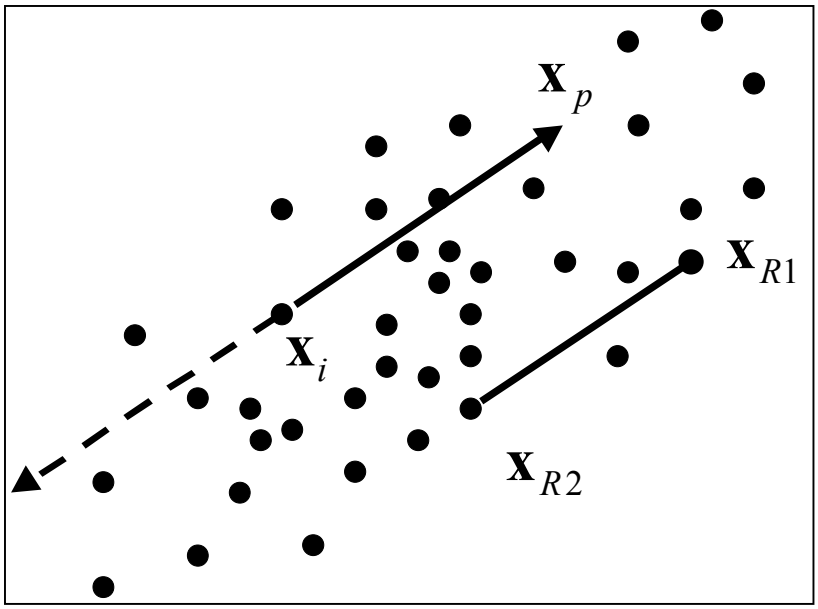
WAGENINGEN UR
For quality of life

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

C.J.F. ter Braak



April 2004

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

Report number 010404

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

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. In this paper the essential ideas of DE and MCMC are integrated into 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 multiple of the differences of two random parameter vectors that are currently in the population. Simulations and examples illustrate the potential of DE-MC. The advantage of DE-MC over conventional MCMC are simplicity, speed of calculation and convergence, even for nearly collinear parameters and multimodal densities.

Keywords: Evolutionary Monte Carlo; Metropolis algorithm; Mixture model; Population Markov Chain Monte Carlo; Simulated Annealing; Simulated Tempering

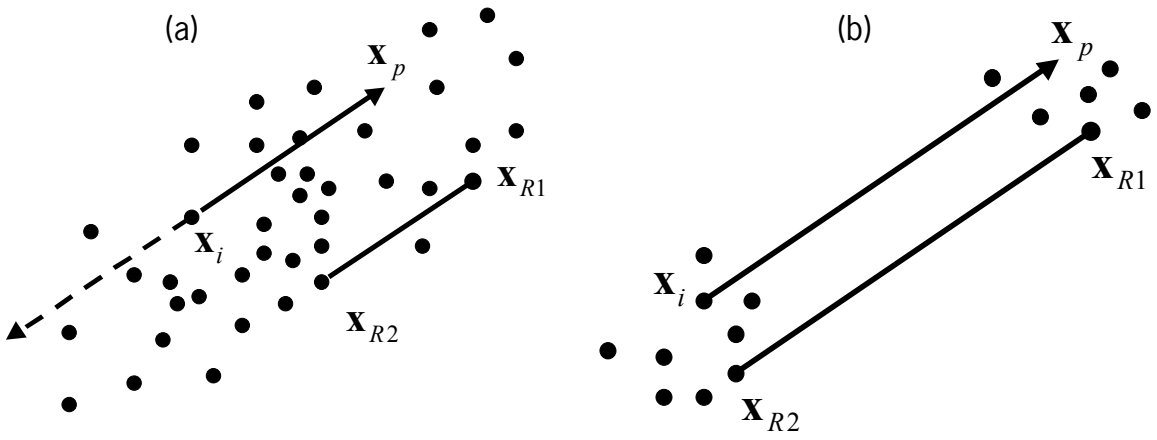
Introduction

In this paper the genetic algorithm called Differential Evolution (DE) (Price and Storn, 1997; Storn and Price, 1997) for global optimization over real parameter space is combined with Markov Chain Monte Carlo (MCMC)(Gilks *et al.*, 1996). 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, Carlin, Stern and Rubin (2004) provide an introduction 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 not new (Liang and Wong, 2001; Liang, 2002; 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). But the combination of DE and MCMC is new and solves an important problem in MCMC in real parameter spaces, namely that of choosing an appropriate scale and orientation for the jumping distribution.

