Student report

# GENETIC ALGORITHMS AND THEIR APPLICATION IN SOLVING SHAPE OPTIMIZATION PROBLEMS IN ELECTROMAGNETICS

Igor TIMARAC[1], Mićo GAĆANOVIĆ[2]

**Abstract:** This paper presents genetic algorithms and one example of their application. It is also a demonstration of combining COMSOL for modeling electromagnetic problems and Matlab for conducting the optimization process. First, an overview of genetic algorithms and motivation for their use is given. As an example, shape optimization of pole rotor face is carried out.

**Keywords:** Genetic algorithms, Shape optimization, Finite Element Method

## INTRODUCTION

Optimization is a process of improving performance towards some goal. From the mathematical point of view, it is a process of finding global maximum (or minimum) of some objective function. There are several approaches to this problem. One class of optimization techniques uses well-known conjugate-gradient (CG) methods for solving systems of linear equations, and Quasi-Newton (QN) methods for the non-linear case. Another class is so-called Simplex method, using the concept of a simplex in choosing in which direction to lead the search. Both these techniques are considered local in that the solution they find is highly dependent on the initial point of search. They can never guarantee that the solution is really the best, but due to their usual tight connection with the problem, they tend to converge to the solution relatively quickly. Another problem with local techniques is that all of them impose some constraints on the objective function in terms of continuity and differentiability, which are sometimes impossible to achieve.

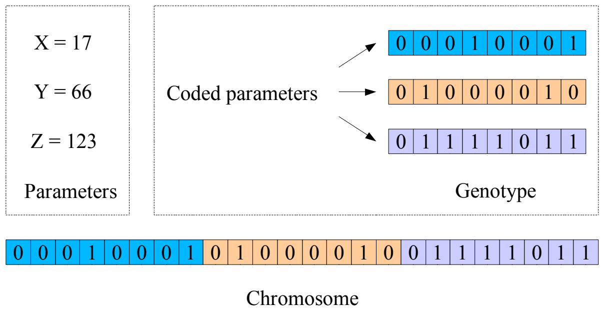
Very different class from these is stochastic optimization. Techniques that fall into this category are considered global and they usually work with population of candidate solutions, not just with one solution, using probabilistic transitions between points in search space. Since they usually do not use any knowledge about the problem, their convergence is slower than that of local techniques. But the fact that they can work with any kind of optimization function, being it non-continuous, non-differentiable, or with any kind of constraints, gives global techniques a great potential for use in various fields where finding the best solution is more important than the convergence time.

Different stochastic methods are developed: genetic algorithms (GA), evolution strategies, Monte Carlo, simulated annealing, particle swarm, etc. GA is one of the methods that proved to be good in solving problems in electromagnetics – they are robust enough and easily implemented in the same time.

## GA OVERVIEW

Genetic algorithms utilize the concept of natural selection and genetics. They work on a very simple principle, which we will present shortly.

First, we need to define encoding of each optimization parameter into one string over some finite alphabet consisting of GA's building units called **genes**. Usually, binary coding is used, so we have genes that take values of 0 and 1. Structure consisting of genes that make up all the parameters is called **genotype**. Merging all these genes, we get the string called **chromosome** (fig 1). Such chromosome represents one point in the search space, that is, encoded parameter vector.




**Fig, 1** - *Parameters, genes and a chromosome*

By selecting N random points from search space and converting them to chromosomes, we get initial population of individuals. This population will now traverse through search space by utilizing three operators [2]:

- selection
- crossover
- mutation

Each of the individuals in population undergoes fitness evaluation, that is, evaluation of the objective function. The resulting number (fitness) is the quantitative measure of solution "goodness" and it represents the only connection between GA and the problem

[1] University of Banja Luka, Faculty of Electrical Engineering, Patre 5, 78000 Banja Luka, Bosnia and Herzegovina, E-mail: tymarats@gmail.com
[2] University of Banja Luka, Faculty of Electrical Engineering, Patre 5, 78000 Banja Luka, Bosnia and Herzegovina, E-mail: bilchy@blic.net

solved. Individual's fitness directly affects the probability of its survival

### Selection

Now, from the current population, the algorithm selects individuals that will proceed (survive) to the next generation. Temporary population is made from the current population by random selection of N individuals; probability of selecting each individual is biased by its fitness – fitter individuals have greater chance to be selected. Each individual can be selected more than once. There are different selection strategies, but the most popular in the literature is roulette wheel selection. Here, individuals are selected based on a probability of selection given in Equation 1 where $f(parent_i)$ denotes the fitness of the $i$-th parent [1],

$$p_{selection} = \frac{f(parent_i)}{\sum_i f(parent_i)}. \qquad (1)$$

This selection method requires the objective function to be strictly positive. If this is hard to achieve, there are different selection schemes (like tournament selection [1], [2]), which do not impose such constraint.

