MAGNETOTELLURIC INVERSION USING PROBLEM-SPECIFIC GENETIC OPERATORS

Pedro Luis Kantek Garcia Navarro
Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil.
kantek@pr.gov.br

Pedro P.B. de Oliveira
Instituto de Pesquisa e Desenvolvimento, Universidade do Vale do Paraíba, São José dos Campos, SP, Brazil.
pedrob@univap.br

Fernando M. Ramos
Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil.
fernando@lac.inpe.br

Haroldo F. Campos-Velho
Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil.
haroldo@lac.inpe.br

Abstract

An evolutionary computation approach is described for a classical geophysics inverse problem, namely, magnetotelluric inversion. The problem's nature and features are briefly described, as well as the procedure used to approach it, based upon the development of three problem specific operators: local search, homogenisation and spatial crossover. The latter is presented in detail and relevant data comparing its performance with uniform crossover are provided. Comparison is then made between the solution obtained with evolutionary computation and another, a classical approach, more akin to the inverse problems field. Conclusions are drawn for the necessity of using *ad-hoc* operators in this problem, and establishes the balance between the two approaches: evolutionary computation generates better results while consuming more computing resources.

1 INTRODUCTION

Evolutionary computation was introduced as a tool for the solution of complex problems. One of its advantages has been the ability to tackle problems that have not yet been completely solved; in these cases, it became unnecessary to know how to find *the* solution, but only to recognise how good a potential solution is, regardless the way it was generated. Naturally, there is a price to be paid as we forego prior knowledge. If, on the one hand, one does not need to dominate 100% of the problem, on the other, one has to build and adequate evolutionary engine associated to the abundant computing resources which are consumed in the process.

Within the universe of evolutionary computation some practitioners argue for using the same simple and sufficiently tested evolutionary framework across the applications. The main justification for this view is the possibility of obtaining theoretical data (as, for example, the schema theorem or the building blocks hypothesis (Goldberg 1989; Forrest and Mitchell 1992) that may support the use of evolutionary computation procedures. Such theoretical data have been obtained mainly for engines that could be called "canonical" and, in principle, would consist of: a) binary codification; b) fitness-proportional or rank-based selection; c) 1-, 2-, n-point or uniform crossover; and d) conventional mutation.

However, as well pointed out in Davies (1991), other researchers make a case for using *hybridisation,* where problem-dependent knowledge is introduced into the evolutionary computation engine, so as attain an improved performance. According to Davies (1991) this can be done using three principles: 1) Algorithms adapted to the problem can be used to generate some of the individuals of the initial population. By using elitism one can guarantee that the performance of the evolutionary engine will not be worse than the already existing algorithm; 2) Incorporating the already known algorithm heuristics or procedures to the genetic operators; 3) Enriching the evolutionary algorithm with specific coding schemas.

Kodyalama *et al.* (1996) say: "the most creative step in the evolutionary process is creation of appropriate genetic operators for the particular problem addressed"; accordingly, they designed a problem-specific operator (called *switching*), and obtained good results in the context of designing satellite components. In a similar vein, Bruns (1993) defines *knowledge-augmented genetic* *operators*, those that incorporate knowledge of the problem so as to improve their performance.

Here we report on an evolutionary computation approach that was able to produce a solution to an inverse problem only through the design and use of *ad-hoc* evolutionary operators.

2 THE PROBLEM

Obtaining the pattern of underground electric conductivities in some region of the Earth, based on measurements of the electromagnetic field at the surface, is a subject of great interest. This problem, called *magnetotelluric inversion,* is a classical problem that appears in many applications in geophysics such as oil prospection, mining, underground water prospection, etc, having high relevance for the exploration of regions that are difficult to study through conventional seismic methods (Ramos and Velho 1995). Overall, what one wants to know then is how to obtain underground (conductivity) data, once surface data (electromagnetic field) are known. The corresponding *forward problem* – obtaining electromagnetic fields at the surface, from underground patterns of electric conductivity – is solved through Maxwell's equations in a way that is much simpler than the inverse problem. To solve an inverse problem means to deduce the unknown causes of a phenomenon from the observation of its effects; this is in contrast to solving a forward problem, in which the we look for the effects of a phenomenon, out of the complete knowledge of its causes (Woodbury 1996).

