Geophys. J. Int. (1999) **138**, 479-494

Geophysical inversion with a neighbourhood algorithm –I. Searching a parameter space

Malcolm Sambridge

Research School of Earth Sciences, Institute of Advanced studies, Australian National University, Canberra, ACT 0200, Australia. E-mail: malcolm@rses.anu.edu.au

Accepted 1999 March 12. Received 1999 March 5; in original form 1998 December 3

SUMMARY

This paper presents a new derivative-free search method for finding models of acceptable data fit in a multidimensional parameter space. It falls into the same class of method as simulated annealing and genetic algorithms, which are commonly used for global optimization problems. The objective here is to find an ensemble of models that preferentially sample the good data-fitting regions of parameter space, rather than seeking a single optimal model. (A related paper deals with the quantitative appraisal of the ensemble.)

The new search algorithm makes use of the geometrical constructs known as Voronoi cells to drive the search in parameter space. These are nearest neighbour regions defined under a suitable distance norm. The algorithm is conceptually simple, requires just two 'tuning parameters', and makes use of only the rank of a data fit criterion rather than the numerical value. In this way all difficulties associated with the scaling of a data misfit function are avoided, and any combination of data fit criteria can be used. It is also shown how Voronoi cells can be used to enhance any existing direct search algorithm, by intermittently replacing the forward modelling calculations with nearest neighbour calculations.

The new direct search algorithm is illustrated with an application to a synthetic problem involving the inversion of receiver functions for crustal seismic structure. This is known to be a non-linear problem, where linearized inversion techniques suffer from a strong dependence on the starting solution. It is shown that the new algorithm produces a sophisticated type of 'self-adaptive' search behaviour, which to our knowledge has not been demonstrated in any previous technique of this kind.

Key words: numerical techniques, receiver functions, waveform inversion.

1 INTRODUCTION

In the past decade and a half, Monte Carlo (MC) methods (Hammersley & Handscomb 1964) have enjoyed a resurgence in popularity amongst geophysicists, particularly in application to inverse problems. The earliest use of MC methods was for probabilistic, or randomized, searching of a (finite-dimensional) parameter space. Notable papers were by Keilis-Borok & Yanovskaya (1967), Press (1968), Wiggins (1969), Anderssen (1970), and Anderssen & Senata (1971). These studies were well known to geophysicists, but it was only when the necessary computational power became widely accessible (around the beginning of the 1980s) that interest in Monte Carlo methods became widespread. Prior to that (and continuing to this day), many of the inverse problems posed by geophysicists were addressed using linear inverse theory (see Menke 1989; Tarantola 1987; Parker 1994, for comprehensive summaries).

The increase in computational power coincided with the realization that, for many interesting problems, it was inconvenient, or simply not possible, to use a linearized approximation. Linearization involves the calculation of partial derivatives of data with respect to model parameters (or functionals), and for some non-linear problems these can either be difficult to calculate, or have a very limited range of applicability. [For a recent discussion on the role of non-linearity in inverse problems see Snieder (1998).] Derivative free 'direct search' methods were seen as an attractive alternative. This was especially true in seismic problems where high frequency body waveforms were being used to constrain earth structure from length-scales of geophysical exploration to those of regional tectonics.

The early direct search methods were based on uniform pseudo-random sampling of a parameter space, sometimes with constraints imposed (as in Wiggins 1969). The inefficiency of this type of approach makes it impractical once the dimensionality of the parameter space increases. Rothman (1985, 1986) introduced the method of simulated annealing (SA) (Kirkpatrick *et al.* 1983) into geophysics, which is a stochastic direct search method designed primarily for global optimization problems. For a survey of papers see Sen & Stoffa (1995), and for detailed descriptions of the method see Aarts & Korst (1989).

More recently, genetic algorithms have made the jump from

their origins in the computer science literature (Holland 1975; Goldberg 1989) to geophysical problems (Stoffa & Sen 1991; Sambridge & Drijkoningen 1992). Again the objective is often stated as seeking a model giving a globally optimal data misfit value (within a pre-defined finite-dimensional parameter space). This type of approach has found many applications in recent years (for surveys see Gallagher & Sambridge 1994; Sen & Stoffa 1995). As with simulated annealing many variants of the basic method have been developed. These often result in the introduction of extra 'control' parameters (in addition to those inherent in the basic method) which must be empirically tuned for each application. The goal of the tuning process is usually to achieve computational efficiency, and some level of robustness against entrapment in local minima.

In this paper we present an entirely new class of direct search methods which we show has some distinct advantages over previous methods. The method is conceptually simple with at most two control parameters, but is able to exhibit complex self-adaptive behaviour in searching a parameter space. Unlike previous methods, the new approach is not designed specifically to perform global optimization, however, we show that in our test problem it performs just as well as a previous approach in this respect. The objective here is to sample the region of parameter space that contains models of acceptable data fit (or any other objective function), and to then to extract robust information from the ensemble of models obtained.

This paper deals largely with the first stage of this problem, i.e. generating an acceptable ensemble, and only in a qualitative manner with the second stage, i.e. appraising the ensemble. A related paper, (Sambridge 1999, hereafter referred to as Paper II), shows how a similar approach to the one presented here can be used to extract quantitative robust information from an ensemble of models. In Paper II, no assumption is made on how the ensemble of models is generated, and in particular it need not necessarily be generated with the method presented in this paper.

We compare the new approach to existing methods on the inversion of seismic receiver functions for *S*-wave velocity structure in the crust. This is a complex non-linear waveform fitting problem, where linearized techniques either fail or have a strong dependence on the starting model (Ammon *et al.* 1990). We also perform some detailed analysis of the new method and show how its computation time scales with quantities such as dimension and number of samples. The new class of methods makes use of some simple geometrical concepts which we show can also be used to enhance existing search methods such as genetic algorithms and simulated annealing.

2 ENSEMBLE INVERSION RATHER THAN OPTIMIZATION

It is common to formulate a geophysical inverse problem as one of optimization in a finite-dimensional parameter (model) space (possibly under constraints). Each point in model space represents a set of parameters describing some physical property of the earth, for example seismic wavespeed, and the objective function usually depends on a measure of data misfit and some function of the model. The data misfit measures the discrepancy between observations and theoretical predictions from a model (determined from the solution of a forward problem). This formulation applies to nearly all geophysical inversion. For example, this is the case in the statistical view of inverse theory (e.g. Tarantola & Valette 1982), where one is often interested in locating a model that maximizes the *posterior* probability density function, and also in the 'extremal model' view, which advocates optimizing some property of the model under the constraint that it fits the data to a satisfactory level (e.g. Parker 1977).

In many problems (even linear ones) the data alone do not determine the model uniquely. There may exist no or an infinite number of models that satisfy the data. Optimal models are one way of characterizing the ensemble of acceptable data-fitting models. When a problem is both non-unique and non-linear, it may still be informative to characterize the ensemble of acceptable solutions by seeking models with extreme values in appropriately chosen model properties (Parker 1977). However, there may be cases where we do not know what model properties might be appropriate to optimize, or the data misfit function may be very complex, containing many local minima, thereby making the optimal data fitting model difficult to find and potentially of little use.

An alternative to seeking single optimal models is to characterize the entire ensemble of acceptable solutions directly by first trying to generate as many members of it as possible, and then analysing them. This two stage approach has been adopted many times before; indeed, it was the original purpose of applying Monte Carlo methods in geophysics (Press 1968; Anderssen & Senata 1971). Since then several authors have stated similar views, and proposed methods of appraising the ensemble (Kennett & Nolet 1978; Kennett 1978; Dosso & Oldenburg 1991; Vasco, Johnson & Majer 1993; Lomax & Snieder 1995; Snieder 1998). Here we adopt a similar approach, although we prefer to make inferences based on properties of all models in the ensemble, not just the acceptable ones, because in principle bad data-fitting models may also tell us something useful. The focus of this paper is on the 'sampling' problem.

3 SEARCHING A PARAMETER SPACE

Stochastic (Monte Carlo) search methods such as uniform Monte Carlo search (UMC), simulated annealing (SA) and genetic algorithms (GA) are becoming increasingly popular for optimizing a misfit function in a multidimensional parameter space. The work presented here was motivated by a question that arose in the author's own use of these methods. *'How can a search for new models be best guided by all previous models for which the forward problem has been solved (and hence the data misfit value evaluated) ?'*. It is instructive to consider how each of the three methods above addresses this issue. UMC by definition makes no use of previous samples in parameter space, since each new sample is independent of the previous samples. Both SA and GA make use of previous samples, but in rather complex ways. The information inherited from previous samples may also be highly dependent on control parameters (GA) or temperature profiles (SA), each of which must be tuned for each problem individually.

(1) Construct the 'approximate misfit surface' from the $n_{\rm p}$ previous models for which the forward problem has been solved;

(2) use this approximation (instead of actual forward modelling) together with a chosen search algorithm to generate the next $n_{\rm s}$ samples;

(3) add $n_{\rm s}$ to $n_{\rm p}$ and go back to (1).

IDEALIZED ALGORITHM

In this paper we argue that a simple way of making use of previous model space samples is to use them to approximate (i.e. interpolate) the misfit (objective) function everywhere in model space. Assuming that this can be done, a simple generalized algorithm for searching a parameter space takes the form shown in the box.

We will refer to this as our 'idealized' search algorithm containing one free parameter $n_{\rm s}$ (initially $n_{\rm p} = n_{\rm s}$). The two missing details are how to construct the approximate misfit surface, and how to generate new samples with it. In the next section we present a simple but (we argue) powerful way of constructing an approximate misfit surface, and in the subsequent sections we see how this can be used with existing search methods such as SA and GA, and also as the basis of a new search algorithm.

3.1 The neighbourhood approximation to a misfit function

Given a set of $n_{\rm p}$ samples in model space for which the misfit function has been determined we use the geometrical construct known as the Voronoi diagram (Voronoi 1908) (see Okabe *et al.* 1992; Watson 1992; Sambridge *et al.* 1995, for full details). This is a unique way of dividing the d-dimensional model space into $n_{\rm p}$ regions (convex polyhedra), which we call Voronoi cells. Each cell is simply the nearest neighbour region about one of the previous samples, as measured by a particular distance measure. Here we use the L_2-norm, and so the distance between models $\mathbf{m}_a$ and $\mathbf{m}_b$ is given by,

$$\|(\mathbf{m}_a - \mathbf{m}_b)\| = \left((\mathbf{m}_a - \mathbf{m}_b)^T C_{\rm M}^{-1} (\mathbf{m}_a - \mathbf{m}_b)\right)^{1/2} \qquad (1)$$

where $C_{\rm M}$ is a matrix that non-dimensionalizes the parameter space. (For example, a prior model covariance matrix.) Note that the size and shape of the Voronoi cells will depend on the choice of $C_{\rm M}$. Its role is to put each parameter on an equal footing. A simple choice would be a diagonal matrix with elements $1/s_i^2$. Here s_i can be interpreted as a scale factor for the ith parameter. In this case we can effectively reduce $C_{\rm M}$ to the identity by rescaling each parameter axis by the set of scale factors, $s_i (i = 1, \ldots, d)$. In what follows we will assume that this is the case and drop the $C_{\rm M}$ matrix.

