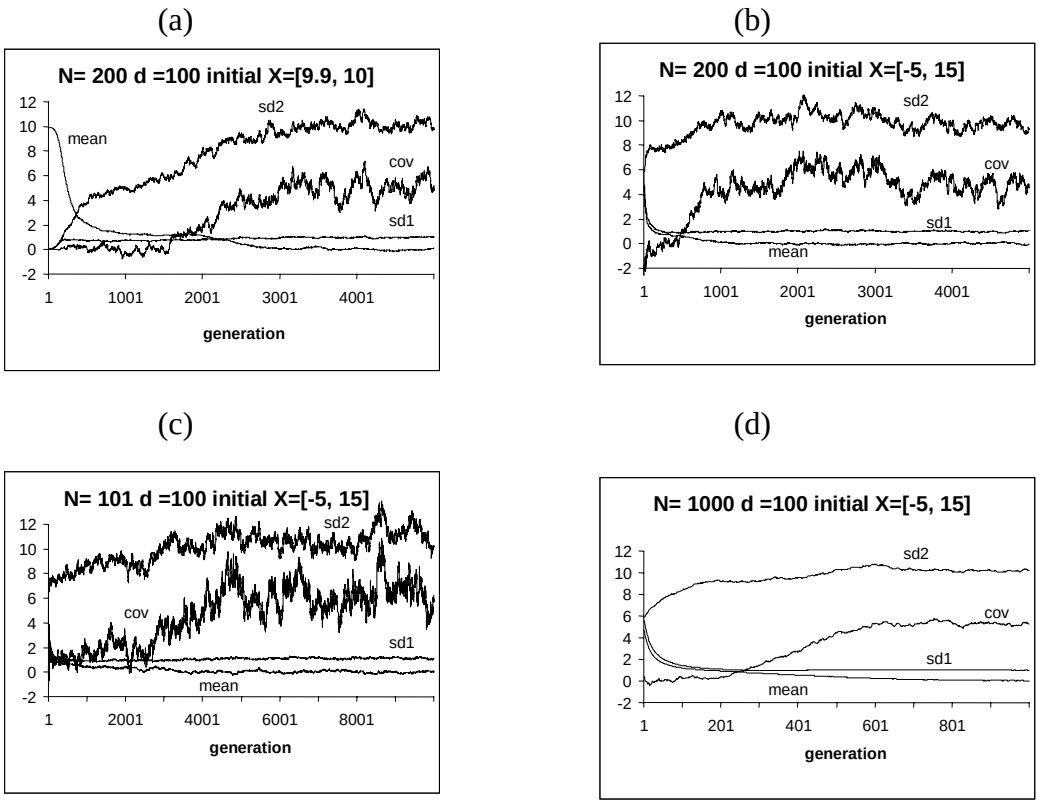
DE-MC was applied to multivariate Normal distributions and Student distributions with three degrees of freedom, both targets centred at the zero vector. The covariance matrix was set such that the variance of the jth variable was equal to j and all pairwise correlations were 0.5. These targets were chosen to reflect the possibly widely differing scales of unknown parameters in applications. Bimodal distributions, in the form of two-component Normal mixtures, were also used as targets. In all simulations and analyses the default $\gamma = 2.38/\sqrt{(2d)}$. In the sequel, draws count the number of proposal evaluations (each one requiring one evaluation of $\pi(.)$) and generations will refer to cycles through the population (Fig. 2).

3.1 Multivariate Normal Target

Fig. 3 shows how the sample means, standard deviations and correlations of the population $\mathbf{X}$ evolve in time for $d = 100$. Fig. 3a and Fig. 3b contrast narrow and broadly distributed initial populations, both with 200 members and mean ~10 for all variables. If the initial population is drawn from a narrow distribution (Fig. 3a), each standard deviation tended to increase in time to a value close to its true value in the

target, being 1 for the first variable and 10 for the hundredth variable. Simultaneously, the means and correlations evolved to values close to their true values; in Fig. 3a, the mean of the first variable evolved from 10 to close to 0, and the covariance between the first and last variable evolved from 0 to around 5 (corresponding to a correlation of 0.5). In Fig. 3b, the initial distribution is much too broad for the first variable and slightly too narrow the hundredth variable, so that the standard deviation of the first variable decreased in time, whereas that of the hundredth variable increased in time. The convergence of DE-MC to probable values was quicker in Fig. 3b than in Fig. 3a.

Fig. 3. How the mean and (co)variance of a Population of N members convergence to true values for a 100-dimensional Normal target in relation to N and initial population **X**. Shown are the mean of the first variable, the standard deviations (sd1 and sd2) and covariance (cov) of the first variable and the last variable. The true values are 0, 1, 10 and 5, respectively. (a) narrow initial population, Uniform$[9.9,10]^{100}$; (b)-(d) broad initial population, Uniform$[-5,15]^{100}$; (a)-(b) N=200; (c) N = 101; (d) N = 1000.



