

## **SOLVING THE GEOPHYSICAL INVERSION PROBLEM USING GENETIC ALGORITHMS**

Vesna [E[UM

Department of Mathematics, Faculty of Mechanical Engineering  
University of Belgrade, Belgrade, Yugoslavia

Jozef KRATICA

Mathematical Institute, Serbian Academy of Sciences and Arts  
Belgrade, Yugoslavia

Du{an TO[I]

Faculty of Mathematics  
University of Belgrade, Belgrade, Yugoslavia

**Abstract:** Tomographic methods are often used to solve the geophysical inversion problem. This is the problem of determining the characteristics of an underground region by using measurement data. With a tomographic method, the pictures of the physical properties in the region between pairs of boreholes can be reconstructed. In this paper, we apply genetic algorithms to cross-hole seismic line-integral data. These data can be generated by the transmission problem through the cross-hole region using seismic energy. Some test-examples for measuring the performance of the applied genetic algorithms are presented. The obtained results are promising.

**Keywords:** Genetic algorithm, tomography, geophysical inversion problem.

### **1. INTRODUCTION**

In the last two decades, tomographic reconstruction techniques have had intensive development. Because of that, these techniques have become widely used both in medical imaging and in geophysical imaging applications. The application of geophysical tomography is widespread. It is used in many industrial domains such as construction, mining, gas and oil exploration, thermal energy exploration, etc. (De Franco and Cavagna [5], Dyke and Young [7], Lytke et al. [11], [12], [13], Mathisen et al. [14], Wong et al. [20], Yamamoto et al. [21]).

In this paper, we propose a method for solving the geophysical inverse problem using genetic algorithms (GA). The proposed method was tested on a lot of examples and some of them are presented here.

Various approaches have been used to solve the geophysical inverse problem. Some of them are: Matrix inversion, the Fourier transform, the convolutional method and the Algebraic Reconstruction Technique (ART). The most often used method is ART (Dynes and Lytle [6]). ART poses the integral equation of ray-traveltime in matrix form. The equations are solved using relaxation techniques taking advantage of the sparsity of the obtained matrix. The results obtained by GA in this paper are compared with results obtained by ART.

### 1.1. Problem formulation

The problem of seismic P-wave velocity estimation from first-arrival traveltimes in a two-dimensional cross-hole geometry is considered. The transmitters are located in one borehole, while the receivers are in another borehole. The rectangular area between two boreholes - some region of space through which the rays have passed - is the examined cross-hole region. A straight-line ray model is used for energy propagation between the boreholes. The velocity distribution is used to characterize the subsurface.

Standard tomographic procedure is based on discretization of the cross-hole region (Dines and Lytle [6], Peterson et al. [15]). The rectangular cross-hole area is divided into a grid of cells and velocity  $v(x, y)$  is assumed to be constant over the area covered by any one cell. (See Fig.1)

Traveltime is related to the velocity structure by the following equation

$$t_k = \int_{R_k} \frac{ds}{v(x, y)} \quad (1)$$

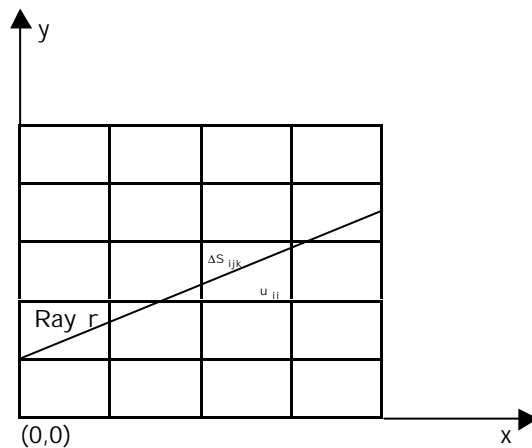
where  $t_k$  is the traveltime of the  $k$ th ray,  $ds$  is the differential raypath length of the  $k$ th ray,  $v(x, y)$  is the two-dimensional velocity function and  $R_k$  is the raypath trajectory of the  $k$ th ray. Reciprocal value  $1/v(x, y)$  designate slowness and is denoted by  $u(x, y)$ . Then Eq. (1) becomes