A formal definition of the Voronoi cell follows: Let $P = \{\mathbf{m}_1, \ldots, \mathbf{m}_{n_{\rm p}}\}$ be a set of points in d-space, where $2 \leq n_{\rm p} \leq \infty$, and let $\mathbf{m}_i \neq \mathbf{m}_j$ for $i \neq j$. The Voronoi cell about point $\mathbf{m}_i$ is given by

$$\begin{aligned} V(\mathbf{m}_i) \quad = \quad & \left\{ \mathbf{x} \mid \|\mathbf{x} - \mathbf{m}_i\| \leq \|\mathbf{x} - \mathbf{m}_j\| \right. \\ & \left. \text{for} \quad j \neq i, (i, j = 1, \ldots, n_{\rm p}) \right\} \end{aligned} \qquad (2)$$

Fig. 1 shows a set of Voronoi cells about 10, 100 and 1000 irregularly distributed points in a plane. Since the data misfit function is known at all previous samples, the 'neighbourhood approximation' to the misfit surface (NA-surface) is generated by simply setting the misfit to a constant inside each cell. Therefore to evaluate the approximate misfit at any new point we need only find which of the previous samples it is closest to.

The NA-surface provides a simple way of performing non-smooth interpolation of an irregular distribution of points in d-dimensions and has some very useful properties. For any distribution and density of samples, the Voronoi cells are always unique, space filling and have size inversely proportional to the sampling density (see Fig. 1). This means that the NA-surface will contain short scale variations in misfit only where they are present in the original samples, and longer-scale variations where sampling is sparse. We can regard the NA-surface as a 'minimal'-featured approximation of the misfit surface influenced equally by all available samples. The interesting feature is that the size and shape of the neighbourhoods about each sample (and hence the regions of better and worse data fit) are completely determined by the samples themselves. In particular, no spatial scalelengths or directionality are imposed externally. It is therefore a useful candidate to drive our idealized sampling algorithm above. Note that if the NA-surface can be used to generate new samples (geared towards regions of lower misfit) then these will only change the Voronoi cells locally, and once their true misfit values are generated they will improve the local resolution of the next NA-surface.

Sambridge (1998) first suggested this approximation as a way of sampling a *posterior* probability density function for evaluating Bayesian integrals. Here we use it to approximate any misfit function and show how it leads to a new class of search algorithms for high-dimensional parameter spaces.

3.2 Incorporating the neighbourhood approximation into existing methods

Our idealized search algorithm above may be summarized as 'intermittently replace the real forward modelling calculation with the approximation provided by the NA-surface'. Obviously, then, it is independent of the search algorithm, and it can be incorporated into any existing direct search method. (Note here that we could not use the NA-surface with any gradient based method since all derivatives are effectively zero or infinite.) Here we briefly discuss how the NA-surface may be used with several popular direct search methods, namely genetic algorithms and two forms of simulated annealing. In each case we insert either SA or GA at step (2) of the idealized algorithm described above.

3.2.1 A genetic algorithm

There are many variants of genetic algorithms and these have been described thoroughly in the references cited. At each iteration a new set of $n_{\rm GA}$ samples is generated using stochastic operators. This population will often contain new models and some which are identical to previous models. The use of the NA-surface here is quite trivial. One could simply replace the misfit calculation (requiring forward modelling) with an evaluation of the NA-surface for a fixed number of iterations, say $n_{\rm f}$. After this, one would calculate the real misfit values for the latest population, and use these to update the NA-surface. After $I_{\rm GA}$ iterations one would have solved the forward problem $I_{\rm GA} n_{\rm GA}/(n_{\rm f}+1)$ times, as compared to $I_{\rm GA} n_{\rm GA}$ times if the NA-surface were not used.

Clearly, then, there is a cost saving. The price one pays for this is the risk that the searching ability of the algorithm with the approximate misfit is less effective than for that with the true misfit. The degree to which this inhibits the effectiveness of the GA depends on the nature of the forward problem and probably varies between applications. On the positive side, however, with the NA-surface one could run a GA for many more iterations [a factor of $(n_{\rm f}+1)$ times] for the same overall computation time.

3.2.2 Simulated annealing

A simulated annealing algorithm works by generating samples, $\mathbf{m}$, whose distribution in model space is asymptotically equal to a prescribed sampling density function, $S(\mathbf{m})$,

$$S(\mathbf{m}) = \exp\left(\frac{-\phi(\mathbf{m})}{T}\right) \qquad (3)$$

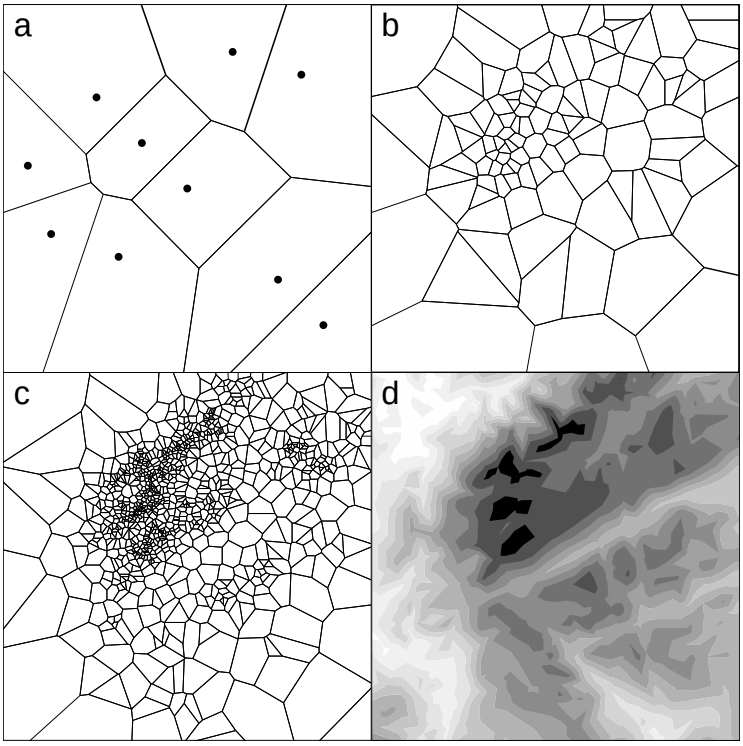

Figure 1. (a) 10 quasi uniform random points and their Voronoi cells. (b) The Voronoi cells about the first 100 samples generated by a Gibbs sampler using the neighbourhood approximation. (c) Similar to b) for 1000 samples. (d) Contours of the test objective function. As samples are generated, the approximation to the multi-modal surface becomes more accurate. Note that all four maxima of the objective function are well sampled but the lower valleys are poorly sampled.

where T is a temperature parameter used to gradually change the sampling density function as the algorithm proceeds. The method is commonly used in a Bayesian inversion (see Tarantola 1987; Duijndam 1988a,b, for a discussion) and for $T = 1$ the function $\phi(\mathbf{m})$ is often chosen so that sampling density, $S(\mathbf{m})$, is equal to the *posterior* probability density (PPD). In principle, $\phi(\mathbf{m})$ can be any data misfit function (for example see Scales *et al.* 1992; Sen & Stoffa 1995). To examine how the NA-surface can be used in this case we need to look at how the samples are generated. Two methods are common: the *Metropolis-Hastings method* (Metropolis *et al.* 1953; Hastings 1970), and the *Gibbs sampler* (Geman & Geman 1984).

3.2.2.1 The Metropolis–Hastings method: Metropolis–Hastings (M–H) performs a Markov-chain random walk in model space (see Gelfand & Smith 1990; Smith 1991; Smith & Roberts 1993; Mosegaard & Tarantola 1995, for details). At each iteration the current model, $\mathbf{m}_A$, is perturbed randomly, usually along one of its d parameter axes, according to a probability distribution, $q(\mathbf{m}_B|\mathbf{m}_A)$. For example, in Fig. 2 the perturbation is from point A to point B. A second (uniform) random deviate, r, is then generated between zero and one, and the walk moves from A to B if

$$r < \min\left[1, \frac{S(\mathbf{x}_B)q(\mathbf{m}_B|\mathbf{m}_A)}{S(\mathbf{x}_A)q(\mathbf{m}_A|\mathbf{m}_B)}\right], \tag{4}$$

where $q(\mathbf{m}_A|\mathbf{m}_B)$ is the probability that a walk at B would be perturbed to A. Usually q is symmetric in its arguments and one obtains the more familiar Metropolis condition on r:

$$r < \min\left[1, \frac{S(\mathbf{x}_B)}{S(\mathbf{x}_A)}\right]. \tag{5}$$

If this condition is not met then the walk stays at A and a new perturbation is made (along the next axis). This accept/reject procedure requires the misfit to be evaluated at the proposed model, B, and so one solution to the forward problem is required for each parameter axis in turn. A single iteration is completed after cycling through all d axes, and therefore requires the forward problem to be solved d times. However, many of the proposed perturbation steps may be rejected, and so the model at the end of the iteration will have only some of its components changed. The walk must proceed for a number of iterations, say $I_{\rm r}$, before the Markov-chain 'relaxes' and the current model becomes statistically independent of the starting model, A. The whole process can then be repeated for the next independent model. It is important to note that it is only the statistically independent samples that are drawn from the desired density distribution $S(\mathbf{m})$, and each one is at a cost of $d \times I_{\rm r}$ solutions to the forward problem. [In practice, the value of $I_{\rm r}$ is controlled by the accept/reject ratio of the chain, which is highly dependent on the character of the misfit function $\phi(\mathbf{m})$ and the value of T: Rothman (1986); Gelfand & Smith (1990); Smith (1991); Smith & Roberts (1993).)