In all further simulations for Normal targets, the initial population was drawn from Uniform$[-5,15]^d$, reflecting prior ignorance about the mean and variance of the target. Figures 3b-d contrast different population sizes. Judged by inspection of the figures, convergence was reached after about 4000, 1000 and 600 generations for N = 101, 200 and 1000, respectively *i.e.* after 400,000, 400,000 and 600,000 functions evaluations. Judged this way, convergence is thus fastest for the smaller population sizes. The figures focus on the variance between the members in the population and not so much on the mean, and not at all on the within member variance. The $\hat{R}$-

statistic of Gelman *et al.* (2004), which compares the within- and between-member variance, drops below 1.2 for all 100 parameters after about 900 – 1000 generations for all three population sizes. Judged by $\hat{R}$-statistic, these populations thus converged about equally fast in terms of the number of generations. Smaller populations thus converge faster than larger ones in terms of the number of draws. As can be expected, the sample means, standard deviations and correlations per generation were more variable in smaller than in larger populations (Fig. 3).
We also monitored the fraction of acceptances per generation. For N = 101, the acceptance fraction varied approximately binomially around 0.20, whereas for $N \geq$ 200 the mean fraction after convergence was 0.23. For N = 200 with narrow initial population (the case of Fig. 3a) the fraction of acceptances started with values above 0.9 and then decreased to values between 0.17 and 0.30 after 2000 iterations. In the case of Fig. 3b (broad initial population) the acceptance fraction was almost immediately in the right range. For N = 1000, the trace for the acceptance fraction started off at 0.34, then dropped to a mean value of 0.18, and then slowly increased to 0.23 at iteration 750. Further experimentation with different starting distributions, e.g. Uniform[0, 5]^d, learned that the shape of this trace is particular to this broad initial population.

Table 1 shows the efficiency of DE-MC with respect to RWM with the optimal Normal jumping distribution (with $c = 2.38/\sqrt{d}$ and $\mathbf{\Sigma}$ set to the true covariance of the target) as obtained from a simulation study for d = 5, 50 and 100 and $N = 2d$, $3d$ and $10d$. The details are as follows. Each figure in the table is based on at least 100 simulations, each consisting of 10^6 draws of each sampler after a burn-in of 10^5 draws. To ensure convergence of DE-MC, the burn-in was extended to at least 500 and 1000 generations for d = 50 and d = 100, respectively. The efficiency is expressed as $100 \times MSE_{RWM}/MSE_{DE\text{-}MC}$, where MSE is mean squared error in the statistic. The statistics were the empirical 2.5, 50 and 97.5-percentiles which, for a d-dimensional target, were determined from the sample for the first and dth variable. The squared error divided by the true variance of the variable did not differ much between these variables and therefore their mean was used in the calculation of the MSE. Because the theoretical MSEs for the 2.5 and 97.5 percentiles are equal, their estimated MSEs were averaged and their average was used to calculate the efficiency under the heading P2.5. It is thus a pooled efficiency for the 2.5 and 97.5 percentiles.
The efficiencies in Table 1 for the Normal target are all above 71% and tend to increase with N/d. For N/d = 10 the estimated efficiencies for the median are all over 100%. This is unexpected for the Normal target, but is not just simulation error.

Table 1. Efficiency (in percentages) of DE-MC with respect to Random Walk Metropolis with optimal Normal jumping distribution for the median (P50) and 2.5% percentile (P2.5) of d-dimensional Normal and Student t_3 distributions.

	Normal						Student t_3			
	d=5		d=50		d=100		d=5		d=50	
N	P50	P2.5	P50	P2.5	P50	P2.5	P50	P2.5	P50	P2.5
$2d$	82	82	91	81	71	74	68	70	88	147
$3d$	100	87	85	80	92	91	86	96	102	191
$10d$	113	86	131	84	127	100	92	99	129	501

NOTE: The estimated MSEs per draw of RWM were, in column order, 20, 59, 174, 396, 335, 823, 12, 962, 121 and 41604.

Possible explanations are that a burn-in of 10^5 draws was not sufficient for RWM when the starting points were drawn from Uniform[-5,15]^d, that DE-MC had the advantage that the initial jumps were much larger than those in RWM and that, for $N > 150$, it was allowed a longer burn-in. The simulated MSEs of RWM (Table 1) were indeed slightly larger than the theoretical ones, but insufficiently larger for a full explanation. (The asymptotic efficiency of RWM compared to independent sampling is $0.3/d$ (Gelman *et al.* 2004), giving MSEs per draw of 167 and 333 for $d = 50$ and 100, whereas in the simulations the MSEs per draw were 174 and 335, respectively (Table 1)).
The acceptance fraction in DE-MC did not vary much with N/d and was remarkably close to that of RWM (0.28 for $d = 5$ and 0.23 – 0.24 for $d = 50$ and 100). The autocorrelations in the Markov chain for each member were similarly close to those in RWM, *e.g.* 0.89 and 0.99 for the lag-1 correlation for $d = 5$ and 50, respectively, and 0.53 and 0.71 for the lag-51 correlation for $d = 50$ and 100 respectively.
The case $N = d + 1$ was investigated separately for $d = 50$ and 100 and resulted in efficiencies of 2-3% or even in clear nonconvergence as judged by the $\hat{R}$-statistic.