### Crossover

From temporary population, we select individuals, with the probability $p_{cross}$, which will serve as parents for crossover. Crossover means exchanging genetic material between individuals by copying parts of the chromosome from one parent to another to produce children. This process would ideally combine "good" portions of each chromosome, producing a child with superior fitness than its parents, but we do not know in advance which portions are "good". The simplest kind is single-point crossover, where we select a random location (locus) in parents' chromosomes, and then exchange the portion of chromosome preceding the selected point from parent 1 to child 1 and from parent 2 to child 2. Portion of chromosome following the selected point is copied from parent 1 to child 2 and from parent 2 to child 1, as presented on Fig.2 [1].

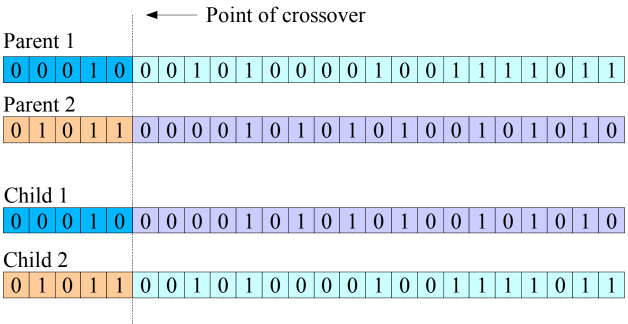



**Fig. 2** - *Single-point crossover*

After crossover, children are put back into the temporary population in place of their parents. $p_{cross}$ is set to an arbitrary value and typically high probability values within range [0.6, 0.8] have been found to work best in most situations [1].

### Mutation

Mutation is a mechanism for extending the search on the new areas of search space. It consists of random altering of bit values inside chromosome with given probability. Random number $p$ is generated for each allele in each chromosome and if it satisfies relation $p < p_{mut}$, selected allele is altered [1] (Fig. 3). $p_{mut}$ is the probability of mutation and it is typically set to low values within range [0.01, 0.1]. While crossover works towards convergence of the algorithm to one solution, mutation works against this goal, enabling the algorithm to look for better solution in yet unexplored areas.

Finally, we replace the current population with temporary population and repeat the process until the termination condition (such as closeness to the solution, maximum number of generations, maximum number of simulations, and so on) is met.

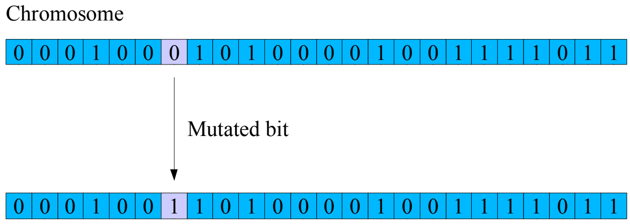



**Fig. 3**- *Binary mutation*

The algorithm described is called simple genetic algorithm (SGA). The Schema Theorem, also known as the Fundamental Theorem of Genetic Algorithms [1], explains why this relatively simple optimization concept really works. There are many improvements to SGA by altering the coding scheme (Gray-coded chromosomes, real-coded chromosomes [1]), selection function (tournament selection, rank selection [1],[2]), or introducing new concepts (elitism, fitness scaling, steady-state GA, diploidy and dominance, inversion, niche and speciation [1],[2]), for usage of GA in different applications.

## OPTIMIZATION PROBLEM

The problem on which we will present the application of GA is shape optimization of rotor pole face (Fig. 4). The pole face should be shaped in such a way that magnetic flux density has a sinusoidal distribution over the region of interest. Fig. 5 presents magnetic flux density distribution of flat-shaped pole together with required sinusoidal distribution.