As with a genetic algorithm, the M–H method can be used directly in step (2) of the idealized algorithm. In this case, however, we have a direct statistical interpretation of how it works. At each stage the M–H method can be used to draw $n_{\rm s}$ statistically independent samples from the current NA-misfit surface, that is the NA-surface is used to replace $\phi(\mathbf{m})$ in eq. (3). The important point is that this requires only nearest neighbour calculations and no solutions to the forward problem. If the computation time of forward modelling is represented by $T_{\rm FM}$, and that of a nearest

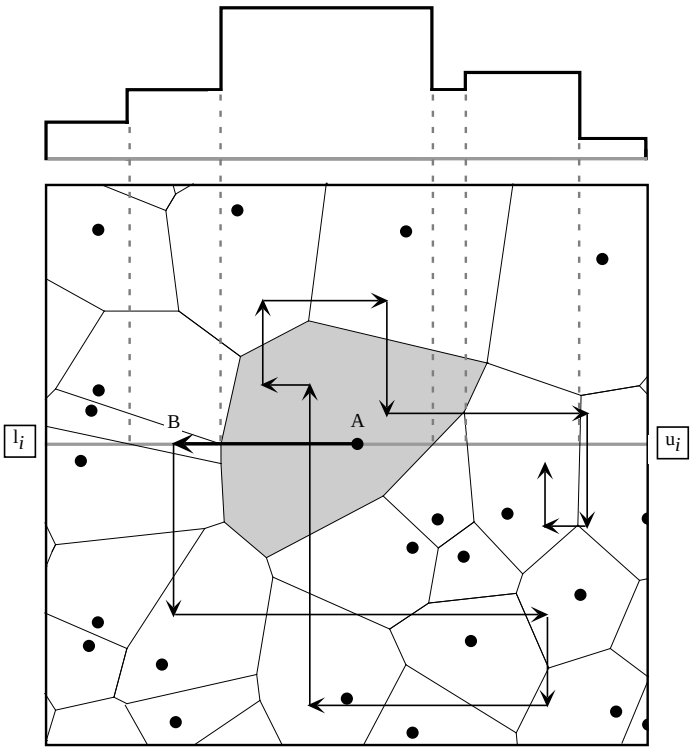

Figure 2. A Markov-Chain Monte Carlo walk using the neighbourhood approximation to the misfit function. The Voronoi cells (polygons) are defined around the set of previous samples (black); the sampling density function is constant inside each cell. The cell in which the walk starts is shaded. The grey line shows the range of values that can be taken in the first step of the random walk which is determined probabilistically according to a conditional PDF (plotted above the figure). Here the walk may leave the cell in which it starts and is attracted to the cells where the data fit function is high (see text).

neighbour search by $T_{\rm NN}$, then the time for generating each independent model with the NA-surface is

$$T_{\rm NA} = T_{\rm FM} + T_{\rm NN} I_{\rm r} d, \tag{6}$$

whereas the equivalent using forward modelling is

$$T_{\rm misfit} = T_{\rm FM} I_{\rm r} d. \tag{7}$$

This leads to a cost ratio of

$$\frac{T_{\rm NA}}{T_{\rm misfit}} = \frac{1}{I_{\rm r} d} + \frac{T_{\rm NN}}{T_{\rm FM}}. \tag{8}$$

In many problems we would expect $I_{\rm r} d \gg 1$ and $T_{\rm FM} \gg T_{\rm NN}$, and hence one could generate many more samples using the NA-surface for same overall cost. By setting $T_{\rm NA} = T_{\rm misfit}$ in the above equations, we find that this factor, F, is given by

$$F = \left(\frac{1}{I_{\rm r} d} + \frac{T_{\rm NN}}{T_{\rm FM}} \right)^{-1}. \tag{9}$$

(Note that, if $T_{\rm NN} = T_{\rm FM}$, then $F \approx 1$ and hence there would be no real benefit from using the NA-surface; one might as well use the true misfit.) As in the case of the GA this improvement in efficiency comes at the cost of intermittently substituting the approximate misfit for the true misfit. Again the usefulness of the NA-surface will depend both on the nature of the forward problem and the sample size, $n_{\rm s}$. As $n_{\rm s}$ is increased the NA-surface is used to generate more samples and so the information contained in the previous models is exploited for longer. Conversely, as $n_{\rm s}$ is decreased the NA-surface is updated more rapidly and less use is made of the global information contained in it.

3.2.2.2 The Gibbs sampler: The Gibbs sampler (also known as a 'heat bath' algorithm) is presented in detail in Geman & Geman (1984) and Rothman (1986). In contrast to M–H this is a one-step method in which a perturbation to a model is generated and always accepted. Fig. 2 gives an example. Again the starting model is at A, but in this case the perturbation to B is produced by drawing a random deviate from the 1-D conditional PDF shown above the figure. This conditional distribution is produced by cutting $S(\mathbf{m})$ along the *x*-axis through A. In Fig. 2, the NA-surface is used instead of the true misfit function, and so the conditional consists of six segments formed from the intersection of the *x*-axis and the Voronoi cells. In each segment the PDF is constant, because the NA-surface is constant inside each Voronoi cell. After cycling through all d axes, a new model is produced, but in contrast to M–H it will have all components changed and be independent of the previous model A.

To implement the Gibbs sampler with the true misfit function one usually approximates the conditional PDF by evaluating it at $n_{\rm a}$ points along the axis, each of which requires a solution to the forward problem (Rothman 1986). Therefore the time for generating each statistically independent sample with the true misfit is

$$T_{\rm misfit} = T_{\rm FM} n_{\rm a} d, \tag{10}$$

whereas with the NA-surface we have

$$T_{\rm NA} = T_{\rm FM} + T_{\rm NN} n_{\rm a} d. \tag{11}$$

An analysis similar to that for the M–H case leads to a cost ratio of

$$\frac{T_{\rm NA}}{T_{\rm misfit}} = \frac{1}{n_{\rm a} d} + \frac{T_{\rm NN}}{T_{\rm FM}}. \tag{12}$$

As can be seen here, the discretization of the axis, $n_{\rm a}$, plays a similar role to the relaxation time, $I_{\rm r}$, in the M–H method. (In numerical experiments with a Gibbs sampler on the receiver function problem discussed in Section 5 we had $d = 24, n_{\rm a} = 20$, and found $T_{\rm NN}/T_{\rm FM} < 1/3000$, indicating that for this case the NA-surface produces considerable savings.) Overall, the NA-surface with the Gibbs sampler results in a similar efficiency to that for the M–H case, and comments and caveats carry over.

Sambridge (1998) presented some results of using the NA-surface in conjunction with the Gibbs sampler to generate samples from successive neighbour approximations to a misfit surface. It was found that the convergence of the Gibbs sampler became excessively slow as the dimension increased. However, it was incorrectly concluded that the 'NA-Gibbs sampler' would be impractical for dimensions greater than 10. Later work showed that the slow convergence was due to the lack of structure in the PDF used in that example.

We see, then, that the neighbourhood approximation can easily be used in combination with any existing direct search method; however, its success will depend on the details of the application. Fig. 1 shows a simple example of searching a 2-D parameter space using the NA-Gibbs sampler for a test function with $T = 1/15$ and

$$\phi(\mathbf{m}) = \frac{f_{\rm max} - f(\mathbf{m})}{f_{\rm max} - f_{\rm min}}, \qquad (13)$$

where the test function range is $f_{\rm max} = 2200$ and $f_{\rm min} = 500$. 10 independent samples are generated at each iteration ($n_{\rm s} = 10, n_{\rm a} = 20$). The initial 10 samples are distributed uniformly, and Voronoi cells are plotted for 100 and 1000 samples. As the algorithm proceeds, the information in the NA-surface is exploited to concentrate sampling in the regions where $S(\mathbf{m})$ is high [shown in part (d)]. Note that all four of the local maxima (darkest regions) are densely sampled, while the troughs in $S(\mathbf{m})$ (lighter) are sparsely sampled. An example of the NA-Gibbs sampler in a 24-dimensional space is presented in Paper II, although there it is applied to the appraisal rather than the search stage of the inverse problem.

(1) Generate an initial set of $n_{\rm s}$ models uniformly (or otherwise) in parameter space;
(2) Calculate the misfit function for the most recently generated set of $n_{\rm s}$ models and determine the $n_{\rm r}$ models with the lowest misfit of all models generated so far;
(3) Generate $n_{\rm s}$ new models by performing a uniform random walk in the Voronoi cell of each of the $n_{\rm r}$ chosen models (i.e. $n_{\rm s}/n_{\rm r}$ samples in each cell);
(4) Go to step 2.

A NEIGHBOURHOOD ALGORITHM

4 A NEIGHBOURHOOD SAMPLING ALGORITHM

The previous section showed how the neighbourhood approximation can be used to enhance any existing direct search method. It was only used, however, to replace the forward problem. In this section we present a new direct search method which uses the spatial properties of Voronoi cells to directly guide the sampling of parameter space.

The algorithm is conceptually simple and summarized in Fig. 3. The key idea is to generate new samples by resampling chosen Voronoi cells with a locally uniform density. The algorithm can be summarized in the four steps shown in the box.

The uniform walk within a chosen Voronoi cell can be generated using a Gibbs sampler, in a manner similar to that described above. Fig. 3 shows an example. At each step the ith component of the current model, $\mathbf{x}_A$, is replaced with a uniform random perturbation restricted to the boundaries of the current Voronoi cell (l_i, u_i). This is essentially the same as in the previous case (shown in Fig. 2), except that the conditional PDF is set to zero outside the cell. The resulting random walk asymptotically generates a spatially uniform distribution of samples within any d-dimensional convex polygon. The philosophy behind the algorithm is that the misfit of each of the previous models is representative of the region of space in its neighbourhood (defined by its Voronoi cell). Therefore at each iteration new samples are concentrated in the neighbourhoods surrounding the better data-fitting models. In this way the algorithm exploits the information contained in the previous models to adapt the sampling.

There are two important features of the new algorithm. The first is that the size and shape of the neighbourhoods are not imposed externally but rather determined automatically and uniquely by the previous samples. Note that the boundaries of each Voronoi cell are determined by all previous samples, and, regardless of how irregularly distributed the samples are, the neighbourhood will be a 'reasonable' one, in the sense of an L_2-norm. The second feature is that the algorithm only requires models to be assessed for their relative fit to the data, because it uses only the rank of a misfit/objective function. This is very useful because in many problems (e.g. seismic waveform inversion) it is often much easier to answer the question 'is model A a better fit to the data than model B ?' than to quantify the difference in a precise way. With most search methods one is forced to define an absolute measure of data misfit (e.g. the chi-square or the PPD), which requires accurate knowledge of the statistics of the data errors. With a rank based approach, the weight of each of the previous samples in driving the search depends only on their position in the rank list and not on any analytical estimate of the noise statistics. Of course noise fluctuations still play a role because, when large enough, they can influence the ranks of the misfit values. Ultimately, an absolute measure of data fit is always needed in order to determine if any models satisfy the data, however, by using only misfit rank, the search process becomes independent of the absolute scale of the misfit function.

With direct search methods that do not use rank, one is often forced to rescale the misfit function to avoid stability problems or loss of efficiency. For example, misfit rescaling is often used with genetic algorithms to reduce 'exhaustion' problems (Goldberg 1989). [We note that misfit rank has also been introduced into GAs to deal with this problem (Goldberg & Deb 1991; Sambridge & Gallagher 1993).] Similarly, the main role of the temperature parameter (crucial to the performance of the simulated annealing algorithm) is to rescale the misfit function (*cf.* eq. 3), thereby controlling the influence of large changes in misfit on the search process. A rank based approach can be applied to any misfit criterion that can discern between competing models (even one based on a combination of criteria or a set of heuristic rules).