3.2 Multivariate Student Target

DE-MC was also compared with Normal jump RWM for multivariate Student distributions with three degrees of freedom. If one would know in advance that the target distribution is Student, then one would of course use a Student jumping distribution rather than a Normal one. However, in practice one does not know the form of the target and often uses the Normal jumping distributions as the default one. The scales c (RWM) and γ (DE-MC) were set such that the acceptance fraction was about 0.28 for $d = 5$ and 0.23 for $d = 50$. Some experimentation learned that the default γ did not need to be changed and that $c = 3.0$ is about right for both values of d.
With 10^5 burn-in, 10^6 draws and initial distribution Uniform[-5,15]^d neither RWM nor DE-MC converged properly as judged on the basis of the $\hat{R}$-statistic. Therefore the problem was simplified by setting the initial distribution to a Normal one with mean and covariance equal to those of the target. Our simulation thus mimics the situation where Normal approximations to the target have been obtained by other means (Gelman *et al.* 2004). With this initial distribution and a burn-in of 10^4 generations for DE-MC, there were no apparent convergence problems. The burn-in for RWM was set to the maximum number of burn-in draws used in DE-MC ($10^5 d$) so as not to favour DE-MC in any sense.
The efficiencies for the Student target in Table 1 are between 68% and 501%, with a clear increase in efficiency with N/d and with higher efficiencies for P2.5 than for P50.

3.3 Normal Mixture Target

The target in this example is a mixture of two Normal distributions

$$\pi(\mathbf{x}) = \frac{1}{3} N_d(-\mathbf{5}, \mathbf{I}_d) + \frac{2}{3} N_d(\mathbf{5}, \mathbf{I}_d)$$

where **5** is the d-vector consisting of fives and $\mathbf{I}_d$ is the d-dimensional identity matrix. The modes were farther apart than in the five-dimensional bimodal example considered in Liang and Wong (2001) with, for d = 5, a distance of $5\sqrt{10}=15.8$ between the modes. This target is notoriously difficult to sample from by RWM. The initial populations for DE-MC were drawn from $N(\mathbf{0}, \mathbf{I}_d)$ and from $N(\mathbf{2.5}, 25\mathbf{I}_d)$, the narrow and the broad distribution in Liang and Wong (2001).
For d = 5 and a burn-in of 1000 generations, DE-MC estimated the expected value (1.667) with a root mean squared error (RMSE) of ~0.023 for both N = 100 and 1000 and for both the narrow and broad initial distribution. The acceptance fraction was ~0.16 in all cases. For d = 10 with N = 1000, DE-MC with default γ converged to around 0.0 for the narrow initial distribution and to 3.7 for the broad initial distribution. Clearly, the sampler is unable to jump from one mode to the other with $\gamma = 2.38/\sqrt{(2d)} = 0.53$. Therefore, we adapted DE-MC such that in every tenth generation $\gamma = 1.0$ so as to allow jumps from one mode region to the other (Fig. 1b). With this adaptation, DE-MC converged to 1.667 with a RMSE of 0.009 and an acceptance fraction of 0.15. Adapted DE-MC reduced the RMSE for the previous d = 5 case from 0.023 to 0.015. These results are based on 100 simulations.

4 Bayesian examples

4.1 One-way Random-Effects Model

The one-way random-effects model is a model for the means of several groups that are linked by the assumption that their expected means are drawn from a common Normal distribution. It can be written as $y_{ij} \sim N(\theta_j, \sigma^2)$ and $\theta_j \sim N(\mu, \tau^2)$ for $j = 1 \ldots J$ groups and, for the jth group, $i = 1 \ldots I_j$. A Bayesian analysis adds prior distributions for the unknowns μ, σ^2 and τ^2 (Liu and Hodges 2003). Commonly used priors are $p(\mu) \propto 1$, $\sigma^2 \sim IG(\alpha,\beta)$, $\tau^2 \sim IG(a,b)$ where IG denotes the inverse-gamma distribution. The analysis shrinks each group sample mean somewhat towards the overall mean. Liu and Hodges (2003) demonstrate that even this simple model may exhibit bimodality in the posterior, at least when there is a prior-data conflict. We re-analyze their peak discharge example, where I = 6 and J = 4, with one of their priors, namely α = 1, β =10, a = 1.85, b = 0.1 and compare the results with WinBUGS 1.4 (Spiegelhalter *et al.* 2003). WinBUGS, short for "Bayesian inference Using Gibbs Sampling", updates each dimension in an iteration by sampling from the full conditional distribution, when available, and by one-dimensional adaptive rejection sampling, slice sampling or current point Metropolis otherwise (Spiegelhalter *et al.* 2003).
To apply DE-MC, the posterior needs to be programmed and the parameters need to be mapped to the vector **x**. We used $\mathbf{x} = (\mu, \log(\sigma^2), \log(\tau^2), \theta_1, \theta_2, \theta_3, \theta_4)$ so d = 7. The problem was expressed in the logarithms of σ^2 and τ^2, because DE-MC is expected to work best in open parameter spaces. The posterior as given in terms of σ^2 and τ^2 by Liu and Hodges (2003, (1)) and Gelman et al. (2004, (5.16)) was multiplied correspondingly by $\sigma^2 \tau^2$ to become