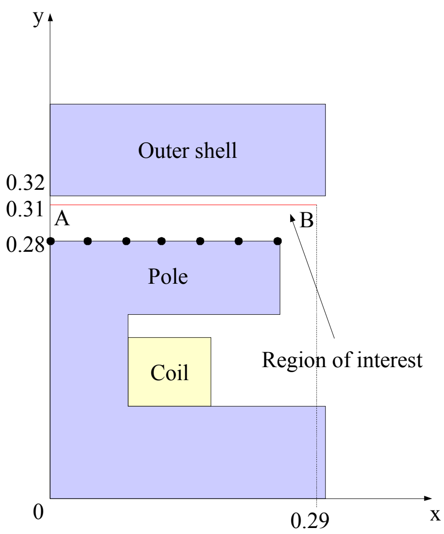



**Fig. 4**- *Rotor pole face*

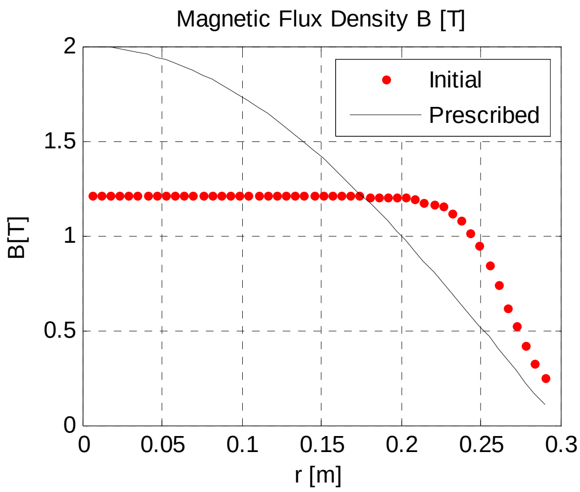



**Fig. 5**- *Required distribution of magnetic flux density*

Model was made in COMSOL Multiphysics, using Electromagnetics Module, Axial Symmetry Magnetostatics model. Linear segments between six control points (Fig. 5) define pole face shape. For pole, linear magnetic material with high relative permeability is used. Magnetic flux is driven by the current density of $5\ \mathrm{A/mm^2}$ through the coil. Exterior environment is air.

Model has been exported to an m-file as a script, so that we can use it in conjunction with Matlab for GA optimization. Control points have fixed positions in horizontal (x) direction, but their vertical (y) coordinates can be changed in order to achieve required distribution. Control points' y-coordinates represent the GA parameters (search space) and they can change anywhere between 0.21 and 0.3 m.

## OPTIMIZATION RESULTS

Optimization was carried out using freely available Matlab toolbox named **GAOT** [7]. For the purpose of this work, its code was somewhat altered and supplemented with support for Gray coding and a simple GUI for easier use and monitoring of optimization performance (Fig. 6). Standard Matlab optimization toolbox may have also been used, but we decided to look for simpler code, which can be modified more easily in order to support Gray coding and optional elitism.

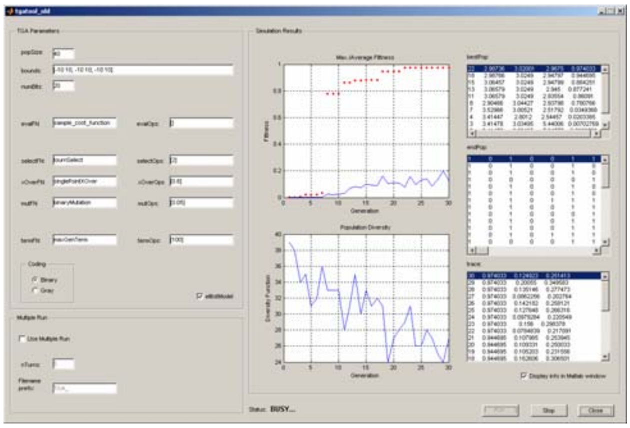


**Fig. 6**- *GUI for GAOT*

Objective function is of form:

$$OF = 100 \cdot \sqrt{\sum_i \frac{\left(B_i - B_{presc_i}\right)^2}{B_{presc_i}^{\ 2}}}, \qquad (2)$$

where $B_i$ represents calculated magnetic flux density at the $i$-th point, and $B_{presc_i}$ represents prescribed magnetic flux density at that point. We strive to minimize the value of OF (ideally, to set it to zero) by changing pole face, so that the flux density distribution we get declines from the prescribed one as little as possible. In our example, we used 50 points, equally distributed along the segment AB (fig 4) and calculated flux density values in these points using COMSOL API function named **postinterp**[6]. As we expect the shape to be descending, we could involve some sort of penalty function for configurations that do not meet such requirement [1], but that was not done here, since it turned out that event this simple objective function almost always finds solution that meets our expectations.