In defining the neighbourhood algorithm a second control parameter, $n_{\rm r}$, has been introduced. Again the influence of this is quite straightforward. For larger values, the sampling (at each iteration) is spread over more cells, and so we would expect the algorithm to be more exploratory in nature. Conversely, for smaller values it is restricted to fewer cells and so the sampling should be more local-

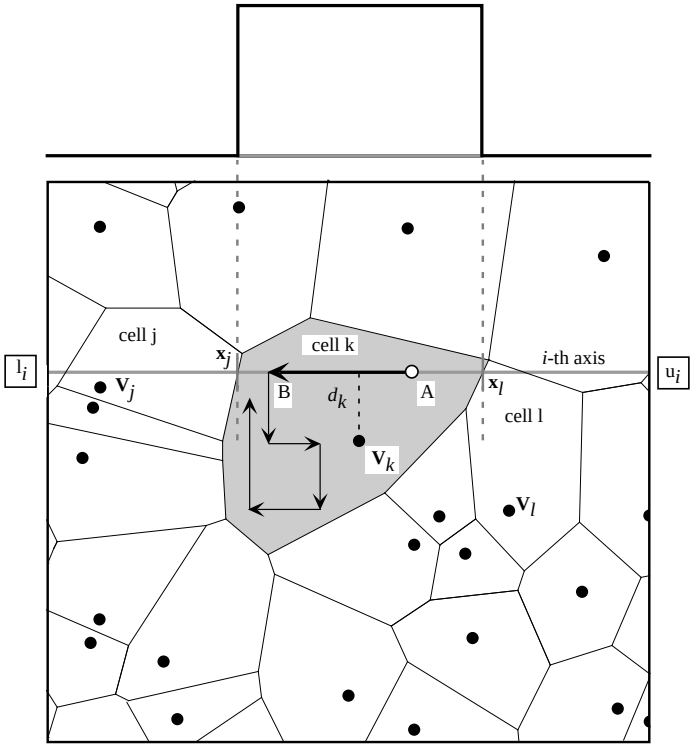

Figure 3. A uniform random walk restricted to a chosen Voronoi cell. A Gibbs sampler is used in a similar manner to Fig. 2; however, the conditional PDF outside the cell is set to zero (plotted above). Asymptotically the samples produced by this walk will be uniformly distributed inside the cell regardless of its shape.

ized. In Section 5, we examine the influence of this parameter in more detail.

4.1 Sampling Voronoi cells in a high-dimensional space

It turns out that, in order to implement the neighbourhood algorithm, full details of the high-dimensional Voronoi diagram do not have to be determined (which would be an impossible task). As can be seen from Fig. 3, all that is required is to find the points where the boundaries of the d-dimensional Voronoi cell intersect the ith axis passing through a given point $\mathbf{x}_A$. The next step of the uniform random walk is then restricted to lie between these two points on the axis (i.e. x_j and x_l in Fig. 3).

4.1.1 A brute force approach

The intersections can be approximated using a simple approach. We discretize the axis into n_a points and for each find its nearest neighbour amongst the previous n_p samples. The approximate position of the intersection is given by a change in the nearest neighbour node. If we calculate the distance to each of the n_p points, then the time for finding a single nearest neighbour satisfies

$$T_\mathrm{NN} \propto n_\mathrm{p} d \tag{14}$$

By using eq. (11), we see that the time for generating each independent sample becomes

$$T_\mathrm{NA} = T_\mathrm{FM} + \lambda_1 n_\mathrm{a} n_\mathrm{p} d^2 \tag{15}$$

where λ_1 is a constant of proportionality. In practice this is very easy to implement, and, with modern vector computation, may be practical for many applications. However, since the 'intersection problem' is at the very heart of the algorithm (and will need to be solved many times), an efficient solution is essential. Note that the ratio of the two terms in eq. (15) is

$$T_\mathrm{R} = \frac{\lambda_1 n_\mathrm{a} n_\mathrm{p} d^2}{T_\mathrm{FM}}. \tag{16}$$

Ideally we seek $T_\mathrm{R} < 1$, so that the overall cost of the algorithm is dominated by the time required to solve the forward problem. Even if this is not the case however, the algorithm is still viable. Note also that, since the cost of generating a single sample depends on all n_p previous samples (eq. 15), the cost for generating all samples depends quadratically on n_p.

4.1.2 A refined approach

It turns out that this simple 'brute force' approach can be improved upon considerably by using a more complex method which avoids a discretization of the axis and also allows exact intersections to be calculated. If we define the kth Voronoi cell as the one about sample $\mathbf{v}_k$, and the point where the boundary between cell k and j intersects the axis as $\mathbf{x}_j$ (see Fig. 3), then by definition we have,

$$\|(\mathbf{v}_k - \mathbf{x}_j)\| = \|(\mathbf{v}_j - \mathbf{x}_j)\| \tag{17}$$

Taking $C_\mathrm{M} = I$ we have,

$$d_k^2 + (v_{k,i} - x_{j,i})^2 = d_j^2 + (v_{j,i} - x_{j,i})^2 \tag{18}$$

where d_k is the perpendicular distance of sample k from the current axis, and a subscript of i denotes the ith component of the corresponding vector. Solving for the intersection point we obtain

$$x_{j,i} = \frac{1}{2}\left[v_{k,i} + v_{j,i} + \frac{(d_k^2 - d_j^2)}{(v_{k,i} - v_{j,i})}\right] \quad (19)$$

If we ignore the calculation of d_k^2 and d_j^2 for the moment then this expression requires just six arithmetic operations to evaluate. To find the required boundaries of the Voronoi cell, eq. (19) must be evaluated for all $n_{\rm p}$ cells and the two closest points either side of $\mathbf{x}_A$ retained. More formally, we have the lower boundary given by

$$\max\left[l_i, x_{j,i}\right], (for\ x_{j,i} \leq x_{A,i}; j = 1, \ldots, n_{\rm p}), \quad (20)$$

and the upper boundary by

$$\min\left[u_i, x_{j,i}\right], (for\ x_{j,i} \geq x_{A,i}; j = 1, \ldots, n_{\rm p}), \quad (21)$$

where l_i and u_i are the lower and upper bounds of the parameter space in the ith dimension, respectively. After cycling over all dimensions a new sample in parameter space is generated, and the time taken for each satisfies

$$T_{\rm NA} = T_{\rm FM} + \lambda_2 n_{\rm p} d \quad (22)$$

which gives a cost ratio

$$T_{\rm R} = \frac{\lambda_2 n_{\rm p} d}{T_{\rm FM}} \quad (23)$$

which in practice will be a considerable improvement over the brute force approach since a factor of $n_{\rm a} d$ has been removed (*cf.* eq. 16).

To complete the description we need the set of squared perpendicular distances, $d_j^2 (j = 1, \ldots, n_{\rm p})$, available at each step of the walk. These can be calculated for the initial axis [an $O(dn_{\rm p})$ calculation], and for each new axis by a recursive update procedure. For example, after the ith step has been completed and the walk moves from $\mathbf{x}_A$ to $\mathbf{x}_B$, the current set of d_j^2 values can be calculated for the $(i+1)$th axis using

$$\left(d_j^2\right)_{i+1} = \left(d_j^2\right)_i + \left(v_{j,i} - x_{B,i}\right)^2 - \left(v_{j,i+1} - x_{B,i+1}\right)^2 \quad \text{for } (j = 1, \ldots, n_{\rm p}). \quad (24)$$

The cost of this procedure for each model also depends linearly on $n_{\rm p} d$, and so only the multiplicative constant in (22) is changed, and overall the time take will be linear in both the number of previous points, $n_{\rm p}$, and dimension, d. This completes the description of the method. In the next section we present a numerical example of its application to the problem of inverting seismic receiver functions for crustal seismic structure. In this case we found that the total time taken for solving the forward problem for all models was a factor of 14 times greater than the total time of generating all of the samples, (i.e. $T_{\rm R} = 1/14$).

5 EXAMPLES

To illustrate the neighbourhood algorithm we examine its performance in the inversion of receiver functions for crustal seismic structure. This is a highly non-linear waveform fitting problem, exemplifying many of the difficulties of seismogram inversion. Typically, receiver functions contain both high-amplitude reverberation phases, due to near surface structure, and much smaller amplitude converted phases from the Moho. It is well known that when linearized techniques are applied to receiver function inversion they can have a strong dependence on the assumed starting model (Ammon *et al.*, 1990). It is therefore a suitable test problem on which to illustrate the algorithm.

We use a synthetic data problem so that the true model is known in advance and can be compared with the results of the algorithm. To be as realistic as possible we use exactly the same parametrization of crustal structure as was used by Shibutani *et al.* (1996). In that study, a genetic algorithm was applied to the inversion of receiver functions recorded in eastern Australia. We also apply this genetic algorithm to our synthetic receiver functions and compare performance with that of the neighbourhood algorithm.

5.1 Parameterization and synthetic receiver functions

The crustal structure is divided into 6 horizontal layers, which we name 'Sediment', 'Basement', 'upper', 'middle' and 'lower Crust', and 'Mantle'. The model comprises of four parameters in each layer: the thickness of the layer (km), *S*-velocity at topmost point in the layer (km s^{-1}), *S*-velocity at bottommost point in the layer (km s^{-1}), and ratio of *P* to *S* velocity in the layer. A linear gradient in velocity is assumed in each layer. The philosophy behind the parametrization is to be as flexible as possible, allowing a very wide range of earth models to be represented by a finite-dimensional ($d = 24$) parameter space. Therefore very loose prior bounds are placed on each parameter. These are apparent in the figures below and also act as the scale factors required to non-dimensionalize the parameter space. (Numerical values and all other details can be found in Shibutani *et al.*1996)

A synthetic receiver function (RF) was calculated from the true model using the Thomson–Haskell matrix method (Thomson 1950; Haskell 1953), and uniform random noise was added in the frequency domain (with a noise-to-signal ratio of 0.25). This resulted in the 'observed' RF shown in Figs. 5c, d etc (blue line). A chi-square misfit function was used to measure the discrepancy between the true, $\phi^{\rm obs}(\mathbf{m})$, and predicted, $\phi^{\rm pre}(\mathbf{m})$, waveforms from an earth model $\mathbf{m}$:

$$\chi^2_\nu(\mathbf{m}) = \frac{1}{\nu}\sum_{i=1}^{N_{\rm d}}\left(\frac{\phi_i^{\rm obs} - \phi_i^{\rm pre}}{\sigma_i}\right)^2 \quad (25)$$

where σ_i is an estimate of the standard deviation of the noise calculated from $\phi^{\rm obs}$ using the approach described by Gouveia & Scales (1998), and ν is the number of degrees of freedom ($\nu \approx N_{\rm d} - d$). (Note that all off-diagonal terms of the data covariance matrix are ignored, and so this misfit measure does not take into account any temporal correlation in the noise.) The trace was sampled at a frequency of 25 Hz with 30 s duration, giving a total of, $N_{\rm d} = 876$, data points.