$$p(\mu, \log(\sigma^2), \log(\tau^2), \boldsymbol{\theta}) \propto$$
$$\sigma^{-2\alpha} \exp(-\beta/\sigma^2)\tau^{-2a} \exp(-b/\tau^2)\prod_{j=1}^{J}\left[N(\theta_j \mid \mu, \tau^2)\prod_{i=1}^{I} N(y_{ij} \mid \theta_j, \sigma^2) \right]$$

where $N(.|.)$ denotes the probability density function of the Normal distribution. Note that the normalizing constants of the inverse gamma distribution are not needed because α, β, a and b are fixed. For numerical stability we used the log-posterior. The initial population was drawn from the prior with μ~Uniform[-20, 20]. Because bimodality was expected, γ was set to 1 every 10^{th} generation; otherwise γ was $2.38/\sqrt{(2d)}$.

Table 2. Percentiles of the posterior of $\log(\xi)$ and φ of the one-way random-effects model.

		$\log(\xi)$			φ		
	N	P2.5	P50	P97.5	P2.5	P50	P97.5
True		-0.94	0.98	4.00	0.06	0.31	0.90
WinBUGS		-0.94	0.98	4.00	0.06	0.31	0.91
DE-MC	14	-0.96	0.97	3.98	0.06	0.31	0.90
DE-MC	21	-0.95	0.98	4.11	0.06	0.31	0.91
DE-MC	70	-0.96	0.98	4.15	0.06	0.31	0.91

Table 2 compares the results of WinBUGS and DE-MC with $N = 2d$, $3d$ and $10d$ for $\log(\xi)$ with $\xi = \sigma^2/\tau^2$ and the shrinkage coefficient $\varphi = \sigma^2/(I\tau^2+\sigma^2)$. These analyses used 10^6 iterations after a burn-in of 10^5. The acceptance fraction in DE-MC was 0.21 in all cases. The results of WinBUGS and DE-MC with $N = 14$ differed at most 0.02 from the true values as calculated by analytical integration over $\boldsymbol{\theta}$ and μ and numerical integration over $\log(\sigma^2)$. For $N = 21$ and 70, there is a discrepancy is 0.1 for the 97.5% point of $\log(\xi)$. The median of the estimated 97.5% point of $\log(\xi)$ in 100 re-runs of each DE-MC analysis was 4.00, 4.01 and 4.12 for $N = 14$, 21 and 70, respectively. The systematic discrepancy for $N = 70$ disappears with longer burn-in, as we verified by rerunning the analysis with a tenfold longer burn-in. For completeness we note that 4.12 is the 98.0% point in the true posterior of $\log(\xi)$ and that the 97.5% point in the prior is 7.55. This example showed that large population sizes may require long burn-in for convergence. The bimodality in $\pi(\log(\xi),\varphi)$ expected from Liu and Hodges (2003: Fig. 1d) could not be confirmed, neither in the analytical work nor from the simulations.

4.2 Nonlinear Mixed-Effects Model

This subsection illustrates DE-MC by re-analyzing the Theophylline data presented in Pinheiro and Bates (2000, p. 444) and available in their *nlme* package in R (R Development Core Team 2003) with a nonlinear mixed-effects model. The data consist of the oral doses of the anti-asthmatic drug Theophylline administered to twelve patients and the serum concentrations of Theophylline in these patients at 11 time points over 25 hours after the oral intake. The pharmacokinetics of this drug is modeled by the first-order open-compartment model

Table 3. Percentiles of the posterior of the key-parameters of the first-order open compartment model for the Theophylline data as obtained by a very long WinBUGS run (2 chains, 3 million iterations each, 50% burn-in) and DE-MC with N = 86 and 50,000 generations, 20% burn-in).

	nlme	WinBUGS very long run			DE-MC N = 86		
	estimate	P2.5	P50	P97.5	P2.5	P50	P97.5
lKe	-2.45	-2.57	-2.46	-2.35	-2.57	-2.46	-2.34
lKa	0.47	0.00	0.49	1.01	-0.02	0.48	0.99
lCl	-3.23	-3.37	-3.23	-3.08	-3.37	-3.22	-3.08
$\log(\tau_e^2)$	-21.66	-11.24	-5.60	-3.21	-10.40	-5.56	-3.21
$\log(\tau_a^2)$	-0.87	-1.46	-0.54	0.63	-1.47	-0.55	0.60
$\log(\tau_c^2)$	-3.58	-4.12	-3.20	-2.05	-4.10	-3.19	-2.04
$\log(\sigma^2)$	-0.69	-0.95	-0.69	-0.40	-0.95	-0.69	-0.40