We encoded each parameter (control point's y-coordinate) into a 14-bit gene, so we got chromosomes 84 bits wide. We used binary as well as Gray coding. Number of individuals in population was variable, but we noticed that population size of 40 was good choice in all experiments. Crossover and mutation probabilities were set to its typical values of 0.8 and 0.05, respectively. Varying of these probabilities within reasonable boundaries showed neither significant improvement, nor deterioration of the optimization process. Binary (size 2) tournament selection and single-point crossover were used, with elitism turned on. Termination condition was maximum generation number, and this was set to 100.

Both versions of experiment (binary and Gray-coded) were conducted several times, to eliminate the effect of stochastic nature of GA on result analysis. First thing we noticed was that GA sometimes converges too fast (regardless of increasing the population size) and produces suboptimal solutions. Several tests were made, and about half of them were far from good, even with the aforementioned penalty function included. It turned out that GAOT's view of elitism was somewhat different from that proposed in literature [1], [2]. For this problem, it shows that better results are obtained when we do not copy the best individual from the preceding to the current population every time, but only when the current population's best individual is less fit than one from the preceding population. In average, Gray-coded GA achieved cost function value of 0.60, while its binary-coded counterpart achieved 0.66. It shows that Gray-coded GA is better suited (in terms of convergence rate and solution accuracy) for this kind of optimization problem, but not as much as we expected [3], because we used linear interpolation between control points. Using higher-order curves would certainly improve the results. The best solutions of both variants are presented on Figures 7 and 8.

## CONCLUSION

We have shown how genetic algorithms can be successfully applied to inverse shape optimization problems in electromagnetics. Although this kind of optimization is inherently slow (one experiment took about two hours to complete), this is not a concern here, since we typically need to conduct the optimization only once. Additionally, we can make several runs of GA with different parameters, or different population sizes, and take the best solution. Real-world shape optimization applications would use more complex problem representation, but GA remains applicable regardless of problem's complexity.

