



A software tool for stochastic optimization of electromagnetic devices

G. Aiello, S. Alfonzetti, E. Dilettoso and N. Salerno

*Dipartimento Elettrico, Elettronico e Sistemistico,
Università di Catania, Italy.*

Abstract

This paper gives an overview of the basic features of a software tool for the optimization of electromagnetic devices. Two stochastic optimization methods are implemented: Simulated Annealing and Genetic Algorithms. This tool works coupled with ELFIN, a finite element code developed by the authors for electrical engineering research. ELFIN allows us to deal with electromagnetic field problems in a fully parametric mode with respect to a set of variables, typically representing the geometrical or constitutive data of the electromagnetic device to be optimized. An example of application is also given.

1 Introduction

The optimization of electromagnetic devices by means of stochastic methods generally requires a large computing effort [1], because each estimation of the objective function calls for the numerical solution of an electromagnetic field problem by means of the Finite Element Method (FEM). Due to this fact, the FEM code needs to be very efficient not only in data structure, memory management and I/O operations, but also in the strategy used to solve the field problem.

In this paper we give an overview of the basic features of a software tool for the optimization of electromagnetic devices by means of two stochastic methods: Simulated Annealing (SA) and Genetic Algorithms (GA). This tool includes ELFIN, a finite element code developed by the authors for electrical engineering research, which allows us to deal with field problems in a fully parametric mode with respect to a set of variables, typically representing the geometrical or constitutive data of the device to be optimized.

The paper is structured as follows. In Sect. 2 the optimization tool is described. In Sect. 3 an FEM procedure is described to reduce the whole optimization time. In Sect. 4 the FEM formulation for eddy current problems in an infinite domain is derived. In Sect. 5 an example of application is given. The authors' conclusions follow in Sect. 6.

2 The software tool

The optimization tool is logically structured in two blocks, as shown in Fig. 1:

- i. *Optimization block*: this block manages the optimization procedure and then the behaviour of the chosen optimization method (SA or GA). It performs calls to the evaluation block when it is necessary to compute the value of the objective function associated with a particular system configuration. In the case of multiobjective optimization, it merges all the objectives of the vector optimization in a single scalar function which expresses the global quality of the system configuration.
- ii. *Evaluation block*: this block contains the evaluation tools, that is, the FEM code ELFIN, which is classically structured into three distinct main programs devoted to preprocessing, processing and postprocessing, which communicate through files. A command procedure, ELFIN_COM, manages the sequential activation of these programs.

All the parts of the software tool are written in Fortran, except for the command procedure which is written in the operating system command language of the workstation used. In the following sections the SA and GA optimization methods are briefly described, as well as the FEM code ELFIN.

2.1 Simulated Annealing

SA algorithms are optimization methods based on a strong analogy between the physical annealing of solids and the issue of solving optimization problems.

In physics annealing is known as a thermal process to obtain low energy states of a solid in a heat bath, following two steps: i) the temperature of the heat bath is increased to a maximum value at which the solid melts; ii) the temperature is carefully decreased until the particles arrange themselves in a minimal energy state. The mathematical model of the physical process was introduced by Metropolis [2]. In SA algorithms the state of the physical system is equivalent to choosing the parameters of the optimization problem and the minimal energy state is equivalent to the global optimum of the problem. Transition from one state to another is ruled by an acceptance criterion, called the Metropolis criterion, that gives the probability that a particular configuration will be accepted. This allows some probability of accepting parameter configurations that have a worse objective function value: this means that SA can escape from local minima. The behavior of the algorithm is ruled by the choice of a group of parameters combined in the so-called cooling schedule; the control parameter of the algorithm is called temperature, in analogy with the physical case.