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 $d(d-1)/2$ 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 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 also show how the method can be modified to 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 data examples, one where bimodality is suspected and one specifying a nonlinear mixed-effects model.

Figure 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 $R1$ and $R2$ by (2) with $\gamma = 2.4/(2\times2)^{1/2} = 1.2$ in (a) and $\gamma = 1.0$ in (b) and $\mathbf{e} = \mathbf{0}$ in both. The dashed arrow in (a) points to the proposal when $R1$ would have been drawn after $R2$. The reverse jump from $\mathbf{x}_p$ to $\mathbf{x}_i$ is obtained by translating the dashed arrow to $\mathbf{x}_p$.



Theory

Random walk Metropolis

In the random walk Metropolis algorithm with a multivariate normal jumping distribution (RWM), a single d-dimensional parameter vector $\mathbf{x}$ is updated by (a) generating a proposal $\mathbf{x}_p = \mathbf{x} + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, c^2\boldsymbol{\Sigma})$, (b) calculating the Metropolis ratio $r = \pi(\mathbf{x}_p)/\pi(\mathbf{x})$ where $\pi(.)$ is the target distribution 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 $\pi(.)$ as unique stationary distribution. In Bayesian analyses, $\pi(.) \propto$ prior $\times$ likelihood. Roberts and Rosenthal (2001) and Gelman et al. (2004) summarize the guidelines for the choice of c and $\boldsymbol{\Sigma}$. Optimally, $\boldsymbol{\Sigma} = \text{cov}(\mathbf{x})$ with $\mathbf{x} \sim \pi(.)$ and c must be set 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.

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 later on, as it is a side-issue in this paper. 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).

Differential Evolution Markov Chain

In order to turn DE into a Markov chain for Bayesian analysis, 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). 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 **e** is drawn from a symmetric distribution with a small variance compared to that of the target. For speed of computation we use **e**~Uniform[$-b,b$]^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 Figure 2.

Theorem. DE-MC yields a Markov chain with $\pi(.)^N$ as its unique stationary distribution.

Proof. The proof consists of three parts. (a) As other population MCMCs (Laskey and Myers, 2003), DE-MC is an N-component Metropolis-Hastings algorithm, in which each component is updated in turn conditionally on the other components. As component-wise Metropolis algorithms converge if each component converges to the correct conditional distribution (Waagepetersen and Sorensen, 2001), it is sufficient to consider each component separately. (b) A single component DE-MC generates a random walk so that, except for trivial exceptions, the chain is aperiodic and not transient (Gelman *et al.*, 2004). It is also irreducible because any state can be reached with positive probability as is guaranteed by the uniform random number in (2). Each component has therefore a unique stationary conditional distribution. (c) The jumps in each component retain detailed balance with respect to $\pi(.)$ at each step. The probability from the jump of $\mathbf{x}_i$ to $\mathbf{x}_p$ is equal to the reverse jump, as we can see from