$$\mu_{it} = \frac{D_i k_{ei} k_{ai}}{c_i (k_{ai} - k_{ei})} \left[\exp(-k_{ei} t) - \exp(-k_{ai} t)\right]$$

where μ_{it} is the expected concentration of the ith patient at time t, D_i is the dose of theophylline administered to the ith patient and k_{ei} , k_{ai} and c_i are unknown patient-specific parameters representing the elimination rate, absorption rate and clearance, respectively. For illustration analysis 2 in Pinheiro and Bates (2000, p. 364-365) was mimicked by using the normal likelihood $y_{it} \sim N(\mu_{it}, \sigma^2)$, the independent normal priors $\log(k_{ei}) \sim N(lKe, \tau_e^2)$, $\log(k_{ai}) \sim N(lKa, \tau_a^2)$ and $\log(c_i) \sim N(lCl, \tau_c^2)$ and improper uniform priors for *lKe*, *lKa*, *lCl* and $\log\sigma^2$. Following Gelman *et al.* (2004), the priors for the τ-parameters were chosen improper uniform on the τ-scale, *i.e.* $p(\log(\tau_x^2)) \propto \tau_x$, for x = *e*, *a*, *c*. The total number of parameters in the posterior density is 3+3+1+12×3 = 43 of which 36 random patient-specific ones.

To apply DE-MC, the log-posterior was programmed in the same spirit as in the previous example: the normal log-likelihood for the data y_{it} plus the normal log-likelihood for 36 patient-specific parameters plus the log-prior for the three $\log(\tau^2)$-parameters. The log-priors of the remaining parameters are all zero. For comparison, a WinBUGS 1.4 program was made, which was run using the Bugs-R interface from Gelman *et al.* (2004). Because convergence tended to take long, the first 20% of each run was discarded.

The initial population for DE-MC and the initial values for WinBUGS were drawn from the priors with the improper ones replaced by uniform distributions. The intervals for *lKe*, *lKa, lCl* and $\log(\sigma^2)$ were *nlme*-estimate ± 0.5 (Table 3) and the intervals for τ_e, τ_a, and τ_c were all [0.01, 0.1].

A very long WinBUGS run was to obtain 'true' values to compare the other results to (Table 3). It lasted 11 hours on a 3.2GHz Pentium 4. DE-MC with $\gamma = 2.38/\sqrt{(2d)}$, $N = 2d$ and 50,000 generations (Table 3) yielded close values, the largest discrepancies being for $\log(\tau_e^2)$. The acceptance probability was 0.15; the convergence diagnostic $\hat{R}$ was 1.1.

Table 4 compares WinBUGS and DE-MC with this setup in terms of root mean squared error (RMSE) for runs with the same number of updates. WinBUGS does d updates per iterations (namely one per dimension) whereas DE-MC with $N = 2d$ does

Table 4. Root mean squared error of percentiles for WinBUGS (with 100,000 iterations), DE-MC (N = 86 and 50,000 generations) and Block DE-MC (N = 9 and 50,000 generations with two inner iterations), based on 97, 73 and 100 simulations of 100 simulations with maximum $\hat{R}$ <1.2 and requiring 10.2, 4.4 and 4.2 minutes per simulation on a 3.2 GHz Pentium 4, respectively.

	WinBUGS			DE-MC N = 86			Block DE-MC N = 9		
	P2.5	P50	P97.5	P2.5	P50	P97.5	P2.5	P50	P97.5
lKe	0.003	0.001	0.003	0.003	0.001	0.002	0.002	0.001	0.002
lKa	0.030	0.013	0.047	0.011	0.005	0.017	0.015	0.008	0.024
lCl	0.009	0.003	0.009	0.003	0.001	0.002	0.005	0.002	0.005
$\log(\tau_e^2)$	1.332	0.144	0.064	1.421	0.060	0.045	1.098	0.087	0.033
$\log(\tau_a^2)$	0.010	0.010	0.031	0.019	0.010	0.030	0.008	0.006	0.018
$\log(\tau_c^2)$	0.010	0.012	0.028	0.014	0.010	0.019	0.007	0.008	0.017
$\log(\sigma^2)$	0.004	0.002	0.003	0.003	0.002	0.006	0.003	0.001	0.002

2d updates per generation (namely one per member chain). WinBUGS was run 20 times with 5 chains of 100,000 iterations each. In three of the runs the maximum $\hat{R}$ over the parameters in Table 4 was over 1.2. After inspection of these runs, three clearly aberrant chains were discarded. Table 4 is based on the remaining 97 chains. Of the 100 DE-MC runs 27 gave maximum $\hat{R}$ > 1.2 and were discarded. Per successful chain (all $\hat{R}$ <1.2), WinBUGS took 1.7 times longer than DE-MC. Compared to WinBUGS, the RMSE of DE-MC is up to a factor of 4 lower for the location parameters *lKe*, *lKa* and *lCl* and up a factor 2 higher for the variance parameters.