5.2 Searching for data-fitting models

Fig. 4 shows the improvement in data fit (of the best model) in three trials with a neighbourhood and a 'well tuned' genetic algorithm starting from various initial (random) populations. (The details of the genetic algorithm are exactly those of Shibutani *et al.*1996). At each iteration the neighbourhood algorithm generates 10 samples of a uniform random walk inside each of the Voronoi cells of the current 2 best models (i.e. $n_{\rm s} = 20, n_{\rm r} = 2$). These values were selected after a few trials with $n_{\rm s}$ in the range 2–100 and $n_{\rm r}$ in the range 1–10. No exhaustive testing was done and we do not expect the values to be in any way optimal. The tuning of the genetic algorithm was accomplished with more care (and not by the author) (Shibutani personal communication).

Our primary interest in Fig. 4 is to compare the characters of the misfit reduction in the two cases, rather than the particular values of data misfit achieved. (The latter will no doubt vary between applications and with the precise details of the algorithms.) Notice

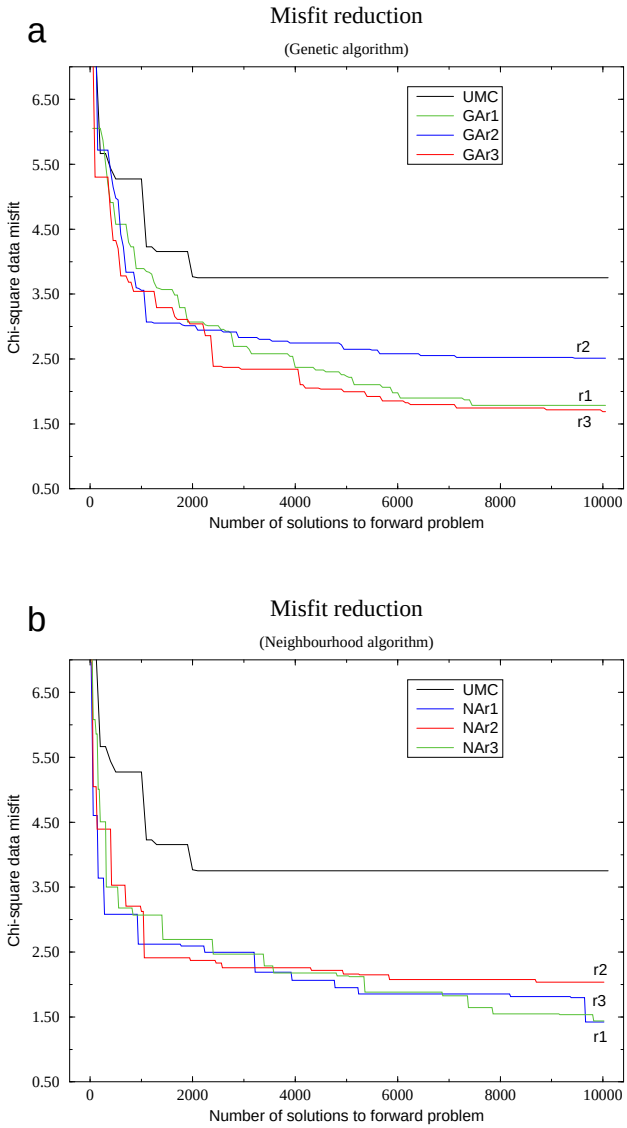

Figure 4. (a) χ^2 data misfit function against number of models (produced by a genetic algorithm) for which the forward problem has been solved. The three curves are produced by different starting random number seeds. (b) As (a) for three runs of the neighbourhood algorithm. In both panels the corresponding curve from a uniform Monte Carlo search is plotted for reference (black). Notice how the GA curves consist of a large number of small changes while the NA curves look more like the early stages of the uniform Monte Carlo search. After 10 000 models have been sampled, two of the three NA curves have lower data misfits than the best GA curve.

how the GA curves in Fig. 4(a) consist of a large number of small changes and a few larger steps which gradually diminish. In contrast, the three NA curves (Fig. 4b) all have a 'staircase'-like feature reminiscent of the early stages of a uniform Monte Carlo search (shown in black); however, unlike the case for uniform sampling, the steps continue as the iterations proceed, and result in reasonably well fit best models (χ^2_ν values of 1.42, 2.04 and 1.44). After 10 000 models have been generated, the uniform sampling gives a χ^2_ν value of 3.75 (a poor data fit), and two of the three NA curves have lower misfits than the best GA curve (χ^2_ν values of 1.79, 2.51 and 1.69), although we do not feel that this is particularly significant.

The character of the misfit reduction is more intriguing. In the GA case the small changes suggest that a 'local search' mechanism is present, (consistent with previous speculation on GAs; Sambridge & Drijkoningen 1992). The absence of these small changes in the NA curves suggests that it exhibits no local search character; rather, it is performing uniform searches in ever-decreasing volumes of parameter space. The important point is that the size and shape of these sampling regions (Voronoi cells) are completely determined by the previous samples, and also evolve as the algorithm proceeds.

5.3 Self adaptive sampling in parameter space

Examining the reduction in misfit of the best model gives only limited information on the character of the search process. Figs 5 and 6 show the distribution of *S*-wave velocity models for the two better runs in the GA and NA, together with the receiver functions of the best model found in each case. The most striking feature of these

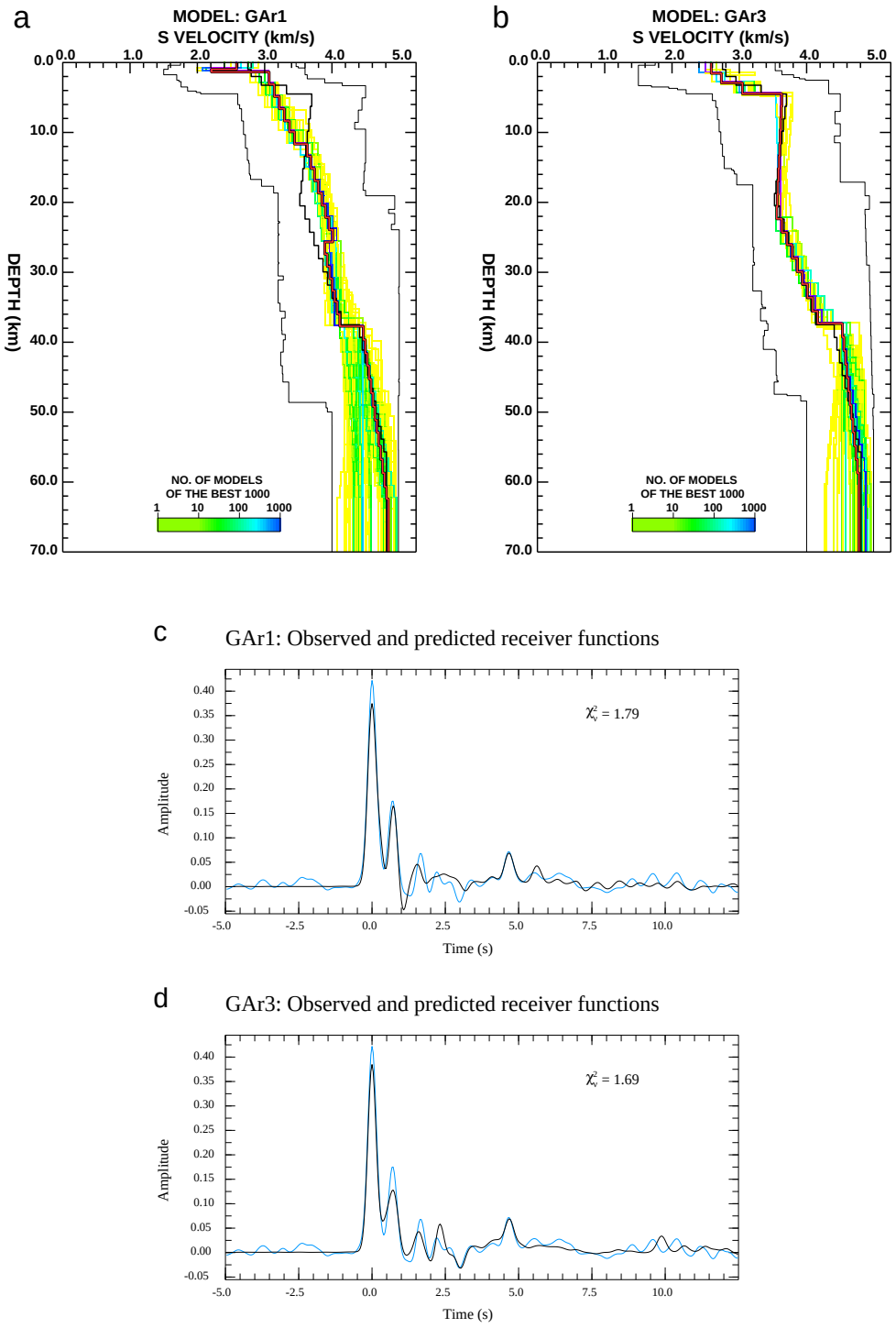

Figure 5. (a) Density plot of the best 1000 data-fitting *S*-velocity models generated by the GA with the first random seed (GAr1). The best data-fitting model is plotted in red, and the true model is in black. The colour scale shows the increase in data-fit, from green to blue. The outline represents the extremes of all 10 000 models generated. (b) As a), for run GAr3. (c) and (d) Receiver functions of the true (blue) and best fit (black) models for GAr1 and GAr3, respectively. In both cases the density of the models is highly irregularly distributed, showing concentrations and 'holes' about the best model.