When the problem being investigated is mathematically described according to its inverse functions it can be treated analytically. It suffices to find the inverse functions and to solve them. But this is rarely the case. In general, there is no such thing as an inverse function. That is why one is forced to accept some degree of degeneration in the results. According to this the inverse problem theory was called "diagnosis theory" (Sabatier 1985) in which decisions are taken within environments with varying degrees of uncertainty. A solution of an inverse problem may be accepted when, even if it does not represent a single and ultimate result, it eventually leads to: a) converging answers around a correct result; b) a larger information set related to this result; c) higher certainty degree in terms of decisions taken on an applied problem; and d) any combination of the previous three in an optimised way, that is, providing an appropriate balance between computing load required and results obtained.

Several techniques may be used when looking for inverse problem solutions. The one discussed here is non-linear optimisation. The objective function is the difference between field or synthetically generated data and those produced by the forward model, which represents an error measure for the candidate solutions. This function is then optimised through an evolutionary computation procedure. This paper follows another, presented by Ramos and Velho (1995), that also uses an iterative method (although based on a standard gradient-based optimiser) in addition to explicit regularisation procedures. The problem here is the same, but the optimiser has been replaced by an evolutionary computation procedure. In either case, the forward model involved is the same one presented in the original paper, where its mathematical description was described. This forward problem is used here on a black-box modality.

Every individual in the population is represented by a matrix of 7 x 10 real numbers, each one referring to a rectangular slice of underground material (in fact, a prism cut by a half-plane) of unknown conductivity. These conductivities multiply the value $4\pi \times \omega \times 10^{-10}$ and are given in mhos/m, and the dimensions of the underground rectangles are Δ y = 10 km and Δz varying from 1 to 10 km. The objective function to be maximised is

$$f = \frac{1}{(1+K)^{\varepsilon}}$$

where f is the fitness of an individual, K is a parameter (kept constant, equal to 0.01, in the present case) that allows for the selective pressure to be tuned, and ε is the *magnetic error* given by

$$\varepsilon = \sum_{i=1}^{440} \left| H_p - H_c \right|$$

In the equation above H_p is the component of the magnetic field generated by the reference individual (obtained from field measurement or, as is the present case, defined synthetic data) and H_c is the calculated component by the evolutionary engine. For simplicity of presentation (but with no loss of generality), only the magnetic field will be considered in this paper. The number 440 results from the fact that measurements are being made at 11 points on the surface of the earth, in 20 frequencies (varying from 0.0001 and 0.01 Hz), yielding both real and imaginary components of the magnetic field at those points. In a situation with real data ε would be the only error capable of being obtained. However, since the present case uses synthetic data, the original individual the evolutionary process will try to reconstruct is already known. So although the evolutionary search is guided by the error ε, it is possible to define a second error measure, the *conductive error,* given by

where E is obtained out of the 70 absolute differences

$$E = \sum_{j=1}^{70} \left| C_p - C_c \right|$$

between the reference and the calculated conductivities. The reason for presenting these two kinds of error resides in the fact that, due to noise (a typical ocurrence in the inverse problem context), minimisation of the magnetic error not necessarily entails minimisation of the conductivity error. In fact, pushing minimisation of the former error too far, may lead to an increase of the latter. Naturally, the evolutionary search process has to avoid being misled by this feature.

3 ON PROBLEM SPECIFIC OPERATORS

Several authors have investigated evolutionary operators adapted to the problem being studied. What one waists in terms of generality for not being able to reuse computer codes is balanced by the increased performance attained by the evolutionary engine. Guerreiro *et al.* (1998) affirm that "in the GA, one is free to devise other operators"; in tune with this statement, they present the idea of using three different crossover operators with different rates for each one.