Tuning γ so that DE-MC has an acceptance rate of 0.23 gave $\gamma = 1.7/\sqrt{(2d)}$. Now 80 out of 100 DE-MC runs had $\hat{R}$ < 1.2. The RMSEs of the location parameters did not change much. Nine out of the twelve entries for the variance parameters decreased to values below those of WinBUGS, the remaining three being within a factor of 1.5. This example is continued in the next section.

5 DE-MC variants

5.1 Crossover and block updating

In high dimensions it may not always be optimal to sample all d elements of $\mathbf{x}_i$ simultaneously. With the crossover mechanism of DE (Storn and Price 1997), sampling takes place in lower dimensional spaces. Before the proposal is compared with $\mathbf{x}_i$, it is modified by crossover. The simplest crossover scheme is binomial in which each element $\mathbf{x}_{pj}$ (j = 1… d) of the proposal is replaced by $\mathbf{x}_{ij}$ with probability 1-CR, with the extra restriction that not all elements are replaced. CR is termed the crossover probability. The sampler described so far thus corresponds to CR = 1. The resulting DE-MC sampler still converges to the required target, as can be seen by noting that the sampler is then a doubly component-wise Metropolis algorithm with both members and dimensions as components. CR = 0 corresponds by its definition in

Storn and Price (1997) to single dimension updating, as in Gibbs sampling. There is however a difference with Gibbs sampling. The proposals in Gibbs sampling are drawn from the appropriate conditional distribution. The proposals in DE-MC for a particular dimension are generated, after convergence, from differences of two numbers drawn from the marginal distribution for that dimension. This shows that crossover in DE-MC would work best (as in Gibbs) if the dimensions that are updated in separate steps are independent. A non-random version of crossover is to split the parameter vector in blocks and to update the blocks in turn. DE-MC can, of course, be applied to some elements to **x**, whereas the others are updated by Gibbs sampling.

Table 4 (last three columns) shows the possible advantages of block updating for the nonlinear mixed-effect model of section 4.2. Here the 43-dimensional parameter vector was split in 15 blocks. The blocks come naturally in this example as the time curve of the expected concentration depends on three correlated parameters for each patient, whereas the parameters of different patients are expected to be uncorrelated. This yielded twelve blocks, one per patient. The location parameters *lKe*, *lKa* and *lCl* also formed a block of three parameters, as did the τ-parameters. The final block consisted of σ^2 only. To reduce the correlation between the location parameters and the patient-specific parameters, the latter parameters were expressed as deviation from the former. This also reduced the correlation between the new patient-specific parameters (for $\log(k_{ei})$ and $\log(c_i)$ from 0.8 to ca. 0.4). This transformation, which was also applied in the WinBUGS runs of section 4.2, does not affect the full-space updates of DE-MC.

If each block update would require the full posterior, each full cycle of block updates would require 15 times more computing time than full-space DE-MC, thus allowing for only 3333 instead of the 50,000 generations in the setup of Table 4. Fortunately there are two ways to gain efficiency. First, the population size N can be decreased to 9, as the maximum block size is 3. With $N = 9$, one can do 9.6 times more generations in the same time. Second, updating a block requires only those parts of the posterior that depend on the parameters of that block. This feature is the key to the efficiency of the one-dimensional updates in WinBUGS. In the example, updating the block of a particular patient does not requires the likelihood contributions of the other patients and updating the τ-parameters does not require the likelihood at all. The possible gain for these blocks is not a factor of twelve (the number of patients) but six, because the block-specific posteriors need to be calculated both for the current parameters and the proposal, whereas full space DE-MC can re-use the posterior of the current point. For the same reason, the full posterior of the remaining two blocks must be evaluated 3/2 times as often. In our implementation, the resulting gain is a factor of 2.4, which can be increased to a factor of 3.1 by carrying out two inner iterations of DE-MC per block update[2]. Without extra costs, one of the two expensive inner iterations for the location parameters and for σ^2 were replaced by full-space DE-MC steps, because these parameters did already well in DE-MC (Table 4). With these optimizations, one generation of block DE-MC with $N = 9$ could be done in the same time as one generation of DE-MC with $N = 86$. Table 4 shows the results of block DE-MC with $\gamma = 2.38/\sqrt{(2d_b)}$, where d_b the number of parameters in the block (1, 3 or 43). All 100 runs were successful. Compared to full-space DE-MC, all variance parameters, *lKe*

[2] Two inner iterations maximize the acceptance probability per unit of the computing cost if the acceptance rate per inner iteration is between ca. 0.18 and 0.41; for lower acceptance rate, the maximum is at three of more inner iterations.

and σ^2 have the same or lower RMSE; *lKa* and *lCl* have up to a factor of 2.5 higher RMSE. The RMSEs of block DE-MC were all lower than those of WinBUGS. With 50% burn-in, the RMSEs were all ~20% worse than those reported in Table 4.
The improvement in convergence of block DE-MC over WinBUGS and full-space DE-MC was even more pronounced when the intervals for the τ-parameters in the initial population were widened to [0.01, 0.5]. This leads to many unlikely initial values for the patient-specific parameters. Nevertheless, block DE-MC in this setup converged, whereas WinBUGS and full-space DE-MC did not.
In the example each variance component could have been drawn directly from its full conditional distribution, but for illustration of the power and flexibility of DE-MC, only DE-MC updates were used. In general, it seems natural to exploit conjugacy, particularly multivariate conjugacy, where possible, and use DE-MC as a Metropolis-within-Gibbs step otherwise.

