GALIS: A Genetic Algorithm for Inverse Scattering

Simone Genovesi, Emanuele Salerno

CNR - Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo"

Pisa, Area di Ricerca CNR di San Cataldo

email: simone.genovesi@isti.cnr.it

Abstract

This technical note is devoted to a new genetic algorithm developed in order to solve electromagnetic problems concerning the inverse scattering for one-dimensional permittivity range profile reconstruction. In the introduction, we shortly present the inverse scattering problem, the different approaches commonly adopted to face its mathematical intrinsic ill-posedness and our model for the interaction between plane waves and inhomogeneous media. Then, in the second part, we address the main properties and theoretical aspects of genetic algorithms and the operators involved in the evolutionary process. Finally, in the last section, we focus our attention on describing in detail the implementation in Fortran code of the mentioned operators and entities, the particular genetic strategies used and the results obtained.

1. Introduction

1.1 Inverse scattering for permittivity range profile reconstruction

In inverse scattering an unknown scattering object is probed by known waves with the aim at estimating the object structure, from measurements of the scattered field. Potentially, the use of electromagnetic waves at microwave frequencies for non invasive diagnostics has a very wide range of applications in civil and industrial engineering, nondestructive testing, geophysical prospecting, biomedical engineering. In spite of their great importance, the available microwave tomographic techniques, are still insufficient for practical applications. In fact, there are theoretical aspects hard to manage, especially the intrinsically nonlinear relationship between the scattered field and the dielectric constant of the object investigated. The first approaches to the inversion of a scalar scattering operator, in the late 1960s and in 1970s, relied on the linearization of the scattering equations (Born or Rytov approximations) which is feasible when the maximum dielectric contrast of the object to be imaged is not too large and the object itself is not too large if compared to the wavelength of the probing radiation [1]. This type of approaches leads to simple Fourier relationships, which, however, are not so useful for practical applications, since the materials and objects suitable to be tested are very few. Still retaining the linearized model, generic prior information can be exploited to derive nonlinear algorithms based on analytic properties of the object functions to be reconstructed [2] [3]. By this approach, it is possible to extend the applicability of microwave imaging to a larger class of objects of interest [4] [5] [6]. The inversion has also been attempted by approximating the complete scattering operator to orders higher than 1 [7] [8].

Of course, a fully nonlinear data model would allow very accurate algorithms to be derived for quantitative electromagnetic tomography. This approach has been followed since the late 1970s [9], with reconstruction algorithms derived from both a functional analytic [10] and a numerical viewpoints [11]. Unfortunately, these algorithms were unfeasible too, because of a number of reasons. First, the nonlinear inverse scattering problem suffers of non uniqueness of solution, and thus some explicit limitation to the set of feasible solutions must be introduced. This means that all the solutions proposed, if only based on fitness of the model to the data, are possibly very far from the true function to be reconstructed, and, in any case, they are very sensitive to even extremely small amounts of measurement noise. The simple Tikhonov regularization approach [12] cannot completely solve the problem, since it almost always leads to over-smoothed solutions. On the other hand, all the proposed algorithms were very expensive computationally, and the computing hardware available in the1980s and part of 1990s was not powerful enough to manage the required calculations for any problem of practical interest. In the last few years, the situation is being changed. Under the regularization point of view, some authors are considering local smoothness constraints, in order to let the contrast discontinuities to be preserved [13]. Under the algorithmic point of view, use of advanced optimization strategies and of new computational paradigms have been made since the 1990s [14] [15] [16]. Among the latter, evolutionary computing has been found to be suitable to efficiently find a global optimum in a fitness landscape, and its use has also been proposed for inverse scattering.

In this report, we present a method for inverse scattering that adopts a genetic algorithm. Its objective is the reconstruction of the permittivity range profile of a layered medium from multifrequency backscattering data. The inversion strategy is to optimize a fitness landscape augmented by an edge-preserving regularization term [17] [18], in order to stabilize the solution. A GA-based FORTRAN code, which implements this strategy, will be presented.

1.2 The direct problem

A lossless dielectric wall of thickness L, immersed in air (see Figure 1.1), is probed by a microwave plane-wave radiation with normal incidence. The addressed problem is one-dimensional, because the wall is supposed homogenous along the transverse planes and its dielectric properties only depend on the depth coordinate.

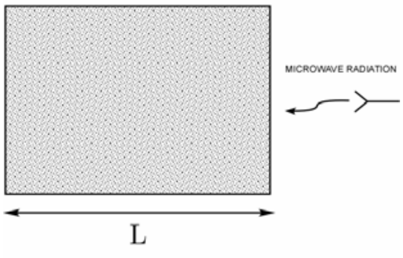

Figure 1.1 Measurement setup

The aim consists in finding an estimate of the permittivity as a function of position, on the basis of the measured values of the complex reflection coefficients at several frequencies. In order to deal with this problem, the wall is discretized into a finite number of homogenous layers (see Figure 1.2).

