Application of Evolutionary Methods to 3D Geoscience Modelling

Brad Alexander
School of Computer Science
University of Adelaide
Adelaide, South Australia
brad@cs.adelaide.edu.au

Jared Peacock
Geology and Geophysics
University of Adelaide
Adelaide, South Australia
jared.peacock@adelaide.edu.au

Stephan Thiel
Geology and Geophysics
University of Adelaide
Adelaide, South Australia
stephan.thiel@adelaide.edu.au

ABSTRACT

Geoscience modelling plays a vital role in mapping and tracking Earth's resources. Magnetotellurics, which maps the electrical resistivity of the subsurface, is a useful and cost-effective technique for sensing large areas at depth. However, due to the inherent difficulty in sensing deep strata, models produced using MT have a degree of uncertainty. Geoscientists can reduce this uncertainty by producing multiple alternative models, and using multiple modelling techniques and settings, to correlate robust model features with field data responses. Population-based evolutionary search techniques are of interest to MT modelling because they offer an alternative to deterministic techniques, and are able to produce multiple models for analysis. Unfortunately, evolutionary techniques have not been successfully applied to 3D MT modelling. In this work we describe a new, more compact, representation of MT models using volumetric functions. Using this representation we successfully apply evolutionary search techniques to 3D MT modelling for both artificial and real models and show how the development of large scale features during modelling can be correlated with the model's fit to field data.

Categories and Subject Descriptors

J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Earth and Atmospheric Sciences*; I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving Control Methods and Search*

Keywords

Modelling, Geoscience, Evolutionary Search, Magnetotellurics

1. INTRODUCTION

Humans are heavily reliant on geological knowledge to discover and conserve the Earth's material, energy and water resources. Due to the inherent difficulty in sensing structures at depth, much of our geological knowledge is provisional and ambiguous[10, 15]. Given this unavoidable uncertainty, there is a need to develop modelling techniques to help geologist explore the space of viable models. One way to explore this space is to have a population of viable models that allows common traits to be observed and correlated with field data. For this reason, there is a strong research interest in the application of population-based evolutionary search to geoscience modelling[6, 11, 4, 13, 18, 7].

One geoscience modelling technique to which evolutionary search has been successfully applied is Magnetotellurics. Magnetotellurics (MT) measures the electrical response of the subsurface due to natural time-varying magnetic fields. MT is a diffusive volumetric method, where the depth of penetration depends on the period of the source field. MT field data can be processed to form a map (model) of the resistivity of the subsurface. MT is relatively cost-effective and can model the earth to great depth. MT is used for mineral exploration[14], mapping geothermal-provinces[9], and mapping fluid reservoirs at depth[17]. The MT modelling process can produce one, two and three-dimensional models. Evolutionary search has been used in the derivation of simple 1D and 2D MT models[9, 13, 18, 7] but, to date, there are no published applications to not 3D models. This omission is significant because 3D models not only provide much more detail than 2D models, they also reflect the complex three-dimensional geometries of earth's subsurface structures [15]. The lack of evolutionary search applied to 3D models is, at least in part, due to the large size of 3D models which can consist of thousands to millions of real-valued parameters.

This work describes an alternative encoding of MT models that requires many fewer parameters but is still capable of expressing non-trivial subsurface structures. Instead of using the common encoding of 3D models as grids of hexahedral cells (structured meshes), we express our model using small set of diffuse 3D ellipsoidal volumetric functions. These functions, inspired by metaballs, or blobs, used in computer graphics[1] are sampled to 3D mesh to represent an MT model. This new representation produces a new search space for which we have developed a hybrid search technique, combining bespoke greedy search with Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES)[5]. We have integrated these components with existing MT modelling tools to produce a new modelling framework. The contributions of this work are two-fold: First, the novel use of volumetric functions to compactly express non-trivial 3D maps of values in a scientific domain; and second, a new hybrid search technique, incorporating evolutionary search,

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.
Copyright 2012 ACM 978-1-4503-1177-9/12/07 ...$10.00.

that traverses the space of volumetric functions to produce good 3D MT models for both artificial and real field data.

The reduced number of parameters used to express our models is significant. Not only do fewer parameters make the use of population-based search methods feasible, but their simple relationship to the actual shape of the blob-structures in the model creates a much more intuitive and comprehensible link between the parameter space and model fitness.

The remainder of this article is structured as follows. In the next section we summarise related work. In section 3 we outline the MT modelling process. We describe our blob representation in section 3.1 and our hybrid search technique and modelling process in section 4. In section 5 we present our results for both an artificial benchmark and real field data. Finally, in section 6 we canvass ways in which the search framework might be improved and further explored.