$$t_k = \int_{R_k} u(x, y) ds \quad (2)$$

and we will look for slowness distribution in the cross-hole area. When the cross-probing area is discretized into  $m \times n$  cells, Eq. (2) can be approximated as

$$t_k = \sum_{i=1}^m \sum_{j=1}^n u_{ij} \Delta S_{ijk} \quad (3)$$

where  $u_{ij}$  is the unknown slowness in the  $(i, j)$ -cell,  $\Delta s_{ijk}$  is the length of the ray segment that intercepts the  $(i, j)$ -cell,  $m$  is the number of vertical cells and  $n$  is the number of horizontal cells. It is understood that  $\Delta s_{ijk} = 0$  for those  $i$  and  $j$  values for which the associated cell is not intercepted by the ray  $R_k$ . For a more detailed explanation see: Dines and Lytle [6], Ivanson [9], Peterson et al. [15].



**Figure 1:** Discrete model of the region between boreholes

## 1.2. Genetic algorithms

Genetic algorithms are based on the model of biological evaluation. These algorithms operate on a population of individuals, which are potential solutions of an optimization problem. Genetic operators, selection, crossover and mutation, are applied to successive populations to create a new population as a better region of the search space. The initial population is randomly determined.

Based on each individual's fitness value that reflects how good an individual is in the population, selection provides a mechanism for better individuals to survive. Individuals with a higher fitness value have a higher probability of contributing offspring in the next generation. Crossover takes parental individuals and mixes their information for forming their descendants.

Mutation is sporadic and random alternation of genes with some probability and it regenerates lost genetic material.

For more detailed information about GA see: Alexouda and Paparrizos [1], Beasley et al. [3], Goldberg [8], Ribeiro-Filho et al. [16], Srinivas and Patnaik [18].

## 2. GA IMPLEMENTATION

### 2.1. Representation

Fundamental to the GA structure is the encoding mechanism for representing the optimization problem's variables. Every discretized cross-hole area (i.e. every individual in the population) has its genetic code. The genes that represent velocities distributed over the cells in the cross-hole region are obtained by extraction from the individual's genetic code. These genes have the same fixed length and in our case it is 5 bits. The number of bits for velocity representation has to be given in advance. In conformity with that, we have obtained the velocity value over a cell as

$$v = \min v + dv * tmp, \quad (4)$$

where  $dv$  is the velocity partition,

$$dv = (\max v - \min v) / 2^{nbit} \quad (5)$$

$\min v$  is the minimum velocity,  $\max v$  is the maximum velocity,  $tmp$  is the value of the binary string that represents the velocity over a cell and  $nbit$  is given in advance.

**Example.** For  $nbit = 5$  there are 32 values for velocity. Let  $\min v = 2.0$  km/s and  $\max v = 5.2$  km/s, then  $dv = 0.1$ . If the binary string, that represents velocity over some cell is 10110 then  $tmp = 22$  and  $v = 4.2$  km/s in that cell.

### 2.2. Objective value function

The minimal traveltimes are computed for every ray by using the known velocity distribution. After that, the cross-hole area is reconstructed to get a new velocity distribution. We compute

$$d = \sqrt{\frac{\sum_{i=1}^m \sum_{j=1}^n (t_{ij}^T - t_{ij}^C)^2}{k^2}} \quad (6)$$

where  $t_{ij}^T$  is the true traveltime,  $t_{ij}^C$  is the computed traveltime and  $k$  is the number of sensors.

Dynamic allocation of the following arrays provides program implementation. For every ray, there is an array that holds:

- the index  $i$  of the cell  $(i, j)$  cutting that ray
- the index  $j$  of the cell  $(i, j)$  cutting that ray and

- the length of the ray segment in the cell cutting that ray.