5.2 Simulated Tempering and Annealing Variants

DE versions for simulated annealing and simulated tempering are obtained by introducing a temperature ladder (Liang and Wong 2001). Fig. 2 shows a simple version in which the temperature ladder depends only on generation. For simulated annealing and tempering, the temperature runs from a large value to 0 and 1, respectively, according to a particular cooling schedule (Schmitt 2004). An interesting feature of these DE-MC variants is that the proposals automatically become less variable with lower temperature.

6 Discussion

DE-MC as proposed in this paper is one of the simplest adaptive MCMC methods, yet attains high efficiency with respect to the Normal jump Metropolis algorithm (Table 1). The scale and orientation of the jumps in DE-MC (2) automatically adapt themselves to the variance-covariance matrix of the target distribution (Section 2.4). It is precisely this that each point in the population learns in DE-MC from the others, nothing more and nothing less. Neither the location nor the fitness of the other points is used in the proposal scheme.

The optimal value of γ suggested by analogy with Normal jump Metropolis with Normal target worked well with the Student target and in the examples. Apparently the differences in (2) sufficiently bear out the increased roughness of the Student target, even though the differences themselves are no longer Student distributed, as the Student distribution is not closed under subtraction. In the nonlinear mixed-effects model, γ needed to be decreased somewhat to get an optimal acceptance rate. The suggested default value of $\gamma = 2.38/\sqrt{(2d)}$ performed well in the bock updating variant of DE-MC (Table 4). If blocks are strongly correlated, γ may need to be decreased.

DE-MC worked well also for bimodal distributions, albeit with the adaptation of the use of $\gamma = 1.0$ every 10^{th} generation. This property of DE-MC is expected to generalize to multimodal distributions; as soon as one point is in a modal region (a large N and wide initial population will make this more likely), more points can jump into it if $\gamma = 1$: any point $\mathbf{x}_i$ can jump into the modal region by proposal (2) if one of

$\mathbf{x}_{R1}$ and $\mathbf{x}_{R2}$ is into it and the other is close to $\mathbf{x}_i$ (Figure 1b). On the other hand, if the initial population covers just a single modal region, there is no chance that other modes that are far away can be reached. It is perhaps better to set γ slightly less than 1, *e.g.* $\gamma = 0.98$. This doubles the number of possible trial vectors compared to $\gamma = 1$ (Lampinen and Zelinka 2000). These observations plea for choosing the initial population not too small in size and not too narrow in distribution when multimodality is a possibility. But also note that each point of the initial population needs time to move to likely values. Large populations thus require more computer time to converge than small ones. The advise is thus to choose $N = 2d$ or $3d$ for simple unimodal targets and $N = 10d$ to $20d$ when the target is more complicated.

Parallel adaptive sampling (Gilks *et al.* 1994, Roberts and Gilks 1994) also uses proposals of the form of equation (2), with $\mathbf{e} = \mathbf{0}$. The treatment of γ forms the difference with DE-MC. Parallel adaptive sampling continues with Gibbs sampling of γ, whereas DE-MC does a Metropolis step with a fixed value of γ. In practice the conditional distribution required for Gibbs sampling γ will often not be available in closed form or it will not be easy to sample from directly, so that the Gibbs sampling step must be replaced by one or more Metropolis-Hasting steps. DE-MC is thus a form of parallel adaptive direction sampling with the Gibbs sampling step replaced by one Metropolis step with a pre-chosen value of γ. The authors of adaptive direction sampling apparently did not notice that the vector differences also contained much information on the scale of the target.

After submission, we learned that Strens *et al.* (2002) also explored the combination DE and MCMC in a comparison of seven MCMC algorithms for sampling a multimodal density. DE-MC, in their paper alternated with one-dimensional Metropolis updates, came out best. Strens *et al.* (2002) chose γ random with $\log(\gamma) \sim N(0, \log(4))$. We extend Strens *et al.* (2002) in providing theory for the optimal choice of γ. In early simulations that we did (not shown), a random γ calibrated to an acceptance rate of 0.23 always lowered the efficiency of DE-MC compared to a fixed γ yielding this acceptance rate. Strens *et al.* (2002) did not consider crossover. Our simulations and applications confirm once more the power of DE-MC.
Our computer experiments show that the rate of convergence of DE-MC is comparable to or higher than that of RWM. When started from an overdispersed initial population, DE-MC starts with large jumps so that it is expected to reach the centre of the distribution more quickly than fixed jump Metropolis. Both samplers converged quickly for Normal targets but quite slowly for Student targets. This rate difference is known for Metropolis from Mengersen and Tweedie (1996). A theoretical analysis of the rate of convergence of DE-MC is much desired. Monitoring of convergence with the convergence diagnostic $\hat{R}$ of Gelman et al. (2004) worked well in practice.