2. RELATED WORK

Stochastic search has been widely applied to MT. Monte-Carlo techniques[6, 11] where parameters for MT models are generated randomly are applied to provide better global search. Faster search is provided by informed search heuristics such as Markov chaining[4], simulated annealing[3] and evolutionary methods[9, 13, 18, 7]. Work using evolutionary methods has been diverse, ranging from direct evolution of 2D models using specialised operators (Flores and Schultz[9]); to hybrid schemes combining genetic algorithms and local search (Tong and Lui, et. al.[18]); to using pareto techniques to balance model fitness and model smoothness (Moorkamp, Jones, and Fishwick[7]). To date, MT work using stochastic methods has been limited to 1D and 2D models. Our work extends the use of evolutionary search to small 3D models by using a parameter-reducing representation.

In terms of representation, parametric models, often describing layers[6], or using simple functions[12], have long been used for MT. Schnegg[12] expressed 3D MT models using polynomial functions in terms of curved layers. His representation, like ours, substantially reduces the parameter space for models. However, our representation is able to represent more complex and diffuse structures, does not rely as much on a-priori knowledge of the model.

3. MAGNETOTELLURIC MODELLING

MT modelling builds a map of subsurface resistivities to match field data. The broad process of MT is shown in Fig 1. The first step of the process is data collection. MT data is collected by measuring orthogonal horizontal components of the electric and magnetic fields. In addition, sometimes a vertical magnetic component is measured[1]. Instruments, called *stations*, are left in the field for an extended period of time depending on the target depth. The time-series extracted are then converted to the frequency domain to estimate the MT transfer function as the ratio of the electric field to magnetic field. Robust statistics are applied to extract the best representation of the complex MT transfer function, known as the impedance tensor. An impedance tensor is derived for each frequency for each station. In in this work, each impedance tensor consists of eight scalar components. We do not detail these components here, but we use the same components as those described briefly in [8] and in a deeper account in [14].

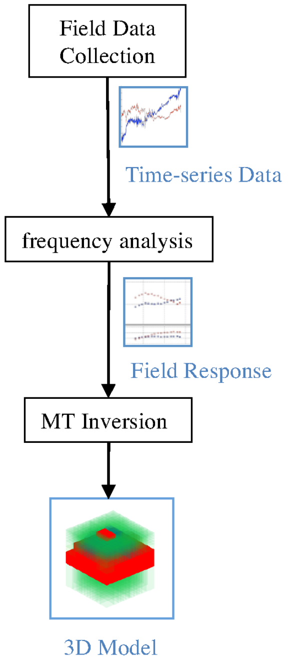


Figure 1: Summary of the MT modelling process

```
1   $\epsilon$ = min_value
2   models = initial_guesses
3   loop
4       models = improve(models)
5       errors = calculate_errors(field_data, models)
6   while min(errors) > $\epsilon$
7   return best_of(models)
```

Figure 2: General algorithm for MT inversion

After derivation of the impedance tensors we commence the third, and most computationally intensive, phase: *MT Inversion*. Inversion converts impedance tensors into an MT model. An MT model is a a one, two or three dimensional mesh of resistivity values – literally, a map of the subsurface below the stations with a resistivity value ρ, measured in Ohms per meter (Ωm), at each mesh cell. A general algorithm for MT inversion is given in Fig 2. The algorithm works by iterative improvement of a population of one or more models. The inversion process can take from a few hours to a few months of processor time depending on algorithm settings and model size. The first line of the algorithm sets ϵ to the minimum acceptable error value for the best of the models. Line 2 initialises the models to a list of initial guesses. The algorithm then loops, improving the set of guesses on line 4, and calculating errors on line 5. Line 6 test the terminating condition and line 7 returns some best sub-set of the refined models[2].

Note there are several open design choices in the MT inversion algorithm in Fig 2, including the size of the popu-

[1] as it is in this work

[2] The criteria for *best* is user-defined but it is usually some combination of low-error and smoothness.