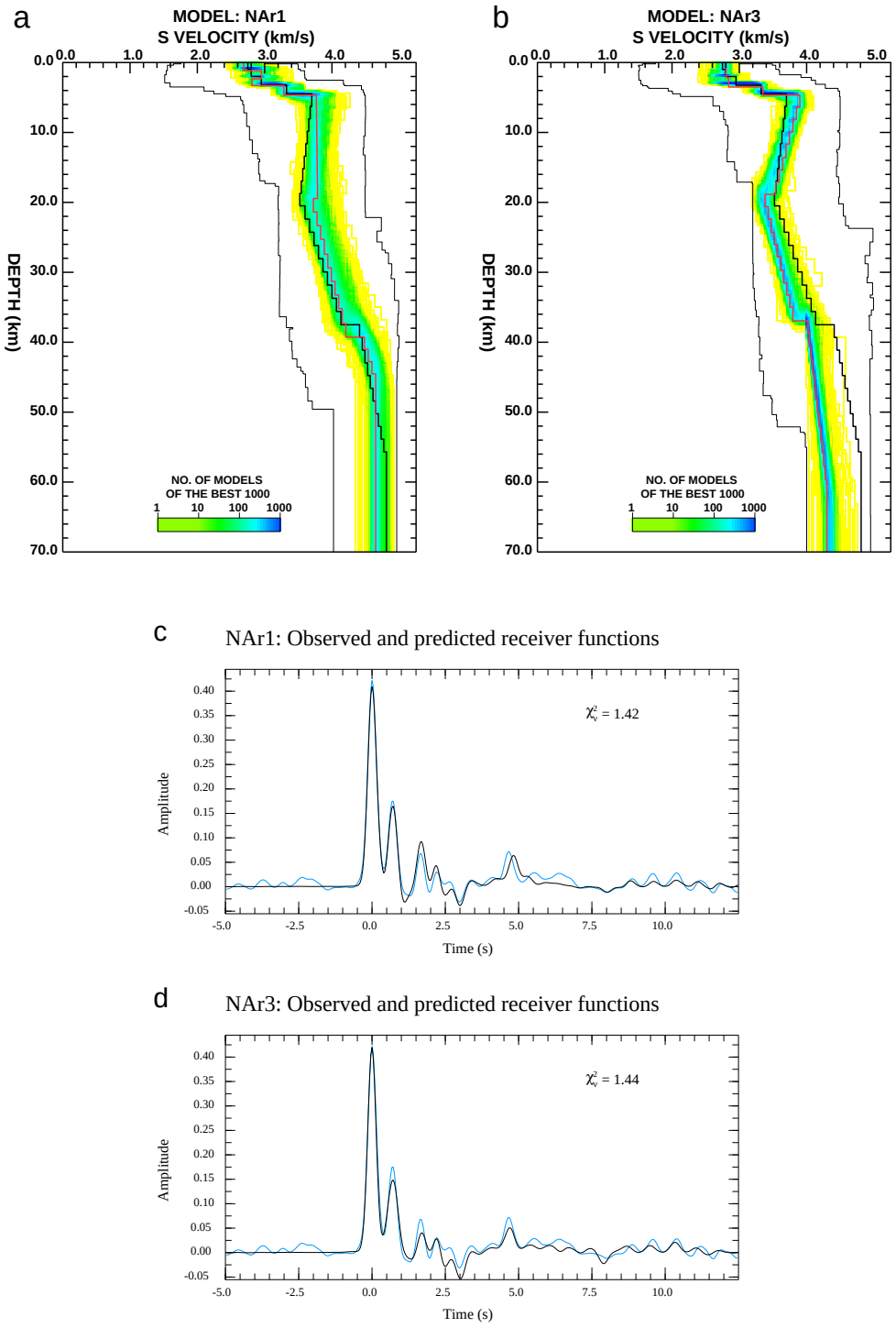

Figure 6. (a) Model density plot for models produced by the neighbourhood algorithm with the first random seed (NAr1) and parameters $n_{\rm s} = 20, n_{\rm r} = 2$. Details are the same as for Fig. 5. (b) As (a), for run NAr3. (c) and (d) Receiver functions of the true (blue) and best fit (black) NAr1 and NAr3, respectively.

figures is the difference in the sampling density of the best 1000 models between the two GA and NA results. In the GA case, both density plots (Figs 5a and b) show a very localized sampling with high concentration in some places and 'holes' in others. This implies localized irregular sampling in parameter space. The density of the 1000 NA models is much smoother in both cases. Again this is more reminiscent of uniform sampling; however, the search has achieved comparable levels of misfit reduction to the GA search. The ranges of models generated (shown as a black outline) are effectively at the boundaries of the parameter space, and are similar in all cases.

A comparison of the best fit model in each case (red line) with the true model (black line) shows that for both GA and NA the depth and velocity jump across the Moho are recovered reasonably

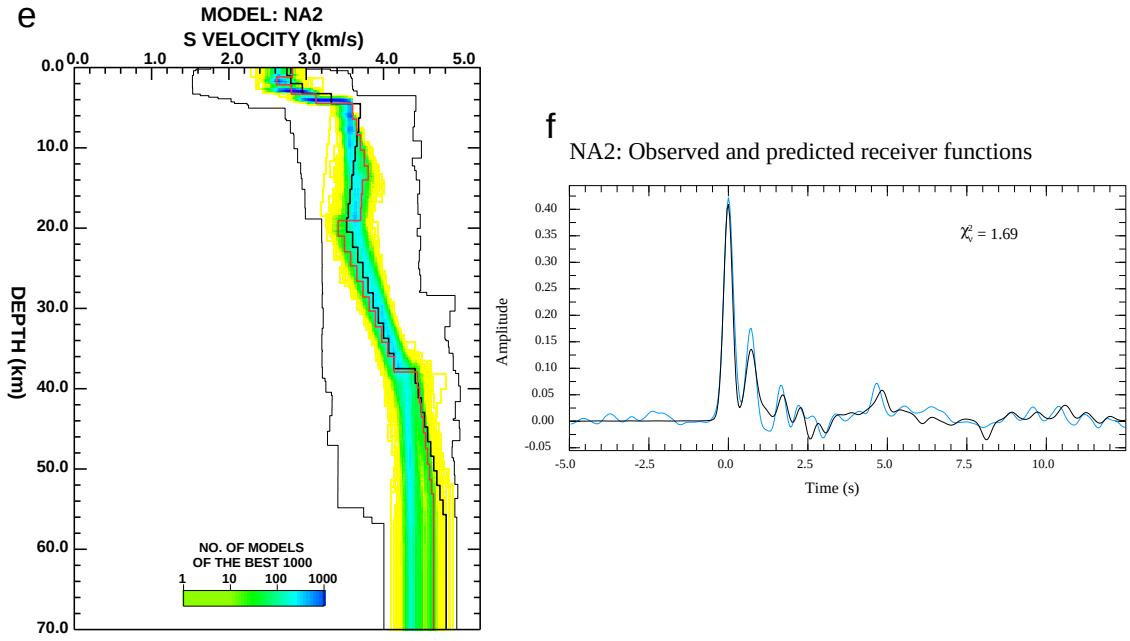

Figure 6. (continued) (e) and (f) Show the results using control parameters $n_s = 2, n_r = 1$. Note that the final data fit is exactly the same as for GAr3.

well. This results in the *P*-to-*S* conversion at 5 s being fit quite well in all cases, although the NA model in Fig. 6(a) has a slightly deeper Moho resulting in a slightly later converted phase (see Fig. 6c). The NA models seem to recover the depth and jump across the basement better than the GA, resulting in better alignment of dominant phases in the reverberations (0–3 s). This may account for the lower overall data fit of the GA models.

Figs 6(e) and 6(f) show results from a separate NA run with parameters $n_s = 2, n_r = 1$. This is included because it (fortuitously) provides an example where the best fit model found has exactly the same numerical value as one of the GA results (see GAr3 in Figs 5b and d). This is useful because it allows a direct comparison of sampling between the two algorithms without the complication of differences in sampling leading to differences in the fit of the best model. The NA sampling is again much more distributed and uniform than the GA case, which confirms the previous conclusion that the more distributed sampling of the NA has not been at the sacrifice of improving the fit of the best model. A comparison of the two best fit models shows that the NA model (Fig. 6e) is similar at the Moho but distinctly different from GAr3 in the reverberation layers near the surface (giving a near perfect fit to the true receiver function in the high amplitude (0–1 s) part of the trace). This also suggests the presence of multiple minima in the misfit surface (as one might expect in a waveform fitting problem).

One might argue that the apparently sparse sampling of the NA is a disadvantage because it will not concentrate as much sampling in the region of parameter space about the true model, compared with a GA perhaps. To examine this question we plot the entire ensemble of models produced by NAr1 and GAr1 projected on to four pairs of parameter axes (see Fig. 7). Immediately one sees the effect of the discretized parameter space used by the GA (left panels). All GA samples fall on a relatively crude grid. (Note that the level of discretization used by the GA is typical of problems of this kind, and had to be limited in order to stop the GA string becoming too long; Shibutani *et al.*1996). The holes in the GA ensemble are also apparent, even close to the best fit model (cross). It is well known that the GA can produce an ensemble containing models that are partial or full copies of each other, which will mean that many models plot on top of each other in the left panel. The NA works with a continuum and produces a diverse cloud of samples with sampling concentrated in the neighbourhood of the best fit model (which, in this case, is on average closer to the true model, marked by an $\times$).

It is interesting to note that even though each cloud in Fig. 7 is a projection from 24 to two dimensions, there is a fair amount of consistency between the data fit of the model and its position relative to the centre of each cloud (colour scale indicates data fit). Furthermore, the sampling density seems to increase closer to the best fit model, indicating that the algorithm has adapted the sampling to concentrate in regions of better data fit. We argue here that the diversity of the NA ensemble is likely to be more useful in the appraisal stage of the inverse problem than that produced by the GA, since it might better characterize the region of acceptable models. (This issue is dealt with in detail in Paper II.)

Fig. 8 shows a similar scatter plot where all 10 000 models have been plotted as a function of Moho depth and velocity jump across the Moho. (These variables are linear combinations of the original model parameters.) One can see that the true values (crosses) are reproduced well by the best fit model and the NA ensemble provides thorough sampling about the true values. We conclude from this that the data contain reasonable information on both the depth and jump across the Moho.

5.4 Properties of NA sampling

In these examples only a small number of Voronoi cells were resampled at each iteration ($n_r = 1$ or 2). Since a small value of n_r suggests a more restrictive search (i.e. more exploitation than exploration), one might suspect that this would severely inhibit the sampling of the NA, perhaps directing it away from a global minimum towards secondary minima. Fig. 9 was produced in an attempt to examine this question, and, it turns out, allows us to observe a re-

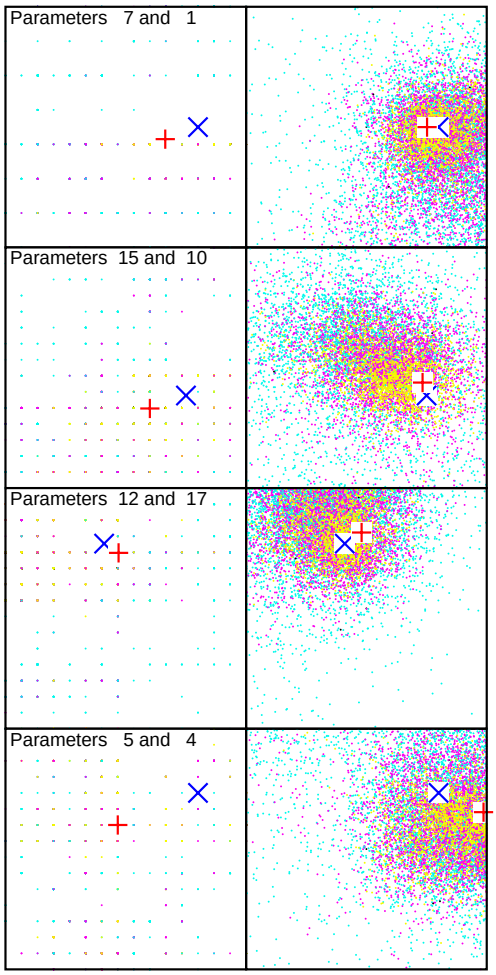

Figure 7. Comparison of ensembles produced by NA (right panels) and GA (left panels) projected onto four pairs of parameter axes (labelled). The best of the 10 000 models in each panel is shown as a red cross and the true model as a blue $\times$. The dots are coloured from blue through red to yellow, as the data fit increases. The GA left panel produces many copies of each model on a crudely discretized axis and yet still has 'holes'. In each case the NA produces a continuum of samples with a higher concentration of better fitting models in the region of the true model.

markable property of the NA sampling. Here we plot, as a function of iteration, the average distance in model space between the current best fit model and the set of models most recently generated inside its Voronoi cell. For the NA this is an average over $n_{\rm s}/n_{\rm r}$ models, and gives a measure of the size of the Voronoi cell about the current best fit model. We also plot the same measure for the GA and MC results for reference (although in these cases the curve has no simple interpretation).