Discussing structural optimisation of three-dimensional objects, Kajiwara and Nagamatsu (1996) suggests that the evolutionary engine should make the conversion of a one-dimensional chromosome into another, defined in three dimensions; in this way, spatial relationships derived from neighbouring elements in the original objects remain preserved in the chromosomes that represent them. For that they created two crossover operators (named *scooping out* and *cutting off*), whose common idea is the generation of three-dimensional blocks that are swapped over among parents.

Cartwight (1993) introduced UNBLOX (Uniform Block Crossover), a two-dimensional crossover operator. It comes from adapting a two point crossover operator to a two-dimensional chromosome, but represented in only one dimension. In the conventional crossover central points from a two-dimensional chromosome that was stretched out present a higher likelihood of being selected comparing to peripheral points. This effect has been called positional bias. UNBLOX corrects this anomaly by ensuring that all points in the chromosome have the same probability of generating cuts. The SPAUC operator to be proposed later in this paper does not present positional bias since it does not simply handle two-dimensional chromosomes by stretching them out, but as full two-dimensional entities.

Tanaka *et al*. (1993) suggested a coding scheme in a geomagnetism problem that introduced the concept of spatial distribution of electric currents. The idea of a rectangular crossover was sketched, but no further development was discussed.

Finally, Mitchell (1997) and Bäck (1996) report on several crossover implementations. The following are given: a) *segment crossover,* in which the total number of cutting points is not fixed but can vary around some expected value, perhaps following a Poisson distribution; b) *shuffle crossover,* in which the parents are shuffled in some way, the crossover made, and then the offspring unshuffled; c) *punctuated crossover,* in which each individual carries information about the number of cuts it will be subjected to, and the actual sites where they should occur (since these also belong to the genotype, their tuning is also left for the evolutionary process).

All in all, discussions about the generation of new genetic operators and their performance is far from over. In fact, as pointed out in Mitchell (1997), the related pieces of work presented so far are usually associated with small sets of test data; worse of all, different investigations give way to conflicting results. As Bäck (1993) says: "during each of the main conferences on GA a few new operators can be expected, especially when a non-traditional problem representation is used".

4 SPATIAL UNIFORM CROSSOVER

Boschetti (1996) textually says: "one of the main problems in the application of genetic algorithms to geophysical problems is the high dimensionality". This is also the case in the present application, even considering it is based on synthetic data; in fact, since each individual is made of 70 real numbers, and assuming the required precision is in the order of one hundredth, the search space has approximately 10^{140} points (with real data the situation becomes dramatically worse). A canonical evolutionary engine working on a noiseless synthetic problem was not able to present appropriate results. A search for *ad-hoc* operators was then initiated. Three were developed: local search, homogenisation, and a specific crossover operator, named SPAUC (SPAtial Uniform Crossover).

Initial experiments we performed on this problem soon suggested the better performance of the uniform crossover, comparing to 1-, 2- or n-point crossovers. So, when the necessity for presenting a new and better crossover operator came up, the new operator came naturally from the uniform crossover, through its generalisation; however, the standard uniform crossover was not disregarded completely. The choice between SPAUC and uniform crossover is given by a p probability, whose value is left for the user, remaining unchanged through the run. So, when p = 1.0, all calls to the operator are responded by SPAUC, whereas if p = 0.0, uniform crossover is fully in charge.

The SPAUC operator considers each individual as a two-dimensional entity. Accordingly, when genetic material is interchanged, full rectangular patches of the candidate solutions are involved. The operator implements the neighbourhood concept both vertically and horizontally. This action prevents two common problems that appear when a two-dimensional individual is represented in one dimension, that is, the possibility that: a) two neighbouring vertical positions in the chromosome no longer be neighbours; and b) originally distant elements get closer to each other than they should (for example, as happens when the last value on a row and the first one in the subsequent row are involved).