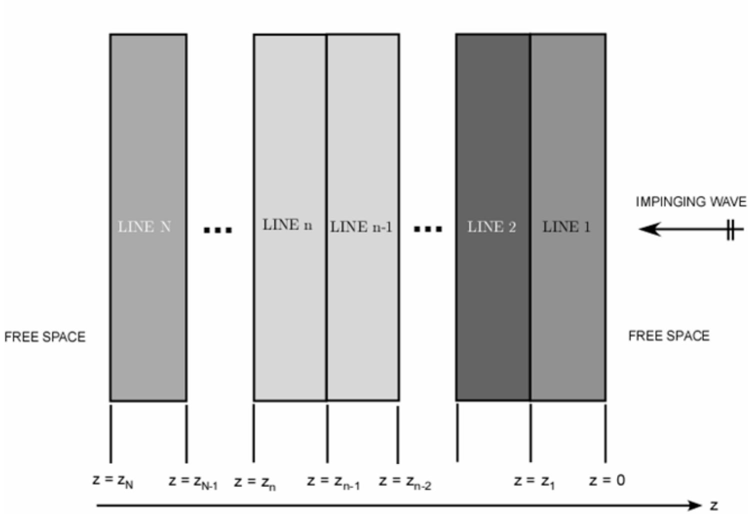

Figure 1.2 Discretized model of the wall

The next step is to consider each layer as a transmission line and to treat the entire structure as a cascade of transmission line segments (see fig. 1.3).

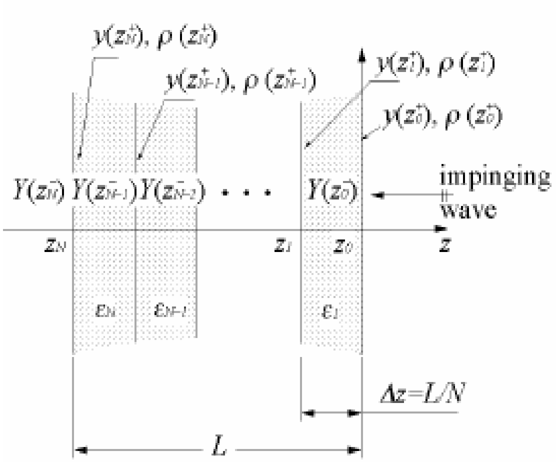

Figure 1.3 Description of the equivalent line

The reflection coefficients are found by evaluating the admittance of the entire equivalent line seen by the wave impinging at the interface z=0. The incident electric field is represented by $E^{inc} = E_0 e^{j\beta_0 z}$, where β_0 is the propagation constant in free space related to a generic frequency. We designate the characteristic admittance of the *n-th* transmission line segment as $\omega\varepsilon_n / \beta_n$, where ε_n is the permittivity of the layer at the specified frequency. We initialize the admittance $Y\left(z_N^-\right)$

seen at the interface z=z_N towards the negative z axis at the value of the characteristic admittance of free space:

$$Y\left(z_N^-\right)=\omega\varepsilon_0/\beta_0 \tag{1.1}$$

Then, for each interface at z_n, until z_n=0, we repeat the following steps:

- Compute the normalized admittance seen through the interface towards the negative z axis

$$y\left(z_n^+\right)=Y\left(z_n^-\right)\Big/\omega\varepsilon_n/\beta_n \tag{1.2}$$

- Compute the reflection coefficient $\rho\left(z_n^+\right)$ at the interface z=z_n

$$\rho\left(z_n^+\right)=\left[1-y\left(z_n^+\right)\right]/\left[1+y\left(z_n^+\right)\right] \tag{1.3}$$

- Rotate the reflection coefficient to the next interface

$$\rho\left(z_{n-1}^-\right)=\rho\left(z_n^+\right)e^{-j\beta_n\left(z_{n-1}-z_n\right)} \tag{1.4}$$

- Compute the unnormalized admittance at the next interface

$$Y\left(z_{n-1}^-\right)=\omega\varepsilon_n\left[1-\rho\left(z_{n-1}^-\right)\right]/\left\{\beta_n\left[1+\rho\left(z_{n-1}^-\right)\right]\right\} \tag{1.5}$$

At the end of this process, we obtain an array containing all the complex reflection coefficients of the current range profile estimate. Let us call this complex vector $\rho_{calc}\left(\varepsilon\right)$. For ε to be a good estimate of the actual permittivity profile, it should produce a good match between $\rho_{calc}\left(\varepsilon\right)$ and the vector of the measured reflection coefficients. Unfortunately, a good match between synthetic and measured data is never enough to find a robust solution to an inverse problem. In the following, we will present a brief description of a genetic strategy to solve this problem and some details of our particular implementation.

2. An Overview of Genetic Algorithms

2.1 Introduction

Genetic Algorithms (*GAs*) are global stochastic search methods that mimic the metaphor of biological evolution. They are patterned after the natural processes of genetic heredity and variation in organisms. GAs operate on a set of potential solutions (*population*) employing the Darwinian principle of survival of the fittest to produce approximations to a solution. These approximations (*individuals*) are encoded as strings (*chromosomes*) using various alphabets (binary, integer, real). The chromosome values represent *genotypes* and are mapped onto the decision variable (*phenotypic*) domain, but the search process is carried out on the encoded variables rather than on the decision variables. In order to asses the performance (*fitness*) of members of a population, chromosomes are decoded to obtain a representation into the solution space. The goodness of a chromosome is evaluated through an objective function that gives a value proportional to the individual's aptitude for the current problem. In biological terms, this number is proportional to the capability to survive in a given environment. A new set of approximations is created at every generation by selecting individuals according to a selection rule (proportionate, tournament) and breeding a new generation by using the *cross-over* operator.

This process drives towards individuals better than the elder ones just as in natural adaptation.

2.2 Global and local optimization methods

Many authors [19]-[21] have shown that GAs can be used to solve a large variety of problems and to find innovative solutions in many application fields [22],[23].