These values are obtained by finding the cutting points between rays and edges of cells, which are calculated only once for every ray, written in arrays and later used for computations of traveltimes from transmitters to receivers.

This approach provides good execution time, but also requires more available memory space.

### 2.3. GA operators

GA implementation, presented in this paper, performs rank-based selection, one-point crossover and simple mutation.

Rank-based selection prevents premature convergence and the loss of genetic material diversity. It is allowed to set an arbitrary nonnegative sequence of rank values. Using an experiment, the most favorable rank sequence can be determined for every particular problem. The fitness function of an individual is the rank value of that individual and depends only on the ordering of individuals in the population. (See Whitley [19]). In our rank-based selection scheme, rank decreased linearly from  $r_{best} = 2.5$  for the best individual to  $r_{worst} = 1.01$  for the worst individual, by steps of 0.01.

In our implementation, the crossover rate is  $p_{cross} = 0.85$ , and the mutation rate  $p_{mut}$  is given in Table 1. The simple mutation operator is based on using a normal distribution. This implies faster execution, because only muted genes are processed. The number of mutations is generated by a random pick in  $N(m, s)$  distribution, where

$$m = (n_{pop} - n_{elite}) \cdot p_{mut}, \tag{7}$$

$$s^2 = (n_{pop} - n_{elite}) \cdot p_{mut} \cdot (1 - p_{mut}), \tag{8}$$

and  $m$  is the average number of mutations in the population,  $s$  is the standard deviation,  $n_{pop}$  is the population size, while  $n_{elite}$  is the number of elitist individuals. After that, the mutation positions are randomly generated.

**Table 1:** Mutation rate

Test-example names	$p_{mut}$
A	0.03
B	0.006
C	0.002
D	0.001
E	0.0055

The mutation rate, given in Table 1, is a good trade-off between two contradictory requirements: exploration and exploitation.

## 2.4. Caching

In this implementation, GA is cached, which is a method for improving the performance of GA. The idea behind such a technique is the repetition of the same genetic strings while running GA. Most often, it is more suitable to remember the given genetic code and its value in the first appearance. Later, in the next appearances, the values are taken directly from the double hash table, instead of repeated calculation. Caching does not influence the accuracy of the results obtained by GA itself, it only serves to improve the GA running time.

In order to successfully cache GA, the same strings in the population should have frequent occurrences over the generations and the objective value function should have a large evaluation time. The Least Recently Used caching strategy is implemented through the double hash table. For detailed information about caching GA see Kratica [10].

## 2.5. GA parameters

We tested our program on a population of over 150 individuals. It was implemented with the steady-state replacement of generations with the elitist strategy, which means that crossover and mutation are applied only on the replaced portion of the population. In every generation, 1/3 of the population (50 worst ranked individuals) was replaced by new individuals, while 2/3 of the population (100 elitist individuals) remained from the previous generation. To prevent the domination of elitist individuals over the population, their fitness was decreased by the formula:

$$\text{new } f_i = \begin{cases} f_i - \hat{f}, & f_i > \hat{f} \\ 0, & f_i \leq \hat{f} \end{cases} \quad 1 \leq i \leq \text{nelit} \quad (9)$$

where

$$\hat{f} = \frac{1}{n_{\text{pop}}} \cdot \sum_{i=1}^{n_{\text{pop}}} f_i \quad (10)$$

is the average fitness in the entire population.

In every generation, using a method that sets their fitness to zero discards duplicate strings from the population. In this way, the occurrence of duplicate strings in the next generation is eliminated. This technique maintains the diversity of the genetic material, decreases the appearance of dominating individuals in the population and decreases the possibility of premature convergence to the local optimum.

Maximal number of generations was 5000. If the best individual in the population did not change in 1000 generations, execution stopped and results were printed.