In order to implement the SPAUC operator, we used the following procedure: a) n horizontal cuts and k vertical cuts are generated on random sites across the individual; (amounting to r patches, $r = (n+1)\times(k+1)$); b) a V mask of size r is generated having random binary values; and c) given the two ancestors that will be operated by SPAUC, the offspring to be generated will obey the following rule: patches with a mask value equal to 0 are filled up with values from the first parent, while patches with a mask value equal to 1 take their values from the corresponding patch in the second parent. Notice that the longer the mask V, the closer SPAUC will resemble the standard uniform crossover.

What follows is an example showing the operation of SPAUC. It starts by generating random values that will indicate the cutting rows across the chromosome. The total number of cuts is user-defined. Assuming that 2 cuts have been generated in rows 3 and 4 of an individual A1, the situation would look like as follows:

$$A1 = \begin{array}{cccccccccc}
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a \\ \hline
a & a & a & a & a & a & a & a & a & a \\ \hline
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a
\end{array}$$

Next, the algorithm generates random cutting columns. Assuming 3 cuts have been generated at columns 3, 4 and 7, the situation above would become:

$$A1 = \begin{array}{ccc|c|ccc|ccc}
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a \\ \hline
a & a & a & a & a & a & a & a & a & a \\ \hline
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a \\
a & a & a & a & a & a & a & a & a & a
\end{array}$$

Notice that the cuts generated 12 distinctive patches within the individual. Consequently, the typical V mask will need to have 12 bits in order to indicate the relation between parents and patches, that is, which parent will generate which patch. Assuming the mask $V = 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0$, and assuming the existence of a second parent, named A2 (filled up with with *b* values), applying SPAUC would generate the following offspring:

$$D1 = \begin{array}{ccc|c|ccc|ccc}
b & b & b & b & a & a & a & b & b & b \\
b & b & b & b & a & a & a & b & b & b \\
b & b & b & b & a & a & a & b & b & b \\ \hline
b & b & b & a & b & b & b & a & a & a \\ \hline
b & b & b & b & b & b & b & a & a & a \\
b & b & b & b & b & b & b & a & a & a \\
b & b & b & b & b & b & b & a & a & a
\end{array}$$

$$D2 = \begin{array}{ccc|c|ccc|ccc}
a & a & a & a & b & b & b & a & a & a \\
a & a & a & a & b & b & b & a & a & a \\
a & a & a & a & b & b & b & a & a & a \\ \hline
a & a & a & b & a & a & a & b & b & b \\ \hline
a & a & a & a & a & a & a & b & b & b \\
a & a & a & a & a & a & a & b & b & b \\
a & a & a & a & a & a & a & b & b & b
\end{array}$$

5 OTHER OPERATORS

The numerical results reported here were obtained using two other operators besides SPAUC: local search and an homogenisation operator. While the local search awards discontinuities, forcing them to appear, homogenisation creates conditions for equal conductivity regions to appear. The use of the 2 operators produced good results.

5.1 LOCAL SEARCH

Local search is implemented as a simple hill-climbing algorithm. At every 5 generations the attempt is made to improve the fitness of the fittest individual through about 13 cycles, of 140 trials each (the number of cycles is an average figure, since it starts smaller and increases throughout the generations). The figure 140 derives from the fact that the chromosome is 70 points long, and each one of them is subjected to two small random increments (one positive and the other negative), that represent probing two neighbouring positions in the fitness landscape. These probed positions with higher fitness are stored until the end of the cycle, when all the changes are applied at once, thus generating a new, higher fitted individual that replaces the original.

5.2 HOMOGENISATION

This operator is based upon the notion that neighbouring regions will have greater probability of having similar conductivities. At every 5 generations an attempt is made to improve the fitness of the fittest individual through an average of 13 cycles of 3 trials each. Every trial starts with the generation of a rectangular random patch and continues, firstly, by replacing all values in the patch, for one of its values – thus generating an homogeneous patch – and checking the resulting fitness; secondly, by repeating the same procedure, individually for all the other values in the patch. The patch the results the best (improved) performance, if any, replaces the original.

Additionally to the use of homogenisation alone, this operator is also used in association with SPAUC. In this case, with small probability (5%) homogenisation is applied to the individuals generated out of SPAUC.