Gibbs sampling dominates in Bayesian data analysis, (a) because of the availability of excellent software (WinBUGS), (b) because it is efficient if components are independent and (c) because the alternatives are more cumbersome to use. Poor mixing is a general problem in Gibbs samplers despite clever tricks to improve it (Gelman et al. 2004, section 11.8). Outside the generalized linear model context WinBUGS is limited to one-dimensional updates. In contrast, DE-MC does simple and efficient multidimensional updates. By applying DE-MC in small blocks, the population size can be kept small. With its multidimensional updates, block DE-MC with $N = 9$ outperformed WinBUGS in the example of a nonlinear random effects

model. The example showed the usefulness of DE-MC in a multi-chain Gibbs sampler.

Laskey and Myers (2003) envisioned population MCMC versions that come close to independence sampling by generating proposals from a semi-parametric model of the current population. Being a nonparametric version of RWM, DE-MC is not such a greedy algorithm. This is an advantage for exploration of the space to find otherwise easily missed modes, but a disadvantage in terms of speed of convergence. The challenge is to find more greedy variants of DE-MC that retain the robustness and simplicity of the version presented here.

Acknowledgements

The author thanks Julius van der Werf for a course on Genetic Algorithms that inspired me to integrate DE and MCMC, Martin Boer and João Paulo for help, Kate Cowles for providing the peak discharge data, and Eligius Hendrix, Hilko van der Voet, Kenneth Price, Andrew Gelman and an anonymous reviewer for comments on the manuscript.

References

Gelman A., Carlin J. B., Stern H. S., and Rubin D. B. 2004. Bayesian data analysis, 2nd edition. London, Chapman & Hall.

Gilks W. R., Richardson S., and Spiegelhalter D. J. 1996. Markov chain monte carlo in practice. London, Chapman & Hall.

Gilks W. R., and Roberts G. O. 1996. Strategies for improving MCMC. In Markov chain monte carlo in practice (eds. W. R. Gilks, S. Richardson and D. J. Spiegelhalter), London, Chapman & Hall, pp. 89-114.

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

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

Lampinen J. 2001 A bibliography of Differential Evolution algorithm. Lappeenranta University of Technology, Lappeenranta, www.lut.fi/~jlampine/debiblio.htm.

Lampinen J., and Zelinka I. 2000. On stagnation of the Differential Evolution algorithm. Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing, Brno, pp. 76–83.

Laskey K. B., and Myers J. W. 2003. Population Markov Chain Monte Carlo. Mach. Learn. 50: 175-196.

Liang F. 2002. Dynamically weighted importance sampling in Monte Carlo computation. J. Am. Statist. Ass. 97: 807-821.

Liang F. M., and Wong W. H. 2001. Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models. J. Am. Statist. Ass. 96: 653-666.

Liu J., and Hodges J. S. 2003. Posterior bimodality in the balanced one-way random-effects model. J. R. Statist. Soc. B 65: 247-255.

Lunn D. J., Thomas A., Best N., and Spiegelhalter D. 2000. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. Statist. Comp. 10: 325-337.

Mengersen K., and Robert C. P. 2003. IID sampling using self-avoiding population Monte Carlo: the pinball sampler. In Bayesian Statistics 7 (eds. J. M.

Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West), Oxford, Clarendon Press, pp. 277-292.

Mengersen K. L., and Tweedie R. L. 1996. Rates of convergence of the Hastings and Metropolis algorithms. Ann. of Statist. 24: 101-121.

Pinheiro J. C., and Bates D. M. 2000. Mixed-Effects Models in S and S-PLUS. New York, Springer Verlag.

Price K. 1999. An introduction to differential evolution. In New Ideas in Optimization (eds. D. Corne, M. Dorigo and F. Glover), London, McGraw-Hill, pp. 79-108.

Price K., and Storn R. 1997. Differential Evolution. Dr Dobb's Journal 264: 18-24.

R Development Core Team 2003. R: a language and environment for statistical computing. Vienna, Austria, R Foundation for Statistical Computing. www.r-project.org.

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

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

Schmitt L. M. 2004. Theory of genetic algorithms II: models for genetic operators over string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. Theor. Comp. Sci. 310: 181-231.

Spiegelhalter D., Thomas A., Best N., and Lunn D. 2003. WinBUGS User Manual version 1.4. www.mrc-bsu.cam.ac.uk/bugs.

Storn R., and Price K. 1995 Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute, Berkeley, TR-95-012, http://www.icsi.berkeley.edu/~storn/litera.html.

Storn R., and Price K. 1997. Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Opt. 11: 341 - 359.

Strens M., Bernhardt M., and Everett N. 2002. Markov chain Monte Carlo sampling using direct search optimization. ICML, Sydney, pp. 602-609.

Waagepetersen R., and Sorensen D. 2001. A tutorial on reversible jump MCMC with a view toward applications in QTL-mapping. Int. Statist. Rev. 69: 49-61.