GAs achieve their best results in exploring multi-dimensional and multi-modal function domains in order to reach an approximate global optimum. Like all global optimization methods, GAs are largely independent of the initial conditions and place few constraints on the solution domain. Thanks to these characteristics, GAs are able to deal with solution spaces having discontinuities and a large number of dimensions with many potential local optima. However, they don’t benefit of local solution-space characteristics and this brings in general to a slower rate of convergence than the local techniques.

2.3 Genetic entities and operators

The evolutionary process is shown in the flow chart in Fig. 2.2.

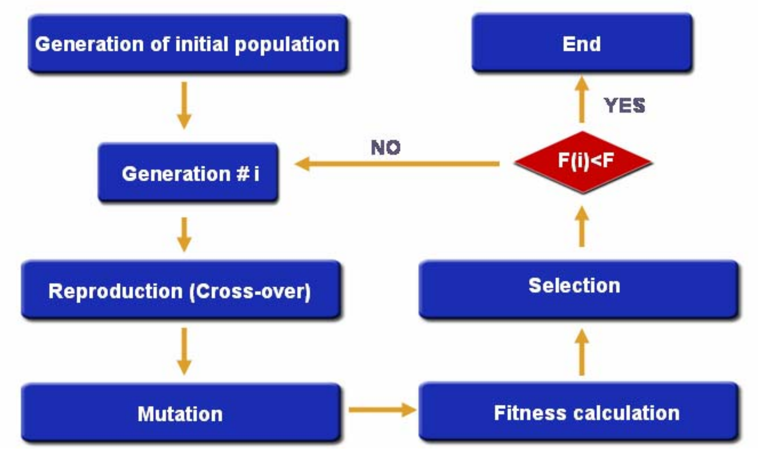

Fig. 2.2 Flow chart of a simple Genetic Algorithm

The formalization of the process described is due to Holland [24] and De Jong [25] and involves the definition of entities like:

- Gene
- Chromosome
- Population
- Generation

and the action of distinct operators:

- Cross over
- Mutation
- Fitness function
- Selection

which we are going to describe in the next sections.

Gene

Gene is the fundamental building element of genetic algorithms. Genes are used to encode the solution parameters with the most suitable codification. The most common representation for genes is the binary alphabet (0,1) although others can be used, e. g. integer, real-valued.

Chromosome

A chromosome represents a trial solution of the real problem we want to solve. The set of genes which are used to map the set of parameters of the physical problem into coded and finite length strings is the chromosome. The number of chromosomes for each step in the evolution process is constant and forms the population at current state.

Population

A population is a set of chromosomes which contains all the proposed solutions generated by the algorithm

Generation

During the first part of the algorithm, as we will explain, a set of many populations of chromosomes are created in order to sweep the widest area of the solutions space. A set of populations is called a generation. Then, depending on the chosen algorithm, after this first stage, we can use a generation (so a set of population) or just a single population in the evolutionary process. In the latter case the two terms “generation” and “population” mean the same thing since any generation is composed of a single population.

Cross-over

The cross-over (or recombination) is the basic operator for producing new chromosomes. It takes two chromosomes (parents) and generates two new ones (children); in this way, new individuals have some parts of both parents’ genetic material. Many types of cross-over have been developed: single and multi point (Fig. 2.3), uniform, shuffled.

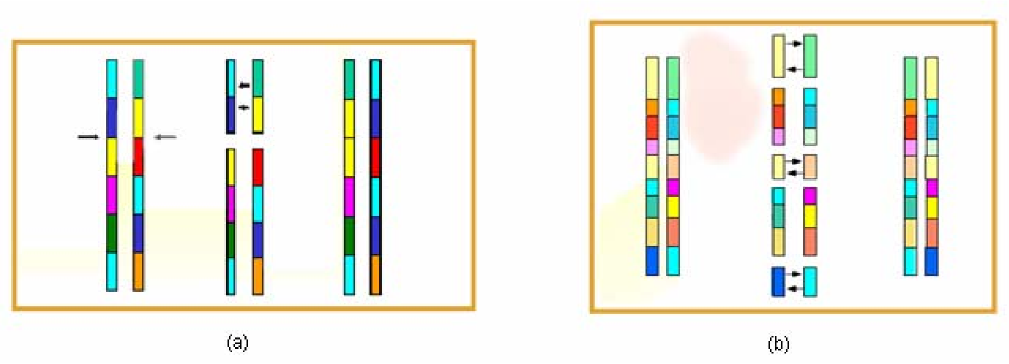

Fig. 2.3 (a) Cross-over single point, (b) Cross-over multipoint

Cross-over takes place with a probability p_{cross} and, in the case of single point, a random location in the two parent's chromosomes is selected and subsequently the two first portions of the two chromosomes are exchanged to create children. The case of multipoint is a simple extension of this with more cross points. The aim is to produce more fit individuals thanks to the exploration of the search space that cross-over encourages.

Mutation

The mutation operator guarantees an exhaustive possibility of exploration all over the whole solution surface. Mutation has a probability *pmut* to occur and corresponds to the change of a random selected element in the string in which we have encoded the information inside the chromosome (Fig. 2.4). In the case of binary coding, the selected bit in the chromosome is inverted (a "1" becomes a "0" and vice versa).