6 RESULTS

We used the genetic algorithm package GALLOPS (Goodman 1996), modified by the 3 operators above that were added to it. The following parameters were used: population size of 100, tournament selection of size 15, 3% mutation per conductivity value, and crossover rate of 85%.

In the experiments to follow, two comparisons have been made. First, SPAUC is compared to uniform crossover, both with noiseless and noisy synthetic data. Then, the SPAUC-based evolutionary process is compared to a classical iterative approach for the problem, that uses a sophisticated gradient-based optimiser plus (entropy-based) regularisation. The second experiment is fully performed in the presence of noise, at the same amount for both approaches.

6.1 COMPARISON BETWEEN SPAUC AND UNIFORM CROSSOVER: NOISELESS DATA

The parameters summarised in Table 1 were used in the experiment below, as they provided a fairly acceptable balance between precision and demand on computer resources. The first experiment relied on noiseless synthetic data, and aimed at a comparison between SPAUC and uniform crossover.

Table 1. Values used in the SPAUC-based run

SPAUC probability	p = 100%
Horizontal cuts	n = 2
Vertical cuts	k = 3
Total number of patches	r = 12

We compared a typical run with pure SPAUC (100% of cases) with pure uniform crossover (100% of cases). The runs went through 224 generations, all other parameters being the same. The initial population was initialised with individuals representing random conductivity values from 0.0 to 100.0. The 10.0 value stands for a background conductivity, meant to represent rock.

The problem instance at issue is depicted as Table 2; it presents a rectangular body embedded in the rock, with a conductivity of 100.0 (that is, ten times more conductive than rock). The solutions found using SPAUC and uniform crossover individually, are shown, respectively, in Tables 3 and 4.

Table 2. The known answer of the problem that the evolutionary engines should find. It represents a patch of rock (conductivity 10.0) with a rectangular block of higher conductivity material (100.0) in it.

10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	10.0	10.0	10.0
10.0	10.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Table 3. Fittest individual using uniform crossover only

10.3	9.4	11.3	8.6	13.6	1.9	41.0	2.7	26.0	3.3
8.4	16.3	**100.0**	**96.7**	**7.8**	**31.6**	**73.1**	21.6	5.1	18.2
13.0	1.8	**89.4**	**75.0**	**79.9**	**71.4**	**79.1**	9.3	3.2	99.1
8.0	18.9	33.0	90.4	1.2	1.2	12.2	10.7	7.3	28.4
9.7	10.1	4.9	5.4	2.2	13.1	12.5	10.3	10.0	5.9
10.1	10.7	2.5	3.8	39.3	2.3	4.6	10.7	10.4	3.1
7.6	15.7	13.6	1.5	70.7	1.8	13.5	8.6	8.1	13.1

Table 4. Fittest individual using SPAUC only

10.0	10.0	9.8	10.2	9.8	10.2	9.7	10.1	9.9	10.1
10.0	9.9	**98.4**	**100.0**	**100.0**	**100.0**	**94.0**	10.6	9.5	10.4
10.0	10.0	**99.9**	**100.0**	**84.5**	**100.0**	**100.0**	10.4	9.4	10.9
10.1	9.9	10.2	20.5	58.9	21.8	11.1	10.1	10.2	9.8
10.0	10.0	10.0	9.7	7.4	8.3	9.9	9.6	10.0	9.7
10.2	10.0	10.0	12.9	26.4	11.6	10.3	10.2	10.0	10.8
9.7	10.0	10.0	9.9	8.5	9.5	10.1	9.5	9.5	9.8

On a single randomly selected run a conductivity error of 860.4 was obtained when uniform crossover was used and of 132.8 when SPAUC was used. However, while the run using uniform crossover requested approximately 150×10^3 objective function calls, the one with SPAUC requested 223×10^3. Because running the process is computationally very intensive only a single run was used in this experiment; however, partial runs were performed that suggested the same trend reported herein, and also, other complete runs were performed for the other experiments (to be described below) which turned out to be coherent with the observations above.