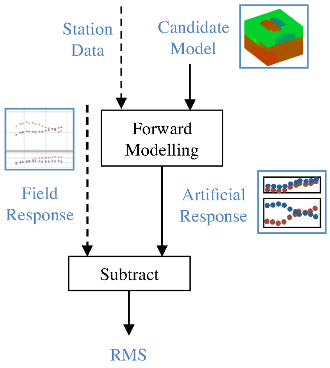


Figure 3: Evaluative function: calculate_errors

lation and the choice of improvement function. For example, a very common choice in MT inversions is to have a population of just one model and to use a standard nonlinear search algorithm[15, 14] for improvement. In contrast, most evolutionary approaches have a population size greater than one and a stochastic or hybrid improvement process[13, 18, 7]. In an evolutionary setting, the calculate_errors function performs the role of evaluative function over the population. Fig 3 shows the structure -of the evaluative function for a single individual. The key component of the evaluative function is a process called *forward-modelling*. Forward-modelling takes as input a candidate model, the set of station locations, and the frequencies to be produced. As output, it produces an artificial response – the set of impedance tensors that *would* have been produced for the stations and frequencies *if* the earth below the stations *were* the candidate model. There are a number of techniques for forward modelling[14]. These typically use iterative approximation to produce its response through finite-difference or finite-element methods. The forward-modelling method we use in this work is Siripunvaraporn's data-space method[16] because it has good speed, generality and accuracy. After forward-modelling has produced an artificial response, the rest of the evaluative function is straightforward: we simply pairwise-subtract the field response from the artificial response (weighting for estimated errors in field data). Then the root-mean-squares (RMS) of these weighted differences is calculated to give a single numeric estimation of error. For any given model, the lower the RMS, the better the fitness.

This concludes our brief introduction to the process of MT. Next we describe our new representation for MT models using blobs.

3.1 The Blob Model

The blob model is a parametric representation of a 3D model using simple ellipse-shaped volumetric functions. In our modelling framework these blob parameters must first be converted a single composite volumetric function and then sampled to produce a candidate model for evaluation. This process is illustrated in Fig 4. We implemented this conversion and sampling process from blob parameters to the *res-map* function in Python because Python supports direct manipulation of functions. We describe the two steps of this evaluation process, function formation, and sampling in turn.

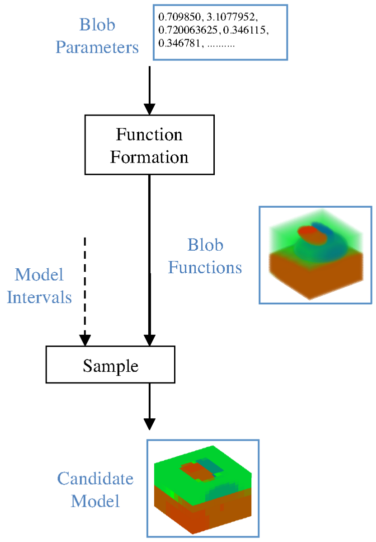


Figure 4: Conversion of blob parameters to MT Model

3.1.1 Function Formation

Function formation converts a set of blob parameters to a function *res-map*(x, y, z) that maps a Cartesian coordinate (x, y, z) into a resistivity value: ρ. *res-map* is a composite function consisting of a generalised average of a smaller set ellipse-shaped volumetric functions (or blobs) $\{f_1, \ldots, f_n\}$ where n is typically in the range of 5 to 20 depending on model complexity. We will describe how the f_i are combined to form *res-map* shortly but first we will describe the encoding the individual blob functions f_i. Each $f_i(x, y, z)$ maps a point (x, y, z) in 3D space into a scalar value representing the deviation of that point in space from a, user-estimated, background level of resistivity (this background level is called the *half-space*). All f_i are kept to the same range, usually, ± 0.5 which is scaled to produce a real resistivity value when all functions are combined to form *res-map* Each blob function f_i has the form:

$$f_i(x, y, z) = \delta_i / ((x_{tr}^2 + y_{tr}^2 + z_{tr}^2)^\alpha + 1)$$

where δ_i is difference of resistivity from the half-space at the centre of the blob; α is a power that determines how fast the blob diffuses into the background – a high α will produce a sharply defined blob whereas a low α produces a very diffuse blob. The input coordinates (x, y, z) are geocentric – x and y are in the horizontal plane and z is the vertical depth. For the sake of simplicity, we normalise x, y and z to be in the domain $[0.0, 1.0]$. The triple (x_{tr}, y_{tr}, z_{tr}) in the function above represents (x, y, z) after it has been transformed by 9 parameters representing translation, scaling and rotation in all three dimensions. In all, including these translation, scaling, and rotation parameters, each blob is defined by 11 parameters. Thus, a large model consisting of 20 blobs will have a total of 220 real-valued parameters. The 11 blob parameters are summarised in table 1. Note that this representation was inspired by, and bears some resemblance to, the volumetric functions used to represent blobs or metaballs in computer graphics [1]. The main difference being the use of a variable sharpness parameter α and the fact blobs are com-

Param	Description	Range
δ	central intensity	$[-0.5, 0.5]$
α	sharpness	$[0.1, 15.0]$
$x_{loc}, y_{loc}, z_{loc}$	x,y,z-location	$[0.0, 1.0]$
x_s, y_s, z_s	x,y,z-scale	$[0.0, 1.0]$
x_r, y_r, z_r	rotation about x,y,z axis	$[0.0, \pi]$

Table 1: Summary of blob parameters

bined by averaging their sampled values rather than joined with spline functions.