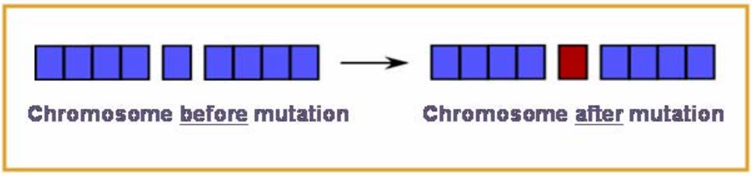

Fig. 2.4 The mutation of a chromosome

In some new implementations *pmut* dynamically varies. It starts from a default value and, if from a generation to the next one we don't find a better value of the best fitness we add a fixed *delta-pmut* to *pmut*. This increment goes on until we reach a maximum value or when the fitness of the best chromosome of the population becomes better than that of the previous generation. In this second case, we bring back *pmut* to the default value.

Fitness function

Fitness function is the core of the entire GA since it is the only connection with the physical problem to be optimized. The fitness function calculates the goodness of a trial solution extracting the physical values proposed by the algorithm from the genes codified in the chromosome.

In most existing literature [26] [27] the fitness function is non negative definite, the optimum being its minimum value. Generally, the process referred as “fitness optimization” is thus a minimization procedure.

Selection

The optimization process takes place under the influence of a selection process , which sets in action the indication given by the fitness function. A great number of selection strategies, both stochastic and deterministic, have been proposed for the GA optimization. A very common deterministic strategy is the population decimation. In this case, the population is ranked according to the fitness values of the chromosomes.

Then an arbitrary fitness threshold is chosen and each individual with a worse fitness is discarded. This simple strategy, which allows fast convergence, has a serious drawback consisting in the elimination of chromosomes that could have positive genetic characteristics but not a good fitness, which are prematurely lost. To prevent this fault, a stochastic strategy, sometimes called “roulette wheel selection” is implemented (Fig. 2.5). It consists in a selection which ensures even unfit chromosomes to have a finite, though small, probability of selection. Its value is:

$$p_{sel}(i) = 1 - \frac{fitness(parent_i)}{\sum_m fitness(parent_m)} \tag{2.1}$$

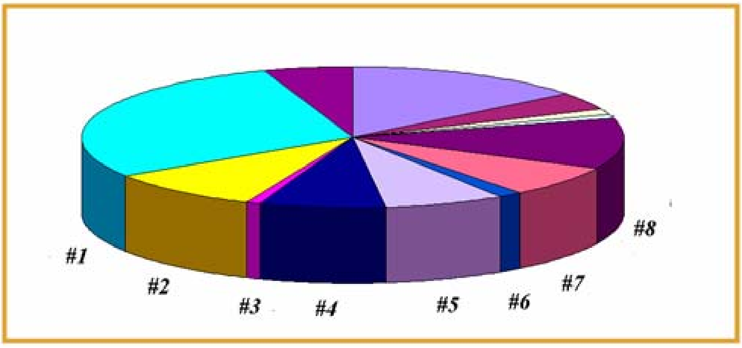

Fig. 2.5 Spaces on the wheel are proportional to each individual's relative fitness

There is a finite probability that highly unfit individuals will participate and “survive” in the next generation giving their support to the evolutionary process.

To better understand the role of each entity described before and its meaning, we want to recall the fig. 2.2 and briefly describe step by step the flow of evolutionary process. During the generation of the initial population, a prefixed number of populations are randomly created. For example, if we assume that each population is formed by 20 chromosomes and that the prefixed number of populations is 40, we have 800 chromosomes at the end of this first step. We want to underline that, with the assumption that a population is formed by 20 chromosomes, and being constant the number of chromosome in a population, the number of chromosomes involved in each step of the process after the generation of the initial population will be 20. Among these 800 chromosomes, we choose the best 20 to create the first generation (which is, in this example, coincident with a population). The second generation is created selecting couples of chromosomes of the first generation and mating them. The selection criteria could be of many types and the importance of different strategies will be discussed. The number of “parents” involved in the cross-over step depends on the selection strategy, too. The cross-over is set to produce a number of “children” equal to that of “parents”, so we have 20 chromosomes at the end of this phase. Then, the mutation could take place but the number of individuals remain the same. The new generation is now ready and the process could restart for the next iteration.

3. Solving with GALIS

3.1 A new Fortran-code for a new GA implementation

Some authors have already approached the inverse scattering problem using genetic algorithm, in particular [28] and [29]. In this section, we propose the new algorithm *Genetic ALgorithm for Inverse Scattering* (*GALIS*), which will be analyzed more in detail. It drives the evolutionary process using a smoothness constraint and an explicit edge-preserving regularization term. The fitness function, the chromosome and the implementation of the operators (cross-over, mutation and selection) presented in the previous section will be described step by step. In the last part, the most interesting results obtained so far will be shown and some possible enhancement to the method will be discussed (parallelization, hybridization).

3.2 Parameters and Operators

The algorithm starts by reading an *input* file, containing the set of parameters (shown in Table 3.1) that define some critical quantities, such as the probability of cross over (*Pcross*), the initial probability of mutation (*Pmut_ini*) and others stated in the next paragraphs.

Pcross : fixed value of the cross-over probability. *Pmut_ini :* starting value of the mutation probability. *Pmut_incr :* increment in the value assumed by probability mutation throughout the evolutionary process. *Pmut_max :* higher value (threshold) of the probability of mutation. *Max_gen:* maximum number of generations allowed in the evolution.	*Freq_min :* minimum working frequency. *Freq_max :* maximum working frequency. *Nc_freq :* number of evaluation points in the defined band. *Thick_wall:* thickness of the probed wall. *Num_Layers:* discretization of the wall. *Min_fit*: minimum value for fitness function. *Cont_sim :* flag to restart the simulation from a previous state of evolution.

Table 3. 1 Parameters in the *input* file.