6.2 COMPARISON BETWEEN SPAUC AND UNIFORM CROSSOVER: NOISY DATA

Noise was introduced at the magnetic field generated by the individual represented by Table 2 (the known answer of the problem). Gaussian noise was used with 0 mean and 1% standard deviation. The population was randomly initialised and three runs were performed with different random seeds using either SPAUC or uniform crossover. Table 5 presents the results.

Table 5. Comparative data using SPAUC and uniform crossover, using noisy data

Test performed	SPAUC	Uniform Crossover
Lowest conductivity error (in $4\pi\times\omega\times10^{-10}$ mhos/m)	262.5	600.6
Average conductivity error (in $4\pi\times\omega\times10^{-10}$ mhos/m)	450.3	744.2
Average number of calls to the objective function	102×10^3	60×10^3

As we can see, the SPAUC-based solution still yields better results in the presence of noise than that of the uniform crossover, even though the result is not as good as in the noiseless case.

6.3 COMPARISON BETWEEN THE PROBLEM-SPECIFIC EVOLUTIONARY METHOD WITH A CLASSICAL APPROACH

Ramos and Velho (1995) investigated the same noisy problem mentioned above using E04UCF, a powerful gradient-based optimiser of the Numerical Algorithms library (NAG 1988).

Table 6. Fittest individual in Ramos and Velho (1995)

10.02	10.25	9.64	10.09	10.03	9.58	10.15	10.07	9.97	9.94
10.02	9.64	**97.10**	**100.00**	**88.98**	**97.62**	**100.00**	10.07	9.95	9.94
9.79	9.64	**97.10**	**99.97**	**88.98**	**97.62**	**100.00**	10.07	9.95	9.93
9.78	9.63	10.08	10.02	12.38	16.85	9.56	10.06	9.95	10.54
9.78	9.62	10.08	10.03	12.38	16.85	9.57	9.97	9.95	10.54
10.25	9.63	10.08	10.02	9.58	10.16	9.57	9.97	9.94	10.55
10.25	9.63	10.09	10.02	9.58	10.15	10.09	9.97	9.94	10.55

Table 7. Fittest individual in the present study

9.89	10.17	9.72	10.34	9.44	10.51	9.73	10.24	10.25	10.19
9.62	9.85	**96.78**	**98.84**	**90.48**	**98.27**	**94.57**	10.29	9.16	10.89
10.61	10.98	**99.84**	**99.96**	**74.33**	**99.47**	**92.17**	10.43	8.78	10.58
10.26	8.83	8.63	16.27	7.06	11.48	12.91	11.19	11.29	10.20
8.82	11.75	10.60	6.95	11.81	11.81	8.85	10.90	9.73	9.45
12.86	6.99	10.98	12.41	8.28	4.22	6.95	8.22	11.74	11.15
4.20	34.73	5.26	10.29	3.23	9.74	19.85	11.25	10.44	6.43

They ran the model 4 times obtaining a conductive error average of 453.8. The fittest individual is represented in Table 6. We ran the same problem 6 times in our evolutionary, SPAUC-based approach, and obtained a conductivity error average of 210.5, with the fittest individual being represented in Table 7.

Considering that the level of noise (1%) was the same for both techniques, the problem-specific evolutionary process developed is clearly justified.

7 CONCLUDING REMARKS

Even though it may be tempting to use a canonical algorithm, that is possibly ready for use (as in Goodman (1996)) which could warrant correct results in a short time, practice has demonstrated that this approach is only appropriate for simple problems. For more complex problems – in particular, for the high dimensional inverse problems – satisfactory results have only been obtained when problem-specific knowledge is explicitly introduced into the evolutionary engine. However, as we did just that in the present approach, we tried to keep the amount of knowledge as minimal as possible in order not to forego robustness of the procedure.

In the reported case such an attempt was successful, the only pieces of knowledge used having been the facts that: individuals are two-dimensional and should be handled as such; conductivity distributions in the underground of the earth should follow homogenous patches; and that a simple hill-climbing algorithm could well be used to help improve the results. Contrast the first premise with Bäumer (1996), where an evolutionary approach to magnetotelluric inversion is also used, but with a more limited, one-dimensional approach.