A two-step process is used to combine the f_i to form a *res-map*. First individual blobs functions f_i are combined:

$$f(x, y, z) = (\frac{1}{n}\sum_{i=1}^{n} f_i(x, y, z)^p)^{1/p}$$

to produce a composite function f which is a generalised average of the sampled value of each blob function f_i at the point (x, y, z). In our experiments to date, we have found good results if we set the power of the average to: $p = 11$ which biases f in favour of the most extreme f_i at each point without completely subsuming the effects of the other f_i at that point. Note that p must be an odd integer to preserve negative values. Finally, after f is formed, we can use it to define the *res-map* function:

$$\textit{res-map}(x, y, z) = 10^{f(x,y,z)/0.2}\Omega m + \textit{half-space}$$

where *half-space* is the background resistivity mentioned previously. *half-space* is a rough-estimate of background resistivity , usually between the values of $10^1\Omega m$ and $10^3\Omega m$, depending on the geological area under study. Once *res-map* has been created it can then be sampled and evaluated. We describe this process next.

3.1.2 Sampling and Evaluation

To convert the *res-map* function to a candidate model we simply sample *res-map* at the desired intervals of the model to produce a 3D hexahedral mesh representing the model. For the sake of speed, we currently sample only one point at the centre of each mesh cell. Once the model is sampled, it is given to the calculate_errors function shown in in Fig 3 which produces an error (RMS) value that guides the search process we describe next.

4. THE SEARCH PROCESS

We can frame the search problem as:

$$\underset{ps=[p_1,\ldots,p_n]}{\text{argmin}} \quad \textit{evaluate}(ps)$$

where *evaluate* is the sampling and evaluation process, described in section 3.1.2 above, which converts the model parameters *ps* to an RMS value. Conventional MT inversion often uses standard non-linear search methods to directly optimise the resistivities of the model cells[3]. Unfortunately, our attempts to apply standard non-linear search techniques blob models resulted in fast convergence to very sub-optimal local minima. Likewise, we found that direct application of a variety of GA's results in only limited progress even for simple models. To investigate further, we closely inspected the fitness landscape surrounding known global optima of artificial blob models. We found this landscape for various combinations of parameters to be reasonably smooth on a large scale but sometimes rough on a local scale. To cope with this mix of global smoothness and local roughness we developed a composite search method consisting of the following steps.

- **Seeding** Progressively adding alternately resistive and conductive spherical blobs of a variety of sizes and depths and allowing them to be optimised for position and resistivity.
- **Priming** Cyclically optimising groups of model parameters.
- **Culling** Removing from the model the blobs that have the least impact on fitness.
- **Model Evolution** Using Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES)[5] to allow for simultaneous search on all parameters.

In our experiments to date, we have needed to run each of the steps above to produce good final models. We summarise each step next.

4.1 Seeding

Seeding adds spherical blobs to the model one at a time. The number of blobs added varies between 5 and 20 depending on a-priori assumptions we make about the complexity of the model. Blob resistivity alternates between higher than half-space and lower than half-space. Blobs vary in size, smaller blobs are added near the surface of the model first due to a higher sensitivity of MT to near-surface features and larger blobs are added at depth later on. In our experiments thus far we have found that our inversion process is relatively insensitive to the precise position and resistivity of blobs added – as long as there is sufficient variety at each depth and shallow blobs are added first. Other parameters are fixed, with rotations set to zero and sharpness set to it's maximum value. When each blob is added it is optimised first for its position parameters and then for resistivity using the bespoke perturbation algorithm shown in Fig 5. Note that this greedy algorithm refines models by cycling through the selected model parameters, perturbing the model parameters by a set amount[4] and recording the perturbation that gives the greatest improvement (if any). The selected perturbation is then applied and the search continues. If no perturbation improves the model then the search continues with reduced step size until a minimum step size is reached[5].

4.2 Priming

After seeding is performed, the model consists of a number of spherical blobs distributed through the model space. The hope is that many of these blobs are, roughly, in the right location and have the right resistivity. Priming performs preliminary optimisation of all the parameters in turn using same greedy search algorithm as the one for seeding (Fig 5).

[3] often using extremely large Jacobian or hessian matrices in the process

[4] In our experiments start perturbations at one-quarter of the maximum parameter range.