### 3. EXPERIMENTAL RESULTS

The described genetic algorithm is written in C programming language. The testing was done using a PC compatible computer AMD 80486 at 120MHz with 8 MB of memory.

#### 3.1. Test-examples

We generated synthetic examples where the transmitter's position, the receiver's position and the initial velocity distributions in the cross-probing area are known. Test examples are grouped according to the refinement of the cross-hole area discretization.

Table 2 contains information about test-example names, the number of vertical (M) and horizontal (N) cells in the cross-hole area, the number of transmitters (K) in one borehole, the distance between boreholes and depth of the boreholes. In our test-examples, the number of receivers, which are located in the other borehole, is equal to the number of transmitters, and the number of rays is  $K^2$ . Since genetic operators are nondeterministic, every test example was run 10 times and we computed the average value.

**Table 2:** Test-examples

Test-example names	M × N	K	Borehole distance (m)	Borehole depth (m)
A1	5 × 5	20	100	100
A2	5 × 5	30	80	60
A3	5 × 5	10	80	60
B1	10 × 10	20	100	100
B2	10 × 10	30	80	60
B3	10 × 10	10	80	60
C1	15 × 15	20	100	100
C2	15 × 15	30	80	60
C3	15 × 15	10	80	60
D1	20 × 20	20	100	100
D2	20 × 20	30	80	60
D3	20 × 20	10	80	60
E	8 × 14	15	80	60

### 3.2. Computational results

The results of running the genetic algorithm are summarized in Table 3-4. For better comprehension of the results given in Table 4 and for their illustration, Table 3 presents the comparative results obtained by applying GA and the algebraic reconstruction technique on test-example E for different velocity distributions in the cross-hole area. The columns in Table 3 contain: name of the test-example, GA results (average objective function value) and ART results.

**Table 3:** Comparative presentation of some results

Test-example name	GA results	ART results
E	0.1231	0.1773
	0.1529	0.1661

The columns in Table 4 contain: name of the test-example, average objective function value, average execution time in seconds and average number of generations.

**Table 4:** Results of running GA on test-examples A, B, C and D

Test-example names	Avg. obj. func. value	Avg. time (s)	Avg. num. Of generations
A1	0.185353	960.49	4054
A2	0.142332	1932.62	3824
A3	0.100935	297.52	4129
B1	0.186264	2050.64	5000
B2	0.134311	4367.56	5000
B3	0.112137	569.88	5000
C1	0.233984	2864.02	5000
C2	0.188705	6051.03	5000
C3	0.111216	801.75	5000
D1	0.227622	3836.98	5000
D2	0.186303	8108.10	5000
D3	0.065027	1041.87	5000

As can be seen from Table 4, which contains the test results of our implementation on synthetic test-examples, GA gives quite good results. With an increase in the problem size, the quality of the solution (i.e. the objective function value) does not decrease much, while the execution time increases quadratically because the number of rays is  $K^2$ .



## 4. CONCLUSION

We have presented a new method for solving the geophysical inverse problem. It seems to be the first GA implementation of this problem. Our version gave promising and satisfactory results. The chosen genetic operators and individual representations have provided a good performance, improved by using caching GA.

Different methods have been used to solve the geophysical inversion problem (Peterson et al. [15], Brzostowski and McMechan [4], Singh and Singh [17], etc.). If there are many local optimums, these methods find one of them, while reaching the global optimum can be omitted. The advantage of the described method is the attempt to reach the global optimum.

The research presented in this paper can be continued and extended in several directions:

- Hybridization of GA with other methods.
- Testing the presented approach on test-examples of a large size.
- Parallelization of GA implementation to distributed and multiprocessor systems.