All the physical parameters are binary coded, so at the beginning we need to create a set of binary strings (which are called *chromosomes*). The program calls the subroutine *TIME,* which gives as output a variable containing the number of seconds from January 1st 1970. This variable is then used as a seed in the subroutine *CREA,* for a generator of random variables uniformly distributed between 0 and 100. This subroutine generates a matrix of bits (*PMAP*) assigning the values of the bits that form the binary coded chromosomes in this way: ‘1’ if the generated random variable lies in the interval 0-50 (extremes included), ‘0’ in the other case. As said before, each row of this map represents a tentative solution for the problem. The length of the chromosome depends on the parameter *Num_Layers.* In fact, in order to represent the most reasonable range of values of the dielectric constant significant for the discussed application, 7 bits are delegated to the representation of the ε_r of each layer. This choice allows us to express ε_r from 1 to 13 with a 0.1 step (which is quite accurate). So, for example, 10 layers mean an occupation of 70 bits. Figure 3.1 summarizes the different parts of a typical chromosome.

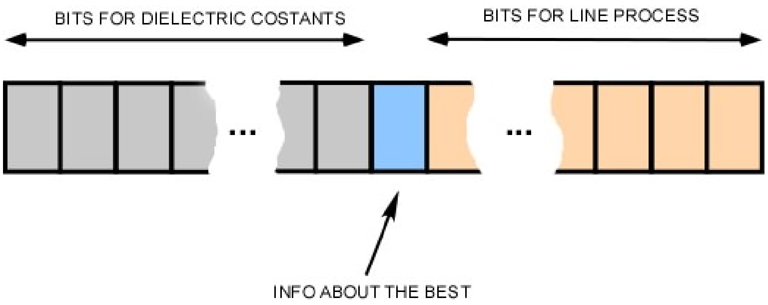

Figure 3. 1 Binary coded structure of each chromosome.

The meaning of the other quantities will be clarified shortly. After the subroutine *CREA*, we have a whole map in which each column is a trial solution. This set of chromosomes is generally called a *population*. During the first run, the program generates a great number of populations (the number can be defined by the user) and, with the subroutine *BUILDBESTMAP*, builds the real first population with the best chromosomes on the basis of the performance expressed by the solutions that they codify. This first wide sweep ensures a good view of the solution domain and it is one of the reasons for the large independence of GAs from the starting point. The next step is the creation of a new set of chromosomes using the Uniform Cross-over operator. While single and multipoint cross-overs define cross points in the genome where a chromosome can be split, uniform cross-over [30] generalizes this scheme to make every point a potential cross-over point. A cross-over mask, with the same length as a chromosome, is randomly created and the value of the bits in the mask indicates which parent will supply the offspring with which bits. Consider the two parents, cross-over mask and resulting offspring shown in fig. 3.2:

P_1 = 1 1 1 0 0 0 1 1 0 1

P_2 = 0 0 0 1 1 1 1 0 0 0

Mask = 0 0 1 1 0 0 1 1 0 0

O_1 = 0 0 1 0 1 1 1 1 0 0

O_2 = 1 1 0 1 0 0 1 0 0 1

Figure 3.2 Parents, mask and offsprings in the Uniform Cross-over.

The first offspring, namely O_1, is produced by taking the bit from P_1 if the corresponding mask bit is 1 or the bit from P_2 if the corresponding mask bit is 0. Offspring O_2 is created using the inverse of the mask or swapping P_1 and P_2. The subroutine *CROVER*, according to the *Pcross* value, drives this operation and repeats it until the fixed number of chromosomes for a population is reached. After the cross-over stage, mutation could take place in some of the new chromosomes. As said before, mutation provides a guarantee of searching any possible string, thus it is a key factor for the sweep of the irregular solution space. The effect of subroutine *MUTAT* on the binary string, implemented in the code with a NOT logical operator, is illustrated in fig. 3.3.

Original string --- 0 1 1 1 0 0 1 0 1 0 0 1 1

Mutated string --- 0 1 1 0 0 0 1 0 1 0 0 1 1

Figure 3.3 Mutation by a NOT logical operator

High mutation rates allow great jumps to be made within the solution space, and are among the tools that prevent the solution being entrapped in local minima. However there is no “best value” able to manage each problem and, even for the same application, it is very hard to find a value well suited for each situation. Our solution is to adopt an adaptive value for the mutation rate. It starts from a default value (*Pmut_ini*) and we add a fixed value to it (*Pmut_incr*) if in the new genome produced we don’t find a better value of the best fitness than the actual. This increment goes on until we reach a maximum value (*Pmut_max)* or when the fitness of the best chromosome of the new population becomes better than that of the previous generation. In this case we bring back the mutation probability to the default value. Together with the simple elitism described later, this variable mutation probability, which increases when there is no further evolution due to other operators, ensures the convergence to the global optimum [31]. The next step is the evaluation of the fitness function for the new individuals and their selection in order to decide who will take part to the successive phase of the evolutionary process.