[5] To speed up search a little, we modified the algorithm to also adaptively increase step size when justified by the progress of the search.

```
input:
   model          //$\{m_1, .., m_n\}$
   params         //$\{p_1, ..., p_k\}$, $k \leq n$
   maxstepsize// in range $(0.0, 1.0]$
   minstepsize // in range $(0.0, \text{maxstepsize})$
algorithm:
stepsize=maxstep
evaluate initial fitness
while stepsize >=minstepsize
   for i in 1 to k
      perturb model at param i up by stepsize
      perturb model at param i down by stepsize
      record best perturbed paramater so far
   end for
   if if best recorded fitness improved
      replace model with best perturbed model
   else
      halve stepsize
   end if
end while
return model
```

Figure 5: Greedy algorithm for perturbation search

Optimisation starts with x, y and z scales of all blobs, progressing to all the resistivities, all sharpness parameters, all rotations and all positions. This cycling through the whole parameter space is typically done twice.

4.3 Culling

After priming, the model usually has a number of blobs making a net negative contribution to fitness. Culling removes a number of these blobs by first sorting the blobs according to contribution and then progressively removing blobs until the RMS plateaus.

4.4 Evolution

After culling, the model has most of the structure necessary to produce a reasonably low error value. However, in MT, many large scale structures, especially resistive bodies at depth, exhibit only a weak signal to the surface probes. A tuning step is necessary to evolve the large scale shape of these structures in response to very subtle changes in RMS. We use CMA-ES applied to all model parameters to perform this tuning. In our experiments to date, the number of model parameters involved varies between 55 and 220. The population, λ, varies between 50 and 200 and the number of individuals retained at each iteration was set to $\lambda/2$. In our later experiments we set the standard deviation (σ) of the initial search distribution to be small, approximately, 1% of the parameter range. This is due to the high sensitivity of fitness to changes of some model parameters[6] We let evolution run for between 1000 iterations for simple models with few blobs to 2000 iterations for models with many blobs resulting in between 50000 and 100000 evaluations[7].

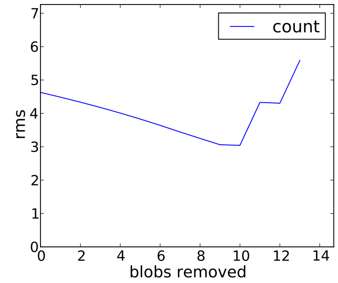


Figure 6: Effect of removing blobs in order of least contribution from primed model

5. RESULTS

The results presented here are for two sets of data. The first experiment ran against a simple standard benchmark for 3D MT, the 3d2 COMMEMI model[19]. The second experiment ran against field data taken from an enhanced geothermal prospect near Paralana in South Australia[17]. In both experiments we used CMA-ES with a population of 50 individuals from which 25 were selected to form the basis of the next iteration. The initial σ was set to 0.01 representing 1% of the range of most parameters. The CMA-ES process was allowed to run for 1000 iterations in the first experiment and 2000 in the second. We describe each result in turn.

5.1 COMMEMI 3d2 model

The COMMEMI 3d2 is one of a series of models developed to test MT inversion techniques. It is a reasonably challenging target in that it has highly contrasting bodies near the surface and alternating conductive and resistive layers. In this context, a further challenge is added by fact that the bodies in the model are rectangular which makes them more difficult to approximate using ellipsoid functions.

To create the response data needed for this experiment we ran forward modelling on the COMMEMI model shown in Fig 7 (a). We placed 50 simulated stations in a rectangular grid near the surface of the model. Data was collected for five frequencies ranging from 2Hz down to 0.0001Hz. The purpose of inversion here is then to re-discover the model from the response data. We start the inversion process with a half-space of $10\Omega m$ and sampled into a 13x14x10 grid. This relatively low grid resolution allowed forward-modelling to run in just a few seconds on a Linux virtual machine running on a 2.8 GHz Intel Core i7 processor. In terms of runtime, the seeding and priming process used 19750 evaluations taking just under 17 hours. Culling required just 29 evaluations and took less than two minutes. The evolutionary stage of the process consisted of 50000 evaluations and took 1.5 days giving a total runtime for the inversion of just over two days. In the first step of the inversion the model was seeded with 14 blobs. The model, after seeding is shown in Fig 7(b) some of the basic features are starting to become visible, particularly near the surface, the RMS value is high to moderate at 7.68 (down from over 50 in the initial empty model). By the end of priming (part (c)) the model has a stronger resemblance to the target, particularly at depth, and the RMS has decreased to 4.62. In the culling

[6] In earlier experiments we found that progress could be made, albeit more slowly, with σ up to 12% of the parameter range. Values larger than this lead to divergence.

[7] From earlier experiments running with different populations, it appears that the total number of evaluations rather than population size is the best predictor of model refinement.