## REFERENCES

- [1] Alexouda, G., and Paparrizos, K., "A genetic algorithm approach to the Buyers' welfare problem of product line design: a comparative computational study", *Yugoslav Journal of Operations Research*, 9 (2) (1999) 223-234.
- [2] Beasley, D., Bull, D.R., and Martin, R.R., "An overview of genetic algorithms: part 1, Fundamentals", *University Computing*, 15 (2) (1993) 58-69.
- [3] Beasley, D., Bull, D.R., and Martin, R.R., "An overview of genetic algorithms: part 2, Fundamentals", *University Computing*, 15 (4) (1993) 170-181.
- [4] Brzostowski, M.A., and McMechan, G.A., "3-D tomographic imaging of near-surface seismic velocity and attenuation", *Geophysics*, 57 (30) (1992) 396-403.
- [5] De Franco, R., and Cavagna, B., *Control of Grouted Tunnels Through Seismic Topography, Gallerie e grandi pere otterrance*, N.47, 1995.
- [6] Dines, K.A., and Lytle, R.J., "Computerized geophysical tomography", *Proc. of IEEE*, 67 (7) (1979) 1065-1073.
- [7] Dyke, V.A., and Young, P.R., "Physical characterization of rock masses using borehole methods", *Geophysics*, 50 (1985) 2530-2541.
- [8] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Pub. Comp., Reading, 1989.
- [9] Ivanson, S., "Seismic borehole tomography - theory and computational methods", *Proc. of IEEE*, 74 (2) (1986) 328-338.
- [10] Kratica, J., "Improving performances of the genetic algorithm by caching", *Computers and Artificial Intelligence*, 18 (3) (1999) 271-283.
- [11] Lytle, R.J., Lager, D.L., Laine, E.F., and Salisbury, J.D., "Monitoring fluid flow by using high-frequency electromagnetic probing", *Lawrence Livermore Lab. rep. UCRL-51979 (ERA 1:17178, available from National Technical Inform. Serv., Springfield, V.A.), 1979.*

- [12] Lytle, R.J., Laine, E.F., Lager, D.L., and Okada, J.T., "Determination of the in situ high frequency electrical properties of permafrost rock", *Radio Science*, 11 (4) (1976) 285-294.
- [13] Lytle, R.J., Dines, K.A., Laine, E.F., and Lager, D.L., "Electromagnetic cross-borehole survey of a site proposed for a future urban transit station", Lawrence Livermore Lab. rep. UCRL-52484 (available from National Technical Inform. Serv., Springfield, V.A.), 1978.
- [14] Mathisen, E.M., Vasiliou, A.A., Cunningam, P., Shaw, J., Justice, H.J., and Guinzy, J.N., "Time-lapse crosswell seismic tomogram interpretation: implications for heavy oil reservoir characterization, thermal recovery process monitoring and tomographic imaging technology", *Geophysics*, 60 (1995) 631-650.
- [15] Peterson, J.E., Paulsson, B.N.P., and McEvelly, T.V., "Application of algebraic reconstruction techniques to crosshole seismic data", *Geophysics*, 50 (10) (1985) 1566-1580.
- [16] Ribeiro-Filho, J.L., Treleaven, P.C., and Alippi, C., "Genetic algorithm programming environments", *IEEE Computer*, (1994) 28-43.
- [17] Singh, R.P., and Singh, Y.P., "RAYPT - A new inversion technique for geotomographic data", *Geophysics*, 56 (8) (1991) 1215-1227.
- [18] Srinivas, M., and Patnaik, L.M., "Genetic algorithms: a survey", *IEEE Computer*, (1994) 17-26.
- [19] Whitley, D., "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best", *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, 116-123.
- [20] Wong, J., Hurley, P., and West, G.F., "Crosshole seismology and seismic imaging in crystalline rocks", *Geophysics Research*, 10 (1983) 686-689.
- [21] Yamamoto, T., Nyc, T., and Kuru, M., "Porosity, permeability, shear strength: crosswell tomography below an iron foundry", *Geophysics*, 59 (1994) 1530-1541.