In this tool a modified SA [3] featuring by a complete and very efficient structure was implemented; the most commonly used cooling schedules are present with high possibilities of tuning their control parameters; moreover, some recently proposed adaptive techniques are also implemented and utilized together to improve the probability of success [4]. The mechanism used to generate new configurations in the solution space is a modified Simplex Method [5], which is more appropriate for the treatment of continuous problems. A technique was also implemented to control the simplex size in order to prevent premature convergence of the algorithm. The software has no limits on the number of optimization variables; the input data is given by means of a text file that contains not only the problem data (range of variables, constraints) but also the algorithm parameters (cooling schedule chosen, maximum number of evaluations, convergence criterion, etc.); the output data is written in a text file that can easily be graphed or manipulated; the parameter configuration is stored in a data file (PCB) that is the input of the evaluation block.

2.2 Genetic Algorithms

The algorithm implemented is a standard GA. GAs are stochastic search strategies that imitate the natural adaptation process of living organisms to their environment [6]. In the standard GAs the optimization parameters have to be coded in a binary system introducing fixed-length strings called chromosomes (or genotypes); each position in the string is called a gene. A population (or generation) contains a finite number of chromosomes. Each chromosome is characterized by its fitness, a scalar value calculated by means of an objective function that measures its ability to survive and produce offspring. The initial population, made up of a fixed number of elements, is randomly created. A new population is generated by means of three main operators: selection, crossover and mutation; the last two are applied with an assigned probability. The chromosomes that survive through several generations will be those which have proven to be the most fit.

This code initializes a random sample of individuals. The selection scheme used is tournament selection with a shuffling technique for choosing random pairs for mating. The routine includes binary and Gray coding for the individuals and the option for one-point or two-point crossover; moreover a number of variants are implemented in the code to improve the performance of the algorithm, such as variable probability of mutation and variable probability of crossover. The other features are similar to the SA software.

2.3 The ELFIN code

The ELFIN code is written in Fortran-77 [8]. The basic features are N-dimensional (1D, 2D and 3D) geometrical discretization, a wide application area, easy treatment of coupled problems, and a generalized iterative procedure. The ELFIN code is able to solve the following kinds of problems: Laplace, Poisson, Helmholtz, skin effect, motional diffusion and eddy currents.

The code is structured according to the classical scheme of FE codes, in which three distinct main programs are devoted to pre-processing, processing and post-processing, respectively. The pre-processing program (PREELFIN) exhibits a mixed interactive-batch user interface: in this part the user can define the geometric structure, the kind of problem, the materials and their properties, mesh generation, boundary conditions and sources. The processing program (ELFIN) works in a batch mode, starting from the data prepared in the pre-processing session and stored in a suitable file; the main aspects of the program are: generalized iteration structures, with four nested cycles (non-linearity, adaptive mesh refinement, boundary conditions, time discretization); numerical techniques (variational or Galerkin approaches are used), universal matrices and coupled problems. The post-processing program (POSTELFIN) exhibits the same interactive-batch user interface as the pre-processing one; available restitutions are displaying/printing of nodal values, 2D contour line plotting, axonometric 3D contour line plotting, and 2D and 3D vector plotting. Global quantities (energy, flux, current, etc.) can also be evaluated by means of integrals on elements, boundary (or local) sides or edges.

The most important feature that allows us to use ELFIN for the stochastic optimization of electromagnetic devices [9] [10] is that the code uses symbolic variables as parameters (instead of fixed numeric values) of the batch input command files CMB and CMD of the pre- and postprocessing programs, respectively. The actual values of these parameters are given in the PCB file. In this way both the pre- and post-processing batch sessions (and so the whole code) can be made parametric with respect to a set of variables, typically representing the geometrical or constitutive data of the device to be optimized. The data flux in the optimization process is shown in Fig. 1. When the optimization algorithm (SA or GA) needs to evaluate the objective function, it calls a Fortran routine, FUNK, that writes a file (PCB) containing information about the parameter configuration, and then runs the command procedure which sequentially activates PREELFIN, ELFIN and POSTELFIN. Once activated, PREELFIN reads the CMB and PCB files, executes the commands listed in the CMB file and, finally, writes a data file (DAT) containing all the problem information (FE discretization, boundary conditions, materials, etc.). Once activated, ELFIN reads this data file and, after the field problem solution, writes the results in the RIS file. Finally POSTELFIN reads the DAT and RIS files and executes the required postprocessing commands of the CMD file in order to obtain the output file OUT for the FUNK module. Finally the value of the objective function is calculated by FUNK and is returned to the optimization module.