3.3 Selection and Fitness function

As we said, the fitness function is used to provide a measure of how chromosomes have performed as a solution. The best individuals will give rise to the lowest numerical values in the associated function. Our fitness function consists in the sum of three different terms:

$$FITNESS = d\left(\rho_{meas} - \rho_{calc}\left(\varepsilon\right)\right) + \lambda \sum_{k=1}^{N-1}\left(\varepsilon_{k+1} - \varepsilon_k\right)^2 \left(1 - l_k\right) + \alpha l_k \qquad (3.1)$$

They can be distinguished into two groups: the former deals with the elaboration of the measured data and the latter in the two terms related to the *a priori* information. The first addendum takes into account the distance between the measured reflection coefficient and the results produced by the tentative configuration of layers proposed by the genetic algorithm. This distance is a linear combination of the root mean square errors between real and calculated magnitudes and phases.In the second part the first member forces a smooth variation of the dielectric constant values inside the wall. The assumption in the formula is that ε_k is the value of the permittivity of the *k-th* layer (as in the model presented in Section 1). This constraint does not permit a sudden change in the permittivity profile and is introduced because it seems to be a reasonable hypothesis for the most part of the investigation domain, except for the presence of some discontinuities. To enable the algorithm to accept a few abrupt changes in permittivity, a line process has been introduced, which has been revealed a powerful enhancement. The choice of the numerical values for the parameters λ and α is quite hard. In fact, as we will show in the results, a large λ will produce an over-smoothed profile, close to the average value of the dielectric constant inside the wall. Conversely, without a smoothness constraint, the searching algorithm is not able to reach the solution in a reasonable time. The α parameter is important in the determination of the values in the *l* array, where the flags for the line process are stored. The binary variable l_k denotes the presence of a discontinuity between the *k-th* and *(k+1)-th* layer. When $l_k = 1$ the regularization functional does not penalize any difference between the values of ε_k and $\varepsilon_{k+}1$ and the contribution to the fitness is constant and equal to α. Any possible discontinuity between the two layers is allowed, with the drawback of a constant penalization. On the contrary, when $l_k = 0$, the difference between the dielectric constants of the adjacent layers is quadratically penalized and a discontinuity is discouraged. The value of $\sqrt{\alpha/\lambda}$

acts as a threshold for allowing a discontinuity to be introduced. The parameter α had to be chosen according to the desired definition of the discontinuity. In order to evaluate the fitness, the chromosome is used as input of the function *EXTRACT*, which decodes, converting from binary to real, the physical parameters of the problem. Then, they are used in the function *FITNESS*, which gives the fitness of the chromosome. At this point, we want to verify if we have reached our goal or, in other words, if we are in the global minimum of the fitness landscape. Due to the presence of the smoothness constraint and the line process and that measured data are corrupted by noise, we can't expect to find zero fitness. So, we fix a minimum fitness value as a threshold and we compare it, at the end of each evolutionary cycle, with the best fitness found until that time. This is done by the subroutine *ITEST*. If this fixed minimum is not yet achieved, the algorithm starts the selection of the chromosomes for the creation of a new population. First of all, the subroutine *VERCRO* compares best chromosome of the last population with the best one of the current step. In the case when the latter is worse than the former, the old chromosome is the first to enter in the new party. This is called simple elitism and, in each stage of evolution, preserves the chromosome with the best fitness from a generation to the following. The flow chart in fig. 3.4 shows the complete procedure.

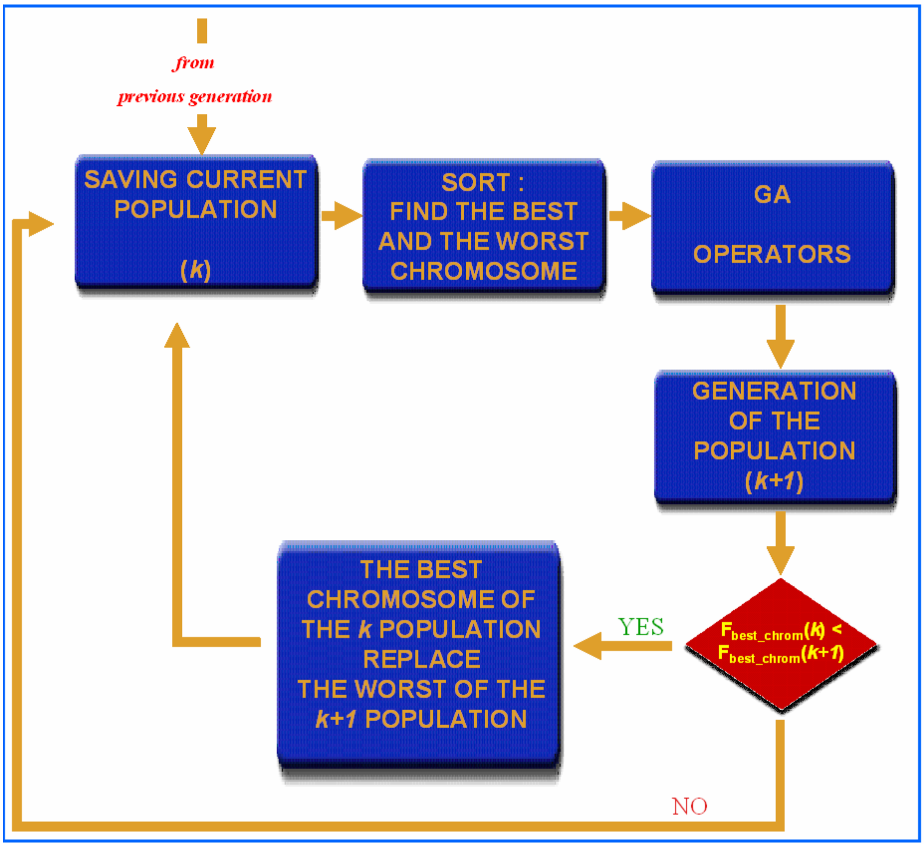

Figure 3.4 Flow chart explaining the simple elitism procedure.