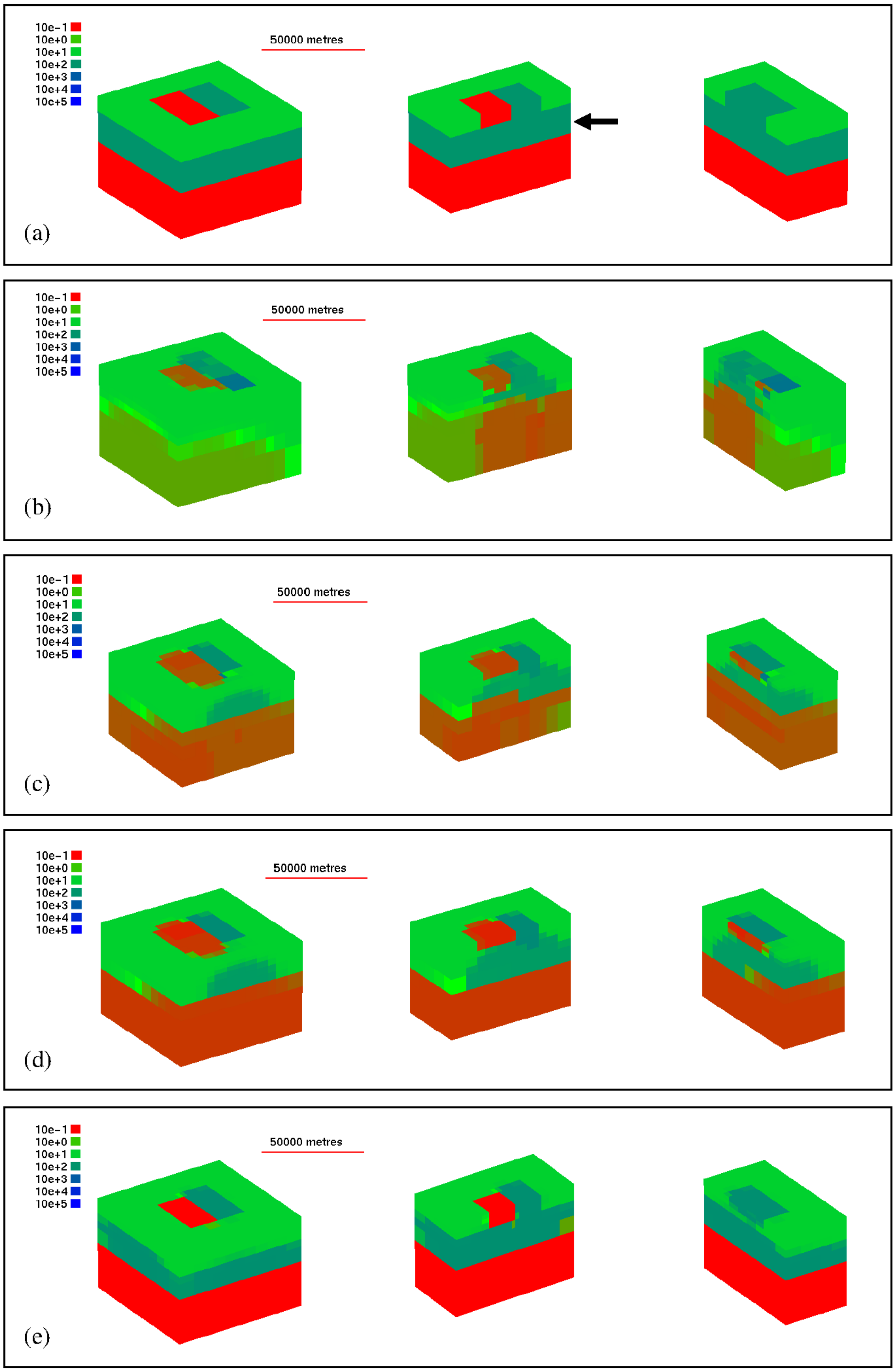


Figure 7: COMMEMI results showing the target model (a), the model after seeding (b), the model after priming (c), the model after culling (d) and the best model after evolution using CMA-ES (e). The arrow in part (a) points to a resistive layer referred to later in this text.

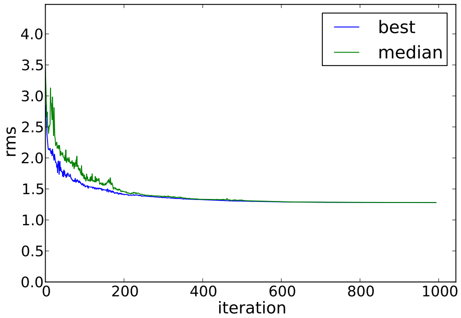


Figure 8: Fitness of COMMEMI model over CMA-ES run

stage we selectively removed blobs to assess their impact, positive or negative, on the RMS value and sorted the models. The graph of the RMS response resulting from removing blobs one-by-one from the primed model is shown in Fig 6. The RMS value plateaus for a short time when nine and ten blobs are removed. To be conservative, we removed only nine blobs[8]. The resulting model is shown in part (d), as can be seen, culling appears to have improved the accuracy of deeper parts of the model somewhat. The final step of the process was to evolve all the remaining 55 model parameters using CMA-ES. In this case we ran the evolutionary process for 1000 iterations. The best individual from this evolutionary step (RMS=1.27) is shown Fig 7 part (e). As can be seen, the evolutionary process has resulted in a model that bears a close resemblance to the target model (part (a)). This result is competitive with other inversion processes for this and similar models[16] though our process is still slower than standard non-linear search which, for this model size, would take hours rather than days.