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

Figure 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 of the population; x_p = proposal d-vector $\mathbf{x}_p$, and fitness(.)= $\pi(.)$, 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++ ){ /* start loop through generations */
    Temperature = CoolingSchedule(s, N_generation)
    for (i=0; i<N ; i++) { /* start loop through 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);
    /* DE1 strategy in Storn & Price 1995 TR-95-012 */
        for (j=0;j<d; j++) x_p= 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) number */
        if ( log(r) > Temperature*log(Uniform(0,1) ) swap(X[i] , x_p);
    } /*end of loop through population */
    Record(X);
} /* end loop through generations */
```

/* Summarize the recorded sample */

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 thus retained 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, the detailed balance also holds in terms of arbitrary sets, as required for reversibility of the Markov chain (Waagepetersen and Sorensen, 2001). Therefore, the unique stationary conditional distribution of each component is $\pi(.)$, which concludes the proof.

Why does DE-MC work in practice?

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

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

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

$$\mathrm{ave}(\mathbf{x}_i) \to \boldsymbol{\mu} \text{ and } \mathrm{ave}\{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\} \to 2\boldsymbol{\Sigma} \qquad \text{for } N \to \infty$$

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

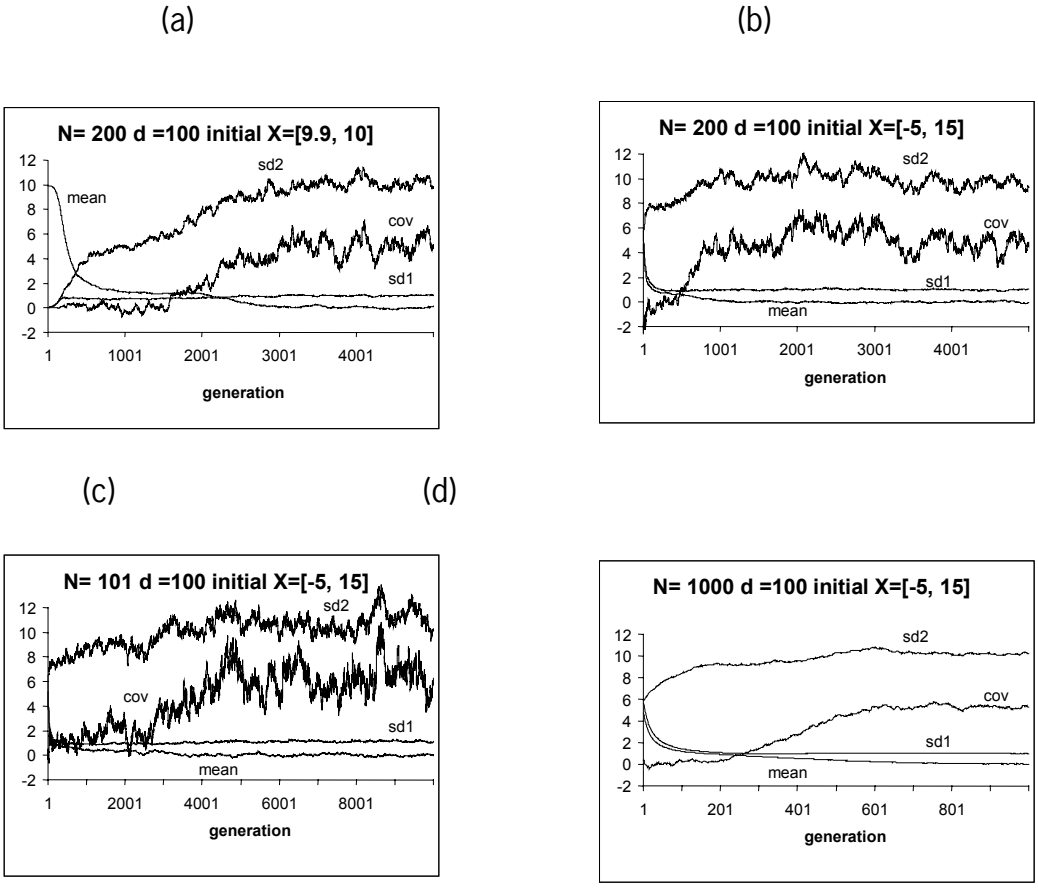
For large N and small b, the proposal (2) thus looks like $\mathbf{x}_p = \mathbf{x}_i + \gamma\,\boldsymbol{\varepsilon}$ with $\mathrm{E}(\boldsymbol{\varepsilon})=\mathbf{0}$ and $\mathrm{cov}(\boldsymbol{\varepsilon})= 2\boldsymbol{\Sigma}$, the covariance matrix of the target. In particular, if $\pi(.)$ 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 $N \le d$? 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 b is small. In the next section the effect of N on the efficiency of DE-MC is studied via simulation for $N > d$.

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 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 $\gamma = 2.38/\sqrt{(2d)}$ and $b = 10^{-4}$, unless noted otherwise. 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 (Figure 2).

Figure 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 $\mathbf{X}$. 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.



Multivariate Normal target

Figure 3 shows how the sample means, standard deviations and correlations of the population $\mathbf{X}$ evolve in time for d = 100. Figure 3a and Figure 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 (Figure 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 Figure 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 Figure 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 Figure 3b than in Figure 3a. 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.
Figure 3b-d contrast different population sizes (N = 200, 101, 1000). Note that the horizontal scales match in terms of the number of draws. These populations converged about equally fast to likely values in terms of number of generations. Consequently smaller populations converged faster than larger ones in terms of 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 (Figure 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 Figure 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 Figure 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 $c = 2.38/\sqrt{d}$ (Roberts and Rosenthal, 2001), 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. 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 variances per point of 167 and 333 for d = 50 and 100, whereas in the simulations the mean squared errors per point were 174 and 335, respectively (Table 1)).

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 variances per point (10^{-6}MSE) of RWM were, in column order, 20, 59, 174, 396, 335, 823, 12, 962, 121 and 41604.

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 difference between MSE and variance across simulations.

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 and γ were set such that the acceptance fraction was about 0.28 for d = 5 and 0.23 for d = 50. Some experimentation learned that c = 3.0 is about right for both values of d and that γ did not need to be changed.