3 An FEM procedure to reduce the optimization time

It is clear that in an optimization process a single field problem solution for a given system configuration has some data in common with all the other solutions, relating to different system configurations. However, this information is lost, because in a great number of cases the optimization and evaluation

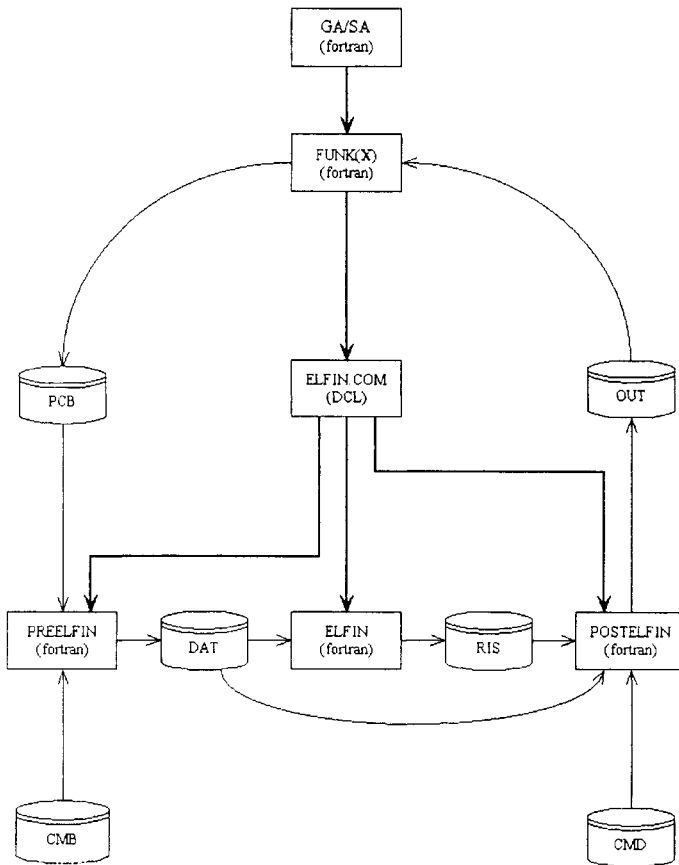


Figure 1: Structure of the optimization tool.

blocks are totally separate. So for every request to evaluate the objective function, the field problem is solved *ex novo* without using the experience gained from previous solutions. This is specifically true for commercial evaluation softwares, which are frequently used like a black box.

In some particular problems a great amount of computational time can be saved simply by using the data relative to previous computations of the objective function. For example, this fact is frequently true for eddy current problems, which are very common in electrical engineering, for example in eddy current non-destructive testing and induction heating applications.

4 Eddy current problems in an infinite domain

Consider a general two-dimensional eddy current problem in an infinite domain [11]. The vector potential A satisfies the following equation

$$-\nabla \cdot (\nu \nabla A) + j\omega \sigma A = J \quad (1)$$

in which ν is the magnetic reluctivity, σ is the electric conductivity, ω is the angular frequency and J is the source current density. The unboundedness of the domain is treated by means of the DBCI (Dirichlet boundary condition iteration) [12], [13] method. DBCI needs the introduction of a fictitious truncation boundary Γ_F , that encloses the conductor and leaves out the current sources (see Fig. 2). Since the potential on Γ_F is unknown, the formulation must be completed with a relation which expresses the Dirichlet condition on Γ_F in terms of the value of the potential in the interior of the bounded domain Ω thus obtained. This relation is of integral type, and thus the formulation adopted in DBCI is a hybrid one. Distinguishing in Ω the free-space region (with zero conductivity) and the eddy current region Ω_e (with vacuum reluctivity ν_0 and $\sigma \neq 0$ conductivity), the potential A at a point P_F on Γ_F can be expressed as follows:

$$A(P_F) = A_{\text{ext}}(P_F) + \frac{j\omega}{\nu_0} \iint_{\Omega_e} \sigma A G(P, P_F) dx dy \quad (2)$$

where $A_{\text{ext}}(P_F)$ is the potential due to the source current external to Ω and $G(P, P_F)$ is the Green's function. By discretizing the bounded domain by means of nodal finite elements the following linear algebraic system is derived from (1) and (2)

$$\mathbf{M}\mathbf{A} = -\mathbf{M}_F\mathbf{A}_F \quad (3)$$

$$\mathbf{A}_F = \mathbf{A}_{\text{ext}} + \mathbf{H}\mathbf{A} \quad (4)$$

where $\mathbf{A}$ and $\mathbf{A}_F$ are the vectors of the value of the potential A at the internal and boundary nodes respectively, $\mathbf{M}$ and $\mathbf{M}_F$ are sparse FEM matrices, and $\mathbf{H}$ is a dense influence matrix. This linear system is solved iteratively.

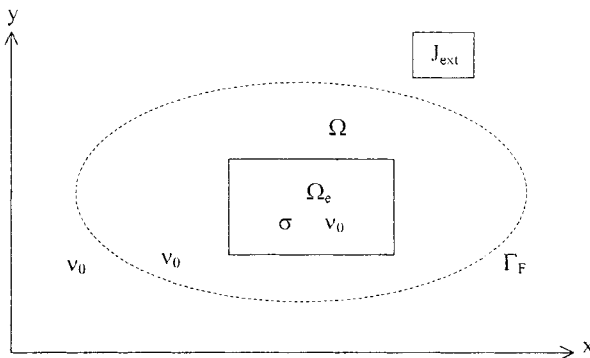


Fig. 2 Eddy current region enclosed by a fictitious boundary.

The matrices $\mathbf{M}$ and $\mathbf{M}_F$ depend on the discretization of the region Ω : a great computational effort is necessary for their construction and their complexity grows with the number of nodes in the FEM discretization. These matrices only depend on the mesh chosen, and, if the discretization is the same along the optimization process, they do not change. Usually, in designing eddy current devices, only the sources are the object of optimization: in most cases the target is to find the shape and intensity that give the desired effect in the eddy current region. If the sources are external, only $\mathbf{A}_{\text{ext}}$ depends on how they are chosen, whereas $\mathbf{M}$ and $\mathbf{M}_F$ are the same and only need to be build in the first solution of the field problem.

The practical realization of this approach is non-trivial and different ways of proceeding are possible. The main issue is to make matrices $\mathbf{M}$ and $\mathbf{M}_F$ available for further use: one possibility is to store them in a file and to acquire them in the processing program when the solving system is constructed. Access to disk for read and write operations is expensive, especially for a large amount of data, and for this reason the best solution is to merge the evaluation and optimization blocks into a single main program with shared memory. Clearly this is not easy to do and is only feasible with open source code. The great modularity of the ELFEN code allows this to be done easily and quickly. The data flow of Fig. 1 is then modified as follows: the FUNK routine calls PREELFIN, ELFEN and POSTELFIN directly; because the mesh is fixed, PREELFIN is only loaded at the first evaluation, then the geometric data is stored in the memory and used, if necessary, in post-processing. ELFEN only constructs the matrices $\mathbf{M}$ and $\mathbf{M}_F$ only at the first evaluation, then stores them; in the subsequent solutions, only the vector $\mathbf{A}_{\text{ext}}$ is updated, and then the system is solved. POSTELFIN provides the data for FUNK. In comparison to the standard approach, the full pre-processing time and a part of the processing time are saved: clearly this saving varies according to the application involved.

In order to describe this approach better, the optimization of an eddy current device is considered in following section.