The biggest differences between the culled model in part (d) and the evolved model in part (e) are, first, more accurately approximated resistivity in the deeper layers in the model and, second, a much more accurate shape for the blue-green resistive layer (see arrow). The trajectory of the RMS value during evolution is shown in Fig 8. Note that, at first RMS drops quite rapidly. From our observations of models during the early part of the evolutionary run, this steep drop in RMS appears to correspond to improved shape of surface features and improved approximations of resistivity. Later stages of evolution show only very slow improvement in RMS. However, even during this quiescent stage, the model is developing substantially. In particular, the blue-green resistive layer is becoming more planar in shape. This late development can be seen in Fig 9, which tracks the size of blob corresponding to the resistive layer in the x,y and z dimensions over time[9]. Note that quite far into the evolutionary process the shape of this layer is still improving even though the change in RMS driving this improvement is very weak. This ability to easily track the

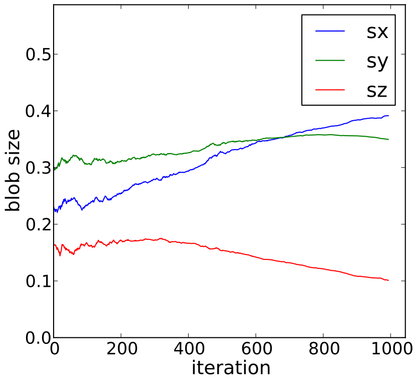


Figure 9: Development of size parameter values for the resistive layer during evolution

correspondence between the development of large-scale features in the model to the RMS is a new advantage of using our parametric model representation.

5.2 Paralana

Our second experiment applies our blob modelling method to real geological data. The data was collected from a site near Paralana in South Australia that shows promise as an enhanced geothermal power source[17]. The response data was collected from 54 stations and processed tensors in six frequencies ranging from 50Hz to 0.01Hz. For the inversion we again, used a low-resolution model (13x13x10 cells).

To start the inversion process, we seeded our model with four layers of four blobs each. The initial seeding and priming process produced a model with an RMS of 2.58. We ran CMA-ES for 2000 iterations - just under four days - and arrived at a model with an RMS of 2.51. This model is shown in Fig 10 (a). For comparison, the output of an earlier 2D inversion using the deterministic OCCAM[2] method is shown in part (b)[10]. Both models seem to capture the conductive surface layer over a more resistive crustal layer. They also appear to capture a conductive body at depth on the right hand side (see arrows). This conductive body is present in all of our other 2D inversions at roughly this location, which gives some confidence that it is real - perhaps corresponding to a large heat-source.

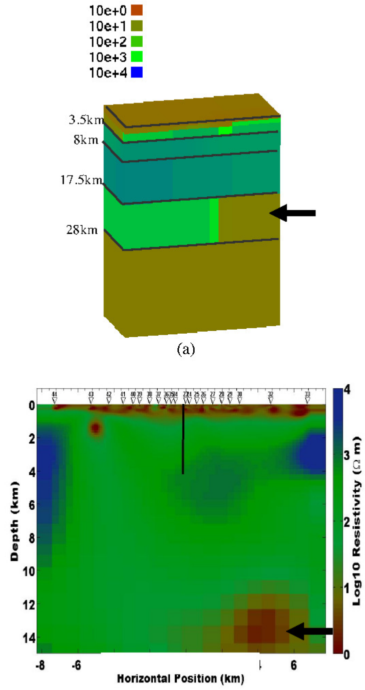
6. CONCLUSIONS AND FUTURE WORK

The results presented in this paper represent a proof-of-concept – they show that CMA-ES, given a compact model representation, can be used in the production of good MT models. There are several useful directions in which this work can be taken. First, as part of its execution, CMA-ES produces a covariance matrix. There is work to be done in terms of interpreting this matrix to convey geologically significant information about the model-space. Second, our greedy search algorithm used by seeding and priming is very naive – there is much scope for refining for both its efficiency and accuracy. Third there is scope for speeding up the modelling process by exploiting parallelism at all stages. Fourth,

[8] It should be noted that any approach to culling is, necessarily, imperfect because it anticipates what blobs will become during the evolution step.