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 a comparison of MSE and variance. Therefore the problem was simplified by setting the initial distribution to a Normal one with mean and covariance equal to those of the target. 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.

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. The initial populations 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 (Figure 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.

Bayesian examples

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 $\alpha = 1$, $\beta = 10$, $a = 1.85$, $b = 0.1$ and compare the results with WinBUGS (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.
Table 2 compares the results of WinBUGS and DE-MC with $N = 10d = 70$ for the $\log(\xi)$ with $\xi = \sigma^2/\tau^2$ and the shrinkage coefficient $\varphi = \sigma^2/(I\tau^2+\sigma^2)$. Both analyses used 10^6 iterations after a burn-in of 10^5. The acceptance fraction in DE-MC was 0.21. The MC-errors for $\log(\xi)$ and φ are 0.003 and 0.0006, as issued by WinBUGS
for the mean on the basis of the block mean method (Spiegelhalter *et al.*, 2003), and 0.013 and 0.0022, as determined for the median on the basis of 100 independent DE-MC runs with the same settings. The largest discrepancy is for the 97.5% point of $\log(\xi)$, where DE-MC gives a larger value. To further investigate this 97.5% point, we re-ran DE-MC 100 times with $N = 500$ with a burn-in of 2000 generations and keeping the next 10^6 samples. The last row of Table 2 reports the mean of these runs. The 95% interval for the 97.5% point of $\log(\xi)$ in this analysis was [4.0461, 4.2083]. The value given by WinBUGS is just outside. The larger value given by DE-MC makes sense as the prior for τ^2 allows very small values which are not contradicted by the data, yielding some mass at very large values of ξ.

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

		$\log(\xi)$			φ		
	N	median	2.5	97.5	median	2.5	97.5
WinBUGS		0.978	-0.942	4.004	0.307	0.061	0.914
DE-MC	70	0.976	-0.960	4.148	0.307	0.060	0.913
DE-MC	500	0.985	-0.938	4.122	0.309	0.061	0.911

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 pharmakinetics of this drug is modeled by the first-order open-compartment model

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

where μ_{it} is the expected concentration of the ith patient at time t, D_i is the dose of 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 $\log\sigma^2$. It is often difficult to choose a good prior for the τ-parameters

(Gelman et al. 2004). Therefore, two analyses were run which differed in the priors for the τ-parameters, one analysis in which these hyperparameters are improper uniform on the τ^2-scale, *i.e.* $p(\log(\tau_x^2)) \propto \tau_x^2$ and another in which these hyperparameters are 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 $3+3+1+12\times3 = 43$ 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. The initial population was drawn from the priors, with the improper ones given a broad range; for *lKe*, *lKa* and *lCl* and $\log(\sigma^2)$ a range of 3 and those for $\log(\tau_x^2)$ are a range of 10. DE-MC was applied with $N = 1000$ with a burn-in of 5000 generations and another 5000 generations of which every 5^{th} was kept. The acceptance probability was 0.17. Table 3 shows the mean of four independent runs. The MC-error in the median is in the last digit for $\log(\tau_e^2)$ or beyond for the other parameters. Table 3 shows a great correspondence of the *nlme* estimates with the Bayesian estimates and only a minor effect of the different τ-priors. Note that the model can also be analysed with WinBUGS, in particular with the help of the front-end PKBugs (Lunn *et al.*, 2002).

Table 3. Median and percentiles of the posterior of the key-parameters of the first-order open compartment model for the Theophylline data.

	nlme	prior uniform in τ^2			prior uniform in τ		
	estimate	median	2.5	97.5	median	2.5	97.5
lKe	-2.45	-2.45	-2.94	-0.91	-2.46	-3.20	-0.64
lKa	0.47	0.47	-1.16	1.62	0.47	-1.68	1.60
lCl	-3.23	-3.22	-3.42	-2.75	-3.22	-3.42	-1.69
$2\log(\tau_e)$	-21.66	-4.20	-7.46	1.04	-5.07	-8.92	0.71
$2\log(\tau_a)$	-0.87	-0.31	-2.02	1.89	-0.49	-2.98	1.51
$2\log(\tau_c)$	-3.58	-2.89	-4.14	-0.73	-3.12	-4.35	-0.97
$2\log(\sigma)$	-0.69	-0.67	-0.96	1.63	-0.66	-0.94	2.47

DE-MC variants

Crossover