As the iterations proceed we see (from Fig. 9) that the size of the Voronoi cell about the best fit model gradually decreases and then increases in a spike-like feature only to decay again. This pattern is repeated several times. The reason for this becomes clear when we compare the curve to the corresponding misfit reduction curve NAr1 in Fig. 4(b) (This curve is shown in Fig. 9 in light grey). The spikes in Fig. 9 occur at exactly the same points as where a new best fit model is found. Therefore the gradual shrinking of the Voronoi cell is a result of more and more samples being generated within it. (We recall that at each iteration the Voronoi cells are updated, and so they must shrink as space is taken up by new cells.) Each spike in Fig. 9 occurs when the sampling switches to a new Voronoi cell about a new best fit model, and this cell is much larger than the previous one.

This illustrates the remarkable property of the NA referred to above; that is, the local sampling density (inverse of Voronoi size) automatically increases and decreases (in a pulsating-like fashion) as new cells are sampled. Furthermore, the centres of the sampling must also change, because they are restricted to a new set of $n_{\rm r}$ Voronoi cells whose centres differ from the old ones. Note that each new best fit model is not necessarily produced by sampling the previous best fit Voronoi cell, but may have come from any of the $n_{\rm r}$ cells. Hence the centres of sampling may jump from place to place. Note also that as the best model is replaced with a better one, the old cell is not necessarily discarded, since it merely moves down the rank list. We see then that the NA is always sampling the 'currently most promising' $n_{\rm r}$ regions in model space simultaneously.

A further question arises from these plots. How can each new Voronoi cell in parameter space be larger than the original cell, if its defining point in parameter space is inside the original cell ? The answer can be illustrated with a simple 2-D test case (see Fig. 10). Fig. 10(a) shows a Voronoi cell (shaded) and the set of new samples generated within it (open circles). After updating the Voronoi

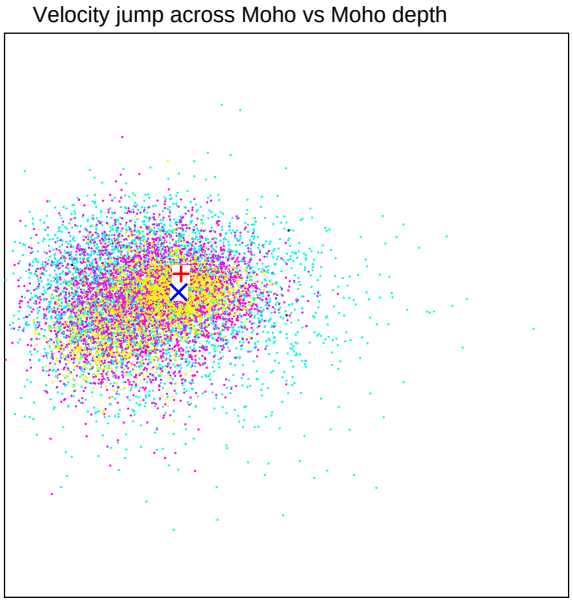

Figure 8. 10 000 models produced by the NA projected onto the axes representing the *S*-velocity jump across (*x*-axis) and depth of the Moho (*y*-axis). Symbols and colour scale as in Fig. 7. The plot range is the complete parameter space (11 to 60 km for Moho depth, -0.5 to 1.8km^{-1} for velocity jump). The best fit and true models are relatively close. The density of the samples is increased in the region around the true solution where data fit is high (yellow shades).

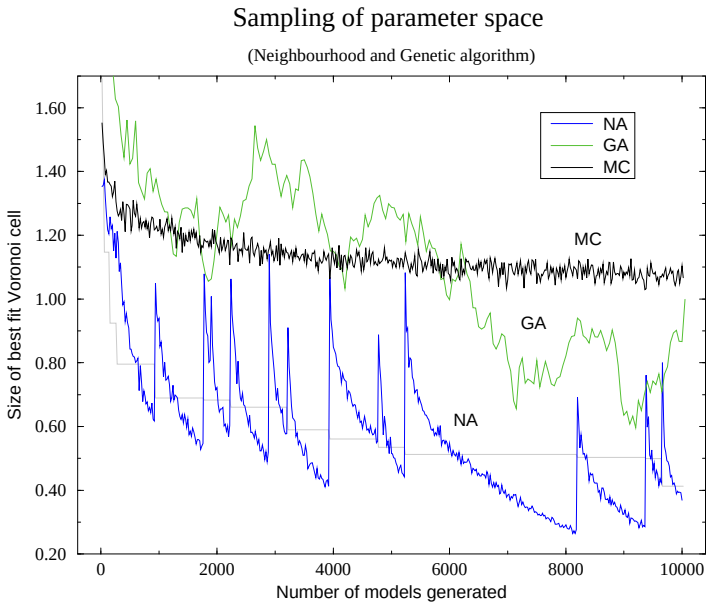

Figure 9. Three curves showing how the distance between the current best data-fitting model and the centroid of the current population varies during the search. For the NA (blue) this is a measure of the size of the Voronoi cell about the best fit model. The spikes in the NA curve coincide with the points where a new best fit model is found (labelled NAr1 in Fig. 4b and shown in grey here). The spike like character demonstrates that the NA has the ability to adapt both the sampling density and the centre of the sampling as better data-fitting models are found.

cells we see that all six of the new Voronoi cells extend beyond the original (shaded) cell. Therefore, even though the sample itself lies inside the original cell, its Voronoi cell can extend beyond the original Voronoi cell. It turns out that the likelihood of this occurring dramatically increases with dimension, and also the new cell is more likely to occupy a larger region of parameter space. (This effect is discussed in more detail by Sambridge 1998.)

5.5 Tuning the free parameters

As mentioned above, the two free parameters ($n_{\rm s}, n_{\rm r}$) used in the examples were obtained after trials over a range of values. Robust conclusions are not possible from these limited tests, and no doubt details will vary between applications. From the design of the algorithm itself, however, it is possible to make some general statements about how we expect the two parameters to influence the algorithm. For example, as the number of resampled Voronoi cells

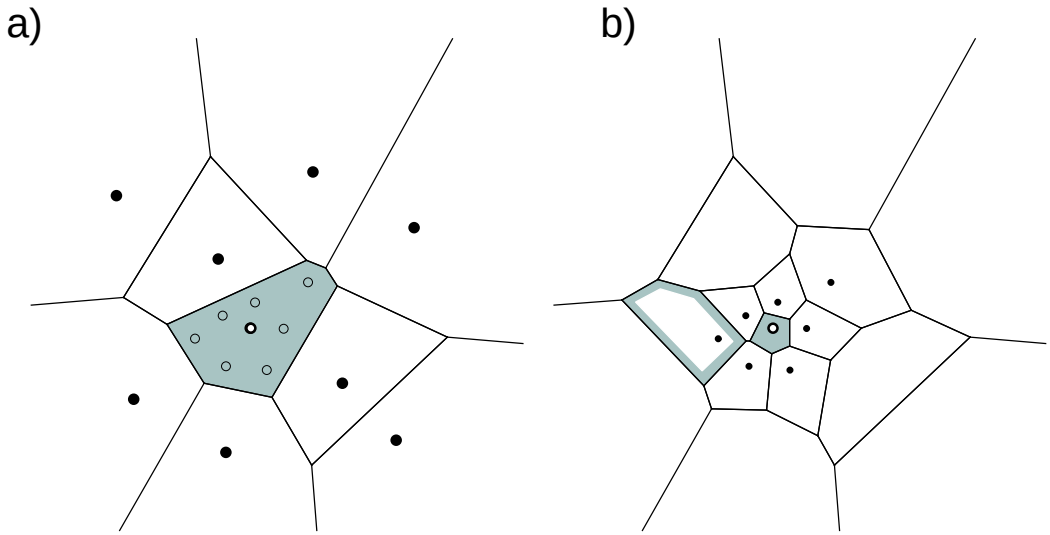

Figure 10. An example of how the NA can increase, decrease and shift the centre of its sampling density in parameter space. (a) Nine points and their Voronoi cells. (b) After adding seven new points to the shaded Voronoi cell in (a) it has shrunk and the local sampling density has increased. If the new model to the left becomes the best data-fit sample then at the next iteration new samples will be generated in its cell (outline shaded). The average density will then decrease and the focus will shift. This type of effect gives rise to the 'spike-slope' features seen in Fig. 9.

$n_{\rm r}$, increases, we expect the algorithm to be more explorative of parameter space and less exploitative (i.e. less local). Therefore as $n_{\rm r}$ decreases we expect it more likely to be trapped in local minima, and as $n_{\rm r}$ increases, less likely (but also less efficient at finding better data-fitting models). The influence of $n_{\rm s}$ is more difficult to assess. What we know is that as $n_{\rm s}$ increases more weight is given to the previous samples because the Voronoi cells are updated less frequently. As both $n_{\rm s}$ and $n_{\rm r}$ are increased together we would expect the overall algorithm to become more explorative and so more robust as a sampler, but less 'efficient' as an optimizer.