After that, the subroutine *BEST* has to select the other chromosomes. Two selection strategies are implemented in this code. The reasons of this choice will be motivated and clarified in the section devoted to the results. The first selection strategy is the Roulette Wheel Selection, which has been described in Section 2. The second is the Direct Selection which sorts all the chromosomes using their fitness values and choose the first *n* best ranked among all others to create the new set, where *n* is the number of chromosomes which forms a population.

3.4 Results

The first thing we employed was that the smoothness constraint alone is not sufficient to obtain a good reconstruction, since it does not allow significant discontinuity to be introduced, especially for high values of λ. An example is shown in fig. 3.5.

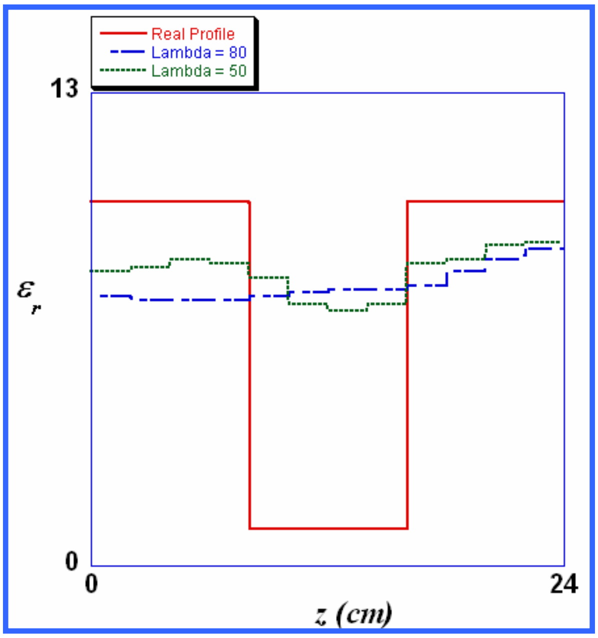

Figure 3.5 Profile reconstruction using smoothness alone.

Introducing the line process should avoid this drawback. However, if we activate the line elements from the beginning, we have observed a very slow convergence and, in order overcome this problem, we decided to organize the evolutionary process into two different stages. In the first stage, the fitness function only contains the smoothness constraint and the space of solutions is sampled to find a good set of guess. In a second stage, the line process takes part in the computation. Besides, in the first stage, the selection criterion is the proportional one, because we should not use a small set of individuals as the starting genoma. In the second stage we use direct selection because we are quite sure to have reached the proximity of the global minimum and we want to converge faster

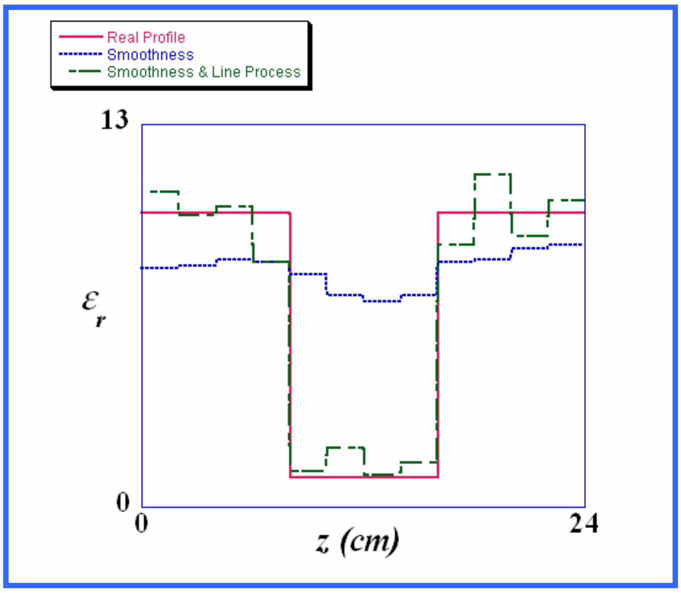

Figure 3.6 Profile reconstruction using smoothness and line process (λ=3.5, α=56) with a SNR=25 dB.

3.5 Conclusions

We have presented a new technique to invert multifrequency backscattering data from a discontinuous permittivity profile. To reach the goal, we have built the described genetic algorithm.

We are now improving our code, especially to face 2D problems and to prevent a slow convergence. A solution could be to increase the computational power, implementing some form of parallelization or to use a local optimization technique together with the GA. We are studying new methodologies to refine the solver, in order to remove the plane wave incidence hypothesis and manage data with a higher level of noise.

REFERENCES

[1] M. Slaney, A.C. Kak, and L. E. Larsen, "Limitations of Imaging with First Order Diffraction Tomography", *IEEE Trans. Microwave Theory Techn.*, vol. 32, No. 8, pp. 860-874, August 1984.

[2] R. W. Gerchberg, "Super-resolution through Error-Energy Reduction", *Optica Acta*, vol. 21, No 9, pp. 709-720, 1974.

[3] A. Papoulis, "A new Algorithm in Spectral Analysis and Band-Limited Extrapolation", *IEEE Trans. On Circuits and Syst.*, vol 22, No. 9, pp.735-742, 1975.

[4] B. A. Roberts and A. C. Kak, "Reflection Mode Diffraction Tomography", *Ultrasonic Imaging*, vol. 7, pp. 300-320, 1985.

[5] F. Gori and S. Wabnitz, "Modification of the Gerchberg Method Applied to Electromagnetic Imaging", *Inverse Methods in Electromagnetic Imaging, NATO-ASI*, vol. C 143, part 2, pp.1189-1203, 1985.