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 most simple crossover is binomial in which each element $\mathbf{x}_{pj}$ ($j = 1 \ldots d$) 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 big 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 cross-over in DE-MC would work best (as in Gibbs) if the dimensions that are updated in separate steps are independent.

DE-MC can, of course, be applied to some elements to $\mathbf{x}$, whereas the others are updated by Gibbs sampling.

Simulated tempering and annealing variants

DE versions for simulated annealing and simulated tempering are obtained by introducing a temperature ladder (Liang and Wong, 2001). Figure 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.

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 in all examples in yielding the acceptance fraction predicted by the analogy and needed no change when the target distribution was Student rather than Normal (Section 3.2). 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. 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. 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 so that convergence is more in terms of number of generations than in number of function evaluations (Figure 3). Large populations thus require more computer time to converge than small ones. The advise is thus to choose $N = 3d$ 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.

On the basis of our computer experiments, we conjecture that the rate of convergence of DE-MC is comparable or higher than that of Normal jump Metropolis. 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 can be done most safely by replicating the entire procedure a number of times as we did in the examples and to monitor the between and within variances (Gelman et al. 2004, p. 296).

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. In one-way random-effects models WinBUGS was more efficient than DE-MC in terms of Monte Carlo error, but there was also an indication that WinBUGS missed part of the bimodal target distribution. Poor mixing is a general problem in Gibbs samplers despite clever tricks to improve it (Gelman et al. 2004, section 11.8). DE-MC is a contender to Gibbs in ease of programming and for its mixing properties.

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. DE-MC selects just two other points from the population and does use the known values of the target. Inspiration how to utilize the information of more than two other points and of the target values may come from adaptive direction sampling (Gilks *et al.*, 1994) and local optimization-based Metropolis (Liu *et al.*, 2000; Gustafson *et al.*, 2004). Another line of research is to investigate whether other genetic algorithms than DE-MC can be turned into MCMC versions for simple and efficient Bayesian computing.

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 Hilko van der Voet and Andrew Gelman 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), pp. 89-114. London: Chapman & Hall.

Gilks, W. R., Roberts, G. O. and George, E. I. (1994) Adaptive Direction Sampling. *The Statistician*, **43**, 179-189.

Gustafson, P., Macnab, Y. C. and Wen, S. (2004) On the value of derivative evaluations and random walk suppression in Markov Chain Monte Carlo algorithms. *Statist. Comp.*, **14**, 23-38.

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. Technical, Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing. Available at http://www.lut.fi/~jlampine/debiblio.htm.

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.

Liu, J. S., Liang, F. M. and Wong, W. H. (2000) The multiple-try method and local optimization in metropolis sampling. *J. Am. Statist. Ass.*, **95**, 121-134.

Lunn, D. J., Best, N., Thomas, A., Wakefield, J. and Spiegelhalter, D. (2002) Bayesian analysis of population PK/PD models: General concepts and software. *J. Pharmacokin. Pharmacodyn.*, **29**, 271-307.

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. L. and Tweedie, R. L. (1996) Rates of convergence of the Hastings and Metropolis algorithms. *Ann. 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. and Storn, R. (1997) Differential Evolution. *Dr. Dobb's J.*, **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.

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 TR-95-012. Berkeley: International Computer Science Institute,

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.

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