The influence of both parameters will be affected by the dimension of the parameter space. As shown by Sambridge (1998), there is a critical number of points in a parameter space of dimension d, above which all Voronoi cells are no longer neighbours of all other cells. If the total number of samples is higher than this value then the Voronoi cells become 'isolated'. A small value of $n_{\rm r}$ will mean that after a few iterations each random walk loses the ability to sample all regions in space, and so the algorithm effectively consists of a set of $n_{\rm s}$ simultaneous local searches. Note that these are not independent random walks because they are restricted to the best $n_{\rm s}/n_{\rm r}$ Voronoi cells, and after each iteration all of the Voronoi cells are updated. In practice, the saturation effect will only be important for small dimensions; for example, for $d = 5$, 90 per cent of saturation is achieved with ≈ 200 uniform random points (using an L_2 norm). Sambridge (1998) showed that the saturation value depends exponentially on dimension, which means that in practice we will nearly always severely under-sample high-dimensional parameter spaces. As a consequence, all Voronoi cells will share faces with all other cells, and as the neighbourhood algorithm switches from one Voronoi cell to another it will always be possible to move into any other point in parameter space (even if $n_{\rm r} = 1$). However, this does not guarantee that, when the algorithm is used as an optimizer, it will avoid local minima, or even perform any better in this respect than any existing method. Secondary minima in high-dimensional spaces may be exceedingly complex, especially if 'ridge-like' narrow valley structures occur. The only safeguard is that when the total number of samples is less than the saturation value the algorithm has the potential to sample any point in parameter space.

6 DISCUSSION

The neighbourhood algorithm is based on some simple principles, contains just two tunable parameters, and yet seems to exhibit sophisticated behaviour in adapting its sampling to the misfit function. Nevertheless, one must be cautious in generalizing these results. Experience with the method is rather limited. We have presented details of its application to a single (synthetic data) problem and compared it numerically with only one alternative method. We do not know how it might perform in much higher dimensional problems (even though we have analysed the computational costs involved). Experience is required across a range of applications before robust conclusions can be drawn.

It has been our goal to produce a method with as few control parameters as possible. No doubt variants and extensions of the approach proposed here could be introduced. For example, if global optimization were of primary interest then it might make sense to combine the NA with a local optimizer (perhaps within each Voronoi cell). In the example presented here this hardly seems necessary because acceptable data-fitting models were found with the NA alone. Other obvious variants might be to choose the cells for resampling in a probabilistic manner based on data fit, or perhaps sample within a Voronoi cell in a non-uniform manner. This might be appropriate if one had a prior probability density function to sample from. Another possibility is to start the NA from a predetermined set of models (rather than a uniform set), perhaps generated using some *a priori* knowledge of the problem (if such information existed), or perhaps using an ensemble generated by some other inversion method.

Overall the NA has been shown to reduce a misfit function efficiently by concentrating sampling simultaneously in different regions of parameter space. A potentially useful feature of the new

algorithm is that it uses only the rank of each model, which makes it independent of the absolute scale of the data fit criterion. Therefore any relative measure (or combination of measures) can be used. In the examples presented here we were able to determine a suitable χ^2 measure of goodness of data fit; however, in other problems more qualitative criteria, non-differentiable functions, or perhaps even heuristic rules could be used equally well.

An interesting feature of the new approach is that it can increase, decrease and shift the density of the sampling (or subportions of it) automatically as the iterations proceed. This is because we use Voronoi cells as the 'local regions of interest', and their size and shape are uniquely determined by the previous samples. In this paper we have also shown how the Voronoi cell concept can be used to enhance existing direct search methods like GA or SA by replacing the forward problem. The neighbourhood algorithm is an alternative to these methods. The original objective with the new approach was to allow the previous samples to guide the search for new models in a straightforward and conceptually simple manner. We suggest that this has been achieved and that the resulting algorithm may find applications in a range of inverse problems.

7 ACKNOWLEDGMENTS

This work has been influenced by a number of people over a long period of time. In particular my thanks go to Jean Braun, Kerry Gallagher, Oli Gudmundsson, and Herb McQueen for stimulating discussions. I am also indebted to Takuo Shibutani who kindly gave me his receiver function inversion code, and tuned the genetic algorithm used in this paper. Brian Kennett, Albert Tarantola and Luis Tenorio suggested improvements to the manuscript. The author's computer programs associated with this work will be made available. See `http://rses.anu.edu.au/~malcolm/na/na.html` for details.

REFERENCES

Aarts, E. & Korst, J., 1989. *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, Chichester.

Ammon, C.J., Randall, G.E. & Zandt, G., 1990. On the nonuniqueness of receiver function inversions, *J. geophys. Res.*, **95**, 15 303–15 318.

Anderssen, R.S., 1970. The character of non-uniqueness in the conductivity modelling problem for the Earth, *Pure appl. Geophys.*, **80**, 238–259.

Anderssen, R.S. & Senata, E., 1971. A simple statistical estimation procedure for Monte Carlo inversion in geophysics, *Pageoph*, **91**, 5–13.

Dosso, S.E. & Oldenburg, D.W., 1991. Magnetotelluric appraisal using simulated annealing, *Geophys. J. Int.*, **106**, 370–385.

Duijndam, A.J.W., 1988a. Bayesian estimation in seismic inversion part I: principles, *Geophys. Prospect.*, **36**, 878–898.

Duijndam, A.J.W., 1988b. Bayesian estimation in seismic inversion part II: uncertainty analysis, *Geophys. Prospect.*, **36**, 899–918.

Gallagher, K. & Sambridge, M. 1994. Genetic algorithms: a powerful tool for large-scale non-linear optimization problems *Comput. Geosci.*, **20(7/8)**, 1229–1236.

Gelfand, A.E. & Smith, A.F.M., 1990. Sampling based approaches to calculating marginal densities, *J. Am. statist. Ass.*, **85**, 398–409.

Geman, S. & Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the bayesian restoration of images, *IEEE Trans. Patt. Analysis Mach. Int.*, **6**, 721–741.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

Goldberg, D.E. & Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms, in *Foundations of genetic algorithms*,pp. 69–93, ed. Rawlins, G., Morgan-Kaufmann, San Mateo.

Gouveia, W.P. & Scales, J.A., 1998. Bayesian seismic waveform inversion: Parameter estimation and uncertainty analysis, *J. geophys. Res.*, , **103(B2)**, 2759–2779.

Hammersley, J.M. & Handscomb, D.C., 1964. *Monte Carlo Methods*, Chapman & Hall, London.

Haskell, N.A., 1953. The dispersion of surface waves in multilayered media, *Bull. seism. Soc. Am.*, , **43**, 17–34.

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

Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press.

Keilis-Borok, V.I. & Yanovskaya, T.B., 1967. Inverse problems of seismology, *Geophys. J.*, **13**, 223–234.

Kennett, B.L.N., 1978. Some aspects of non-linearity in inversion, *Geophys. J. R. astr. Soc.*, **55**, 373–391.

Kennett, B.L.N. & Nolet, G., 1978. Resolution analysis for discrete systems, *Geophys. J. R. astr. Soc.*, **53**, 413–425.

Kirkpatrick, S.C., Gelatt, D. & Vecchi, M.P., 1983, Optimization by simulated annealing, *Science*, **220**, 671–680.

Lomax, A. & Snieder, R., 1995. Identifying sets of acceptable solutions to non-linear geophysical inverse problems which have complicated misfit functions, *Nonlinear Proc. Geophys.*, **2**, 222–227.

Menke, W., 1989. *Geophysical data analysis: discrete inverse theory*, rev. edn, Academic Press, San Diego.

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

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems, *J. Geophys. Res.*, **100(B7)**, 12 431–12 447.

Okabe, A., Boots, B. & Sugihara, K., 1992. *Spatial Tessellations Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, New York.

Parker, R.L., 1977. Understanding inverse theory, *Ann. Rev. Earth planet. Sci.*, **5**, 35–64.

Parker, R.L., 1994. *Geophysical Inverse Theory*, Princeton Univ. Press, Princeton.

Press, F., 1968. Earth models obtained by Monte Carlo inversion, *J. geophys. Res.*, **73**, 5223–5234.

Rothman, D.H., 1985. Nonlinear inversion statistical mechanics, and residual statics corrections, *Geophysics*, **50**, 2784–2796.

Rothman, D.H., 1986. Automatic estimation of large residual statics corrections, *Geophysics* **51**, 332–346.

Sambridge, M., 1998. Exploring multidimensional landscapes without a map, *Inverse Problems*, **14(3)**, 427–440.

Sambridge, M., 1999. Geophysical inversion with a neighbourhood algorithm –II. Appraising the ensemble, *Geophys. J Int.*, , **138**, 727–746.

Sambridge, M. & Drijkoningen, G.G., 1992. Genetic algorithms in seismic waveform inversion, *Geophys. J. Int.*, **109**, 323–342.

Sambridge, M. & Gallagher, K., 1993. Hypocentre location using genetic algorithms, *Bull. seism. Soc. Am.*, , **83(5)**, 1467–1491.

Sambridge, M., Braun, J. & McQueen H., 1995. Geophysical parametrization and interpolation of irregular data using natural neighbours, *Geophys. J. Int.*, **122**, 837–857.

Scales, J.A., Smith, M.L. & Fischer, T.L., 1992. Global optimization methods for multimodel inverse problems, *J. Comput. Phys.*, **103(2)**, 258–268.

Sen M. & Stoffa, P. L., 1995. *Global Optimization Methods in Geophysical Inversion*, Advances in Exploration Geophysics Vol. 4, Elsevier, Amsterdam.

Shibutani, T., Sambridge, M. & Kennett, B., 1996. Genetic algorithm inversion for receiver functions with application to crust and uppermost mantle structure beneath Eastern Australia, *Geophys. Res. Lett.*, **23(14)**, 1829–1832.

Smith, A.F.M., 1991. Bayesian computational methods, *Phil. Trans. R. Soc. Lond.*, A, **337**, 369–386.

Smith, A.F.M. & Roberts, G.O., 1993. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods, *J. R. statist. Soc.*, B., **55**, 3–23.

Snieder, R., 1998. The role of non-linearity in inversion. *Inverse Problems*,

14, 397–404.
Stoffa, P.L. & Sen, M.K., 1991. Nonlinear multiparameter optimization using genetic algorithms: Inversion of plane wave seismograms, *Geophysics*, **56**, 1794–1810.
Tarantola, A., 1987. *Inverse Problem Theory*, Elsevier, Amsterdam.
Tarantola, A. & Valette, B., 1982. Generalized nonlinear inverse problems solved using the least squares criterion, *Rev. Geophys. Space Phys.*, **20**, 219–232.
Thomson, W.T., 1950. Transmission of elastic waves through a stratified solid, *J. appl. Physics*, **21**, 89–93.
Vasco, D.W., Johnson, L.R. & Majer, E.L., 1993. Ensemble inference in geophysical inverse problems, *Geophys. J. Int.*, **117**, 711–728.
Voronoi, M.G., 1908, Nouvelles applications des paramètres continus à la théorie des formes quadratiques, *J. reine Angew. Math.*, **134**, 198–287.
Watson, D.F., 1992. *Contouring: A Guide to the Analysis and Display of Spatial Data*, Pergamon, Oxford.
Wiggins, R.A., 1969. Monte Carlo inversion of body wave observations, *J. geophys. Res.*, **74**, 3171–3181.