[9] Almost no rotation was applied to this blob so the directions of each dimension are intact.

[10] Note, we use a 2D inversion here because there are no extant 3D inversions available for this site. Also note, the vertical line in part (b) represents a geothermal test-well.



(a)

(b)

Figure 10: West/East cross-section of 3D Paralana Blob model (a) and West/East 2D OCCAM inversion.

we would like to extend the search process to periodically allow highly fit blobs to divide and less fit blobs to be culled. Such a biological growth model has the potential to greatly increase the adaptivity of the search process. Finally, we would like to experiment more with the use of stochastic search a global as well as a local level to assess it's impact on model diversity.

Acknowledgment

Thank you to Craig Mudge and Pinaki Chandrasekhar for their unwavering moral and technical support. Thank you to Petratherm for providing the data for the Paralana inversion.

7. REFERENCES

[1] J. F. Blinn. A generalization of algebraic surface drawing. *SIGGRAPH Comput. Graph.*, 16:273–, July 1982.

[2] C. deGroot Hedlin. OccamâĂŹs inversion to generate smooth, two-dimensional models from magnetotelluric data. *Geophysics*, 55(12):1613, 1990.

[3] S. E. Dosso and D. W. Oldenburg. Magnetotelluric appraisal using simulated annealing. *Geophysical Journal International*, 106(2):379–385, 1991.

[4] H. Grandis, M. Menvielle, and M. Roussignol. Thin-sheet electromagnetic inversion modeling using monte carlo markov chain (mcmc) algorithm. *Earth Planets Space*, 54:511–521, 2002.

[5] N. Hansen and P. K. P. E. Ch. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11:1–18, 2003.

[6] A. G. Jones and R. Hutton. A multi-station magnetotelluric study in southern scotland. monte-carlo inversion of the data and its geophysical and tectonic implications. *Geophysical Journal of the Royal Astronomical Society*, 56(2):351–368, 1979.

[7] M. Moorkamp, A. G. Jones, and S. Fishwick. Joint inversion of receiver functions, surface wave dispersion, and magnetotelluric data. *J. Geophys. Res.*, 115, 2010.

[8] J. C. Mudge, P. Chandrasekhar, G. S. Heinson, and S. Thiel. Evolving inversion methods in geophysics with cloud computing - a case study of an escience collaboration. *eScience, IEEE International Conference on*, 0:119–125, 2011.

[9] M. A. Pérez-Flores and A. Schultz. Application of 2-D inversion with genetic algorithms to magnetotelluric data from geothermal areas. *Earth, Planets, and Space*, 54:607–616, May 2002.

[10] A. M. Reading, M. J. Cracknell, and M. Sambridge. Turning geophysical data into geological information or Why a broader range of mathematical strategies is needed to better enable discovery. *Preview, CSIRO Publishing*, 151:24–29, 2011.

[11] M. Sambridge and K. Mosegaard. Monte carlo methods in geophysical inverse problems. *Rev. Geophys.*, 40(3), 2002.

[12] P.-A. Schnegg. A computing method for 3D magnetotelluric modelling directed by polynomials. *Earth, Planets, and Space*, 51:1005–1012, Oct. 1999.

[13] C. Schwarzbach, R.-U. Börner, and K. Spitzer. Two-dimensional inversion of direct current resistivity data using a parallel, multi-objective genetic algorithm. *Geophysical Journal International*, 162:685–695, Sept. 2005.

[14] F. Simpson and K. Bahr. *Practical Magnetotellurics.* Cambridge University Press, 2005.

[15] W. Siripunvaraporn. Three-Dimensional magnetotelluric inversion: An introductory guide for developers and users. *Surveys in Geophysics*, pages 1–23, May 2011.

[16] W. Siripunvaraporn, G. Egbert, Y. Lenbury, and M. Uyeshima. Three-dimensional magnetotelluric inversion: data-space method. *Physics of the Earth and Planetary Interiors*, 150(1-3):3–14, 2005.

[17] S. Theil, J. Peacock, G. Heinson, P. Reid, and M. Messeiller. First results of monitoring fluid injection in egs reservoirs using magnetotellurics. In *Australian Geothermal Conference*, 2010.

[18] T. Xiao-zhong, L. Jian-xin, S. Ya, L. Wen-tai, and X. Ling-hua. A hybrid optimization method based on genetic algorithm for magnetotelluric inverse problem. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1 –4, dec. 2009.

[19] M. Zhdanov, I. Varentsov, J. Weaver, N. Golubev, and V. Krylov. Methods for modelling electromagnetic fields results from commemi the international project on the comparison of modelling methods for electromagnetic induction. *Journal of Applied Geophysics*, 37(34):133 – 271, 1997.