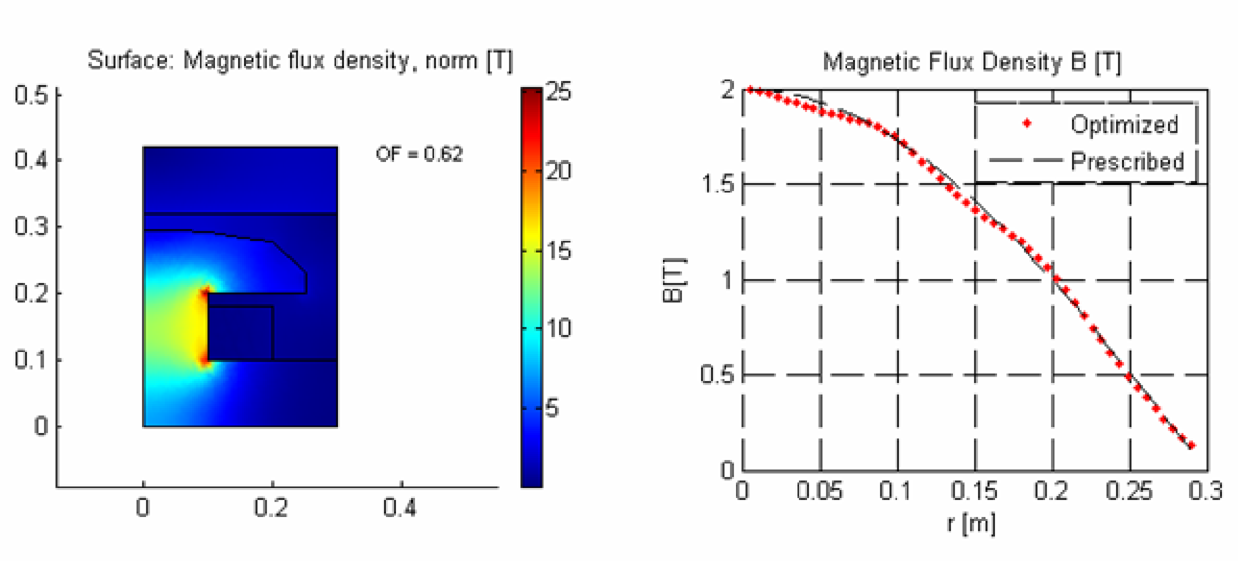



**Fig.7**-*Optimized pole face and attained distribution of magnetic flux density (binary coding)*

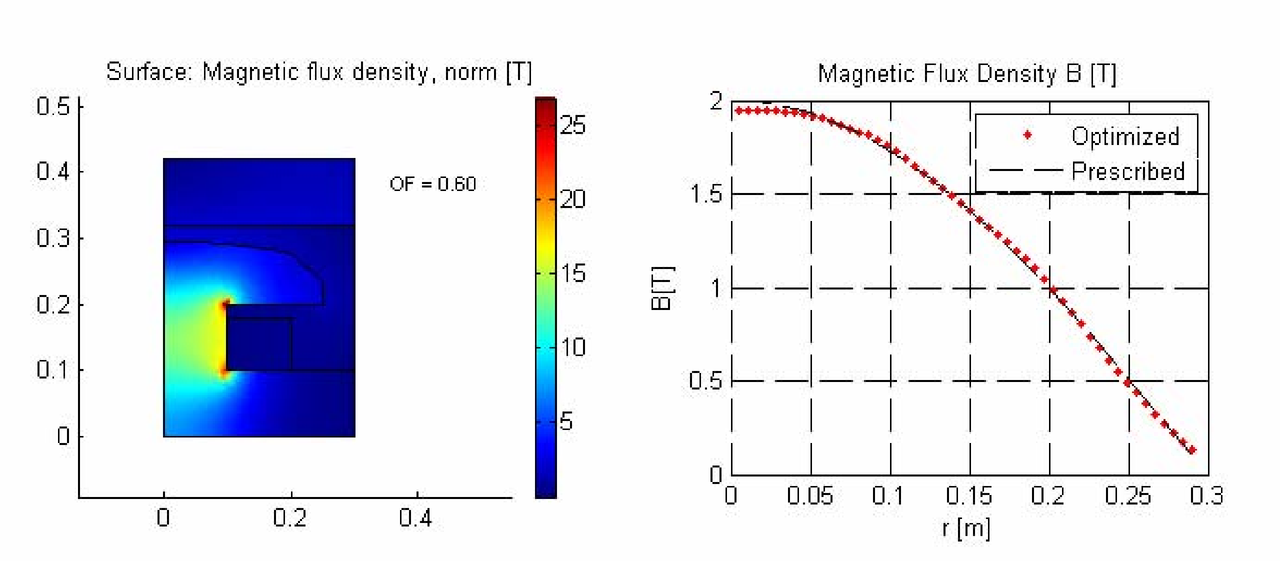



**Fig.8**-*Optimized pole face and attained distribution of magnetic flux density (Gray coding)*


## REFERENCES

[1] Y. Rahmat-Samii., E. Michielssen: "Electromagnetic optimization by genetic algorithms", Wiley, 1999

[2] D. E. Goldberg, D. Edward: "Genetic algorithms in search, optimization and machine learning", Addison-Wesley, 2004

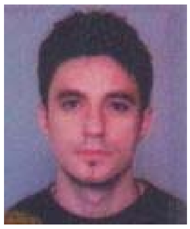
[3] Y. Yokose, V. Cingoski, K. Kazufumi and H. Yamashita: "Performance comparison between gray coded and binary coded genetic algorithms for inverse shape optimization of magnetic devices", Proc. 1st Japanese-Bulgarian-Macedonian Joint Seminar on Applied Electromagnetics and Mechanics, Sofia, 1998

[4] H. Brauer, M. Ziolkowski: "Shape optimization with adaptive simulated annealing and genetic algorithms", ISEF 2003 – 11th International Symposium on Electromagnetic Fields in Electrical Engineering, Maribor, Slovenia, 2003

[5] Matlab 7 Documentation (http://www.mathworks.com)

[6] COMSOL Multiphysics 3.2 Documentation (http://www.comsol.com)

[7] GAOT Documentation (http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/)

**Timarac Igor** was born in Odžak, Bosnia and Herzegovina, in 1983. He is a student of 5th year on Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina.

In 2006, he spent three months on Technical University of Ilmenau where he was working on his diploma thesis concerning technical applications of genetic algorithms.

**Dr. Mićo Gaćanović** was born in 1952. He is recognized and known internationally as a scientist in the field of applied electrostatics, where he has given his contribution through original solutions, which are patented in 136 countries throughout the world and applied in production.

He received many prestigious world-known awards and certificates for his creative work. Hence, he is included in the work of world groups of creativity, research and new technology in Brussels, Moscow, Pittsburgh and other world cities. He is also involved in research projects from the field of theoretical electrical engineering in Germany, Belgium and Russia.