Another point to be remarked is that the evolutionary engine presented better results than a conventional optimiser, although at higher cost. While the best result from Ramos and Velho (1995) establishes an approximate amount of 14×10^3 objective function calls, our best result shown here requested approximately 100×10^3 calls. For the case we reported on, which relies on synthetic data, both approaches have an undeniable merit.

However, on real-world data, where accuracy is the critical concern, the evolutionary technique presented here is certainly an appealing alternative; another possibility, as concluded by Bäumer (1996), is using evolutionary computation in order to find the best region of the search space, and then performing an efficient local search so as to find the best point of that region.

Acknowledgements

The authors are grateful for financial support from FAPESP (Proc. 96/7200-8) and the British Council (Ref SPA/126/881). Additionally, PLKGN and PPBO thank the Brazilian Council for Scientific and Technological Development, respectively, for a scholarship (Nº 141124/94-6) and a research grant (PQ Nº 300.465/95-5).

References

Bäck, T. 1996, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press.

Bäumer, O. 1996, "Inverting Magnetotelluric Data Using Genetic Algorithms and Simulated Annealing". *Inverse Methods: interdisciplinary elements of methodology, computation and applications*. Springer-Verlag.

Boschetti F.; Dentith, M.C. and List, R.D. 1996, "Inversion of Seismic Refraction Data using Genetic Algorithms". *Geophysics*. 61(6):1715-1727.

Bruns, R. 1993, "Direct Chromosome Representation and advanced Genetic Operators for Production Scheduling". In: *Proceedings of fifth International Conference on Genetic Algorithms*. Illinois.

Cartwright, H.M. and Harris, S.P. 1993, "Analysis of the Distribution of Airborne Pollution using Genetic Algorithms". *Atmospheric Environment*. 27A(12):1783-1791.

Davis, L. 1991, *The Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Forrest, S. and Mitchell, M. 1992, "Relative Buildind-Block Fitness and the Building-Block Hypothesis". *Foundations of Genetic Algorithms 2*, D. Whitley (ed.). Morgan Kaufmann.

Goldberg, D.E. 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison Wesley Publishing Company.

Goodman, E.D. 1996, *The Genetic Algorithm Optimised for Portability and Parallelism System*. Michigan State University.

Guerreiro, J.N.C.; Barbosa, H.J.C.; Garcia, E.L.M.; Loula, A.F.D. and Malta, S.M.C. 1998, "Identification of Reservoir Heterogeneities Using Tracer Breakthrough Profiles and Genetic Algorithms". *SPE Reservoir Evaluation and Engineering*, June 1998.

Kajiwara, I. and Nagamatsu, A. 1996, *Structural Topology Optimisation by Genetic Algorithm (Approaches for Improvement of Calculation Efficiency)*. Technical Report.

Kodiyalam, S.; Naggendra, S. and DeStefano, J. 1996, "Composite Sandwich Structure Optimisation with Application to Satellite Components". *AIAA Journal*. 54:613-622.

Mitchell, M. 1997. *An Introduction to Genetic Algorithms*. MIT Press.

NAG. "E04UCF Routine", 1988, *NAG Fortran Library Mark 13*, Oxford, UK.

Ramos, F.M. and Velho, H.F.C. 1995, *Estimation of the Earth Condutivity by a Maximum Entropy Reconstruction Technique*. Technical Report, Intituto Nacional de Pesquisas Espaciais, Brazil.

Sabatier, P. 1985, "Inverse Problems - an Introduction". *Inverse Problems*. 1:1-4, 1985

Tanaka, Y.; Ishiguro, A. and Uchikawa, Y. 1993, "A Genetic Algorithms Application to Inverse Problems in Electromagnetics". In: *Proceedings of fifth International Conference on Genetic Algorithms*. Illinois.

Woodbury, K. 1996, *What are Inverse Problems*. Technical Report. University of Alabama.