5 An example of application

One of the devices optimized by means of the tool described is the induction heating system shown in Figure 3. It consists of a solid graphite disk plate (radii $R_1=20$ mm, $R_2=290$ mm, thickness $t=14$ mm, electric conductivity $\sigma=151.5 \cdot 10^3$ S/m, magnetic permeability $\mu_0=4\pi \cdot 10^{-7}$ H/m) faced by a pancake inductor coil with 12 concentric turns (height $h=10$ mm, widths: $w_1=w_2=w_3=w_4=10$ mm, $w_5=12$ mm, $w_6=14$ mm, $w_7=19$ mm, $w_8=20$ mm, $w_9=21$ mm, $w_{10}=11$ mm, $w_{10}=w_{11}=10$ mm), in which a time-harmonic current flows ($I=712$ A, frequency $f=4$ kHz). Due to mechanical constraints, minimum distances must be kept between the inductor and disk ($a=16$ mm) and the internal inductor turn and the z-axis ($b=49.5$ mm); they define the so-called forbidden regions, indicated by dashed lines in the figure.

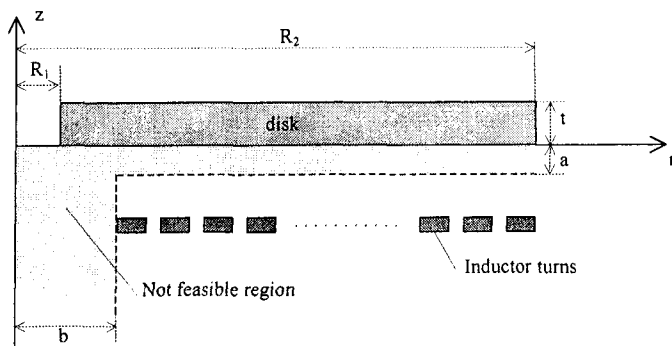


Figure 3: Induction heating system

The goal of the optimization is to find the axial co-ordinates of the turns that allow the power distribution in the graphite disk to approach the target power distribution as closely as possible. The first two turns are fixed at a distance of 16 mm from the disk, while the other ten may be vertically moved from 16 mm to 41 mm from the disk. The objective function f to be minimized has been selected as:

$$f = \sqrt{\sum_{i=1}^{55} \left(\frac{w_i - w_{0i}}{w_{0i}} \right)^2} \quad (5)$$

where w_i and w_{0i} are the calculated and desired power densities, respectively, in 55 equidistant nodes on the horizontal axis of the disk.

Estimation of the actual power distribution due to a particular configuration of the axial positions of the turns is carried out solving the field problem by means of the ELFIN code, using DBCI method. The domain discretized by FEM is the region around the disk, enclosed by a fictitious truncation boundary Γ_F placed at a distance $d=8$ mm from the disk. All the turns are left outside as external sources. The domain was discretized with 232 second-order triangular finite elements and 531 nodes. Since the system is axisymmetric, the problem is solved for the auxiliary potential V , given by $V=A_\phi/r$, A_ϕ being the azimuthal component of the magnetic vector potential. The power distribution is obtained as:

$$w = \sigma \omega^2 \Gamma^2 |V|^2 \quad (6)$$

For both optimization methods, 3000 evaluations of the objective function were used; the optimal solutions found by the two optimization strategies are different, with a slightly worse minimum for SA. However, both the final configurations of the turns accurately approach the target power density curve (see Fig. 4), the residual differences being due to the geometrical constraints of the coils.

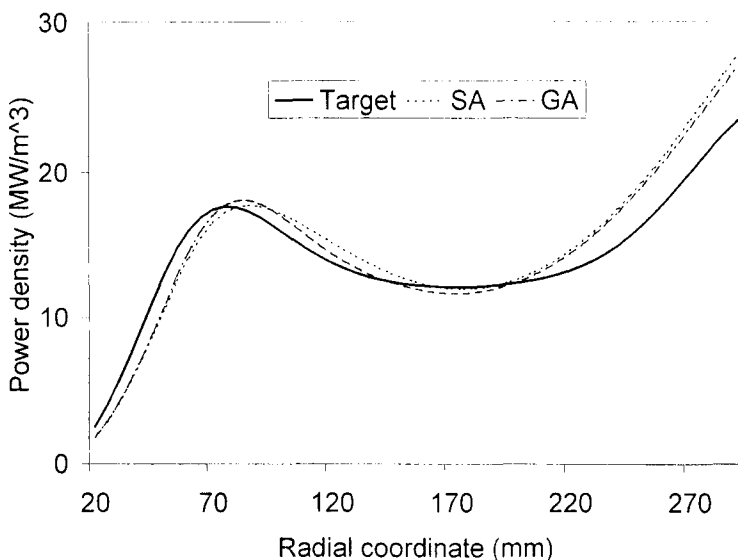


Figure 4: Desired and obtained power density curves.