[6] E. Salerno, "Microwave tomography of lossy objects from monostatic measurements", *IEEE Trans. Microwave Theory and Techniques*, vol. 47, Issue: 7, pp.986-994, July 1999.

[7] R. Pierri, and A. Tamburrino, "On the Local Minima Problem in Conductivity Imaging via a quadratic Approach", Inverse Problems, vol. 13, pp.1547-1568, 1997.

[8] R. Pierri and A. Brancaccio, "Imaging of a rotationally symmetric dielectric cylinder by a quadratic approach", *J. Opt. Soc. Am.*, vol. 14, No. 10, pp.2777-2785, 1997.

[9] V. H. Weston, "Nonlinear Approach to Inverse Scattering", *J. Math. Phys.*, vol. 20, pp.53-59, 1979.

[10] S. J. Norton, "Iterative Seismic Inversion", *Geoph. J. Roy. Astr. Soc.*, vol.94, No3, pp.457-468, 1988.

[11] D.K. Ghodgaonkar, O.P. Gandhi, and M.J. Hagmann, "Estimation of Complex Permittivities of Three-Dimensional Inhomogeneous Biological Bodies", *IEEE Trans. Microwave Theory Techn.*, vol. 31, No. 6, pp. 442-446, June 1983.

[12] A. N. Tickonov and V. Y. Arsenin, *Solution of ill-posed problems*, Washington, Winston-Wiley, 1977.

[13] P. Lobel, L. Blanc-Feraud, C. Pichot, and M. Barlaud, "A New Regularization Scheme for Inverse Scattering", *Inverse Problems*, vol. 13, pp.403-410, 1997.

[14] L. Garnero, A. Franchois, J.-P. Hugonin, C. Pichot, and N. Joachimowicz, "Microwave Imaging - Complex Permittivity Reconstruction by Simulated Annealing", *IEEE Trans. Microwave Theory Techn.*, vol. 39, No. 11, pp.1801-1807, November 1991.

[15] E. Bermani, S. Caorsi, and M. Raffetto, “Microwave detection and dielectric characterization of cylindrical objects from amplitude-only data by means of neural networks”, *IEEE Trans. Ant. Propag.*, vol. 50, No. 9, pp.1309-1314, September 2002.

[16] E. Bermani, A. Boni, S. Caorsi, and A. Massa, “An innovative real-time technique for buried object detection”, *IEEE Trans. Geosci. Remote Sensing*, vol. 41, No. 4, pp.927-931, Part 2, April 2003.

[17] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”, *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 6, No. 6, pp.721-740, November 1984.

[18] H. Hidalgo, J.L. Marroquın and E. Gomez-Trevino, “Piecewise Smooth Models for Electromagnetic Inverse Problems”, *IEEE Trans. Geosci. And Remote Sens.*, vol. 36, No. 2, pp. 556-561, March 1998.

[19] J. M. Johnson, and Y. Rahmat-Samii, “Genetic Algorithms in Engineering Electromagnetics”, *IEEE APS Magazine*, vol. 39, 1997.

[20] R. L. Haupt, “An Introduction to Genetic Algorithms for Electromagnetics”, *IEEE APS Magazine*, vol. 37, 1995.

[21] D. S. Weile and E. Michielssen, “Genetic Algorithm Optimization Applied to Electromagnetics: A Review”, *IEEE Trans. on APS*, vol. 45, 1997.

[22] G. Manara, A. Monorchio and R. Mittra, “A new genetic algorithm-based frequency selective surface design for dual frequency applications”, *APS IEEE Int. Symp.*, vol. 3, pp. 1722-1725, July 1999.

[23] D. S. Weile, E. Michielssen and D. E. Goldberg, “Genetic Algorithm Design of Pareto Optimal Broadband Microwave Absorbers”, *IEEE Trans. on EMC*, vol. 38, pp. 518-526, August 1996.

[24] J. H. Holland, “Adaptation in Natural and Artificial Systems”, Ann Arbor, The University of Michigan Press, 1975.

[25] K. A. De Jong, “Analysis of the Behavior of a Class of Genetic Adaptive Systems”, *PhD Dissertation*, University of Michigan, 1975.

[26] A. Monorchio, G. Manara, and L. Lanuzza, “Synthesis of Artificial Magnetic Conductors by Using Multilayered Frequency Selective Surfaces”, IEEE AWP Letters, Vol. 1, pp. 196-199, 2002.

[27] S. Caorsi, A. Massa, M. Pastorino, and M. Donelli, “Improved Microwave Imaging Procedure for Nondestructive Evaluations of Two-Dimensional Structures”, IEEE Trans. On APS, vol. 52, pp. 1386-1397, June 2004.

[28] S. Kent and T. Gunel, “Dielectric Permittivity Estimation of Cylindrical Objects Using Genetic Algorithm”, *J. Microw. Power Electrom. En.*, vol. 32, No. 2, pp. 109-113, 1997.

[29] S. Caorsi, A. Massa and M. Pastorino, “A computational technique based on a real-coded genetic algorithm for microwave imaging purposes”, *IEEE Trans. Geosci. Rem. Sens.*, vol. 38, No. 4, pp. 1697-1708, July 2000.

[30] G. Syswerda, “Uniform crossover in genetic algorithms”, *Proc. ICGA 3*, pp. 2-9, 1989.

[31] G. Rudolph, “Convergence Analysis of Canonical Genetic Algorithms”, *IEEE Trans. On Neural Networks*, vol. 5, No. 1, January 1994.