Application of the suggested approach to this problem allows us to save a great amount of computing time; for this system and with the chosen mesh, the pre-processing program takes about 25% of the total objective function evaluation time. Moreover, in the processing phase, the construction of the solving system takes approximately 30 % of the evaluation time; so altogether a saving of 55% is obtained.

6 Conclusions

In this paper a software tool for the optimization of electromagnetic devices has been presented. The software implements Simulated Annealing and Genetic Algorithms, with large possibilities of tuning the parameters that rule the behavior of algorithms: thanks to these features it is able to tackle optimization problems with different objective functions, always giving good results. Moreover, a great amount of computing time is saved by strongly coupling the tool with ELFIN, a finite element code for field problem solution. This saving depends on the kind of problem involved and on the FEM discretization, but in some cases it has reached more than 50% of the required time. These features are even more important in the case of practical applications in electrical engineering, with expensive objective functions, because they allow us to obtain a good speedup of the entire optimization process.



References

- [1] Russenschuck, S., "Synthesis, inverse problems and optimization in computational electromagnetics," *Int. J. Numer. Modelling: Electronic Networks, Devices and Fields*, special issue on 'Computational Magnetics', vol. 9, pp. 45-57, January-April 1996.
- [2] Metropolis, N., Rosembluth, A. W., Rosembluth, M. N., Teller, A. H., Teller, E., "Equation of state calculation by fast computing machines", *J. Chemical Phys.*, vol. 21, n. 6, pp. 1087-1092, 1955.
- [3] Aarts, E., Korst, J., "*Simulated Annealing and Boltzmann Machines*", John Wiley & Sons Ltd., New York, 1989.
- [4] Alotto et alii, "Stochastic algorithms in electromagnetic optimization," *IEEE Trans. Magn.*, vol. 34, pp. 3674-3684, Sept. 1998.
- [5] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B., *Numerical Recipes in Fortran: the Art of Scientific Computing*, Cambridge University Press, Cambridge (U. K.), 1992.
- [6] Holland, J. H., *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, 1975.
- [7] Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989.
- [8] Aiello, G., Alfonzetti, S., Borzi, G., and Salerno, N., "An overview of the ELFIN code for finite element research in electrical engineering", *Electrosoft '99*, Seville (E), May 17-19 May, 1999.
- [9] Aiello, G., Alfonzetti, S., Borzi, G., Coco, S., Salerno, N., "Shape optimization of the magnetic channel of a superconducting cyclotron", *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering (COMPEL)*, vol. 17, n. 1-2-3, 1998, pp. 123-127.
- [10] Alfonzetti, S., Dilettoso, E., Dughiero, F., Salerno, N., "Stochastic optimization of an induction heating system by means of DBCI" *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering (COMPEL)*, vol. 19, n. 2-3-4, 2000, pp. 569-575.
- [11] Aiello, G. and Alfonzetti, S., "Finite Element computation on axisymmetric eddy currents in an infinite domain" *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering (COMPEL)*, vol. 19, n. 2-3-4, 2000, pp. 167-172.
- [12] Aiello, G., Alfonzetti, S., and Coco, S., "Charge iteration: a procedure for the finite-element computation of unbounded electrical fields," *Int. J. Numer. Methods Engng*, vol. 37, pp. 4147-4166, December 1994.
- [13] Aiello, G., Alfonzetti, S., Coco, S., and Salerno, N., "A theoretical study of charge iteration," *COMPEL*, vol. 15, pp. 22-46, September 1996.