Aspects Regarding the Optimization of the Induction Heating Process using Fmincon, Minimax functions and Simple Genetic Algorithm

Mihaela Novac*, Ecaterina Vladu** , Ovidiu Novac** and Mircea Gordan*

* Department of Electrotechnics, Measurements and using of Electrical Energy,
University of Oradea, Faculty of Electrical Engineering and Information Technology,
410087,1 , Universității Str., Oradea, Romania, E-Mail: mnovac@uoradea.ro

** Department of Computers,
University of Oradea, Faculty of Electrical Engineering and Information Technology,
410087,1 , Universității Str., Oradea, Romania, E-Mail: evladu@rdslink.ro

Abstract: The aim of this paper is to study the induction heating processes by using optimal design with Genetic Algoritms (GAs) respectively fmincon and minimax optimisation methods. The optimisation is performed in order to obtain a high value of electrical efficiency of the inductor, as well a minimum non-uniformity of the temperature distribution in the workpiece. The optimisation is performed in two different ways: by using a simple genetic algorithm and with fmincon and minimax functions.

Keywords: induction heaters, numerical modelling, optimisation, simple genetic algorithm.

I. INTRODUCTION

The aim of this paper is to investigate the ways the evolutionary optimisation technics can be applied in the design of electromagnetic devices, in particular of the induction heaters in order to automate some parts of the designing process and to compare the obtained results with other optimisation methods, namely fmincon and minimax. For the electromagnetic problems, the optimal design is usually a multiobjective task where different specifications, often in conflict among them, have to be pursued with several design degrees of freedom, which are the design parameters. Also, the device parameters have to satisfy some technological and physical constraints. Therefore it is necessary to use a multiobjective optimisation strategy.

Due to the presence of multiple quasi-optimal solutions and to the typical complexity of electromagnetic computations, the automatic optimal design of electromagnetic devices is a very complex task.
The design process of the induction heating devices, takes into account the inductor's electrical efficiency,[1]. By this point of view, induction coils with a small coupling gap are more efficient than loosely coupled ones, so that the magnetic flux would flow through the workpiece surface, generating eddy currents. Higher frequency is also beneficial, [2],[4],[5]. Carrying out the design must consider the heating of the entire internal length of a workpiece, such as the temperature distribution would be as uniform as possible.
In this work considerations regarding simple genetic algorithm and fmincon and minimax functions optimization are presented, having the aim to increase the electrical efficiency, as well as to assure a uniform temperature distribution along the workpiece. These are important steps in the optimal design of the induction heating processes.
For optimisation, three methods where applied: simple genetic algorithm, in which the two objectives are combined in a single fitness function by using weghting coefficients, optimisation with fmincon function and optimisation with minimax function.

II. MODELLING THE INDUCTION HEATING DEVICE

The induction heater for cylindrical pieces was selected for this experiment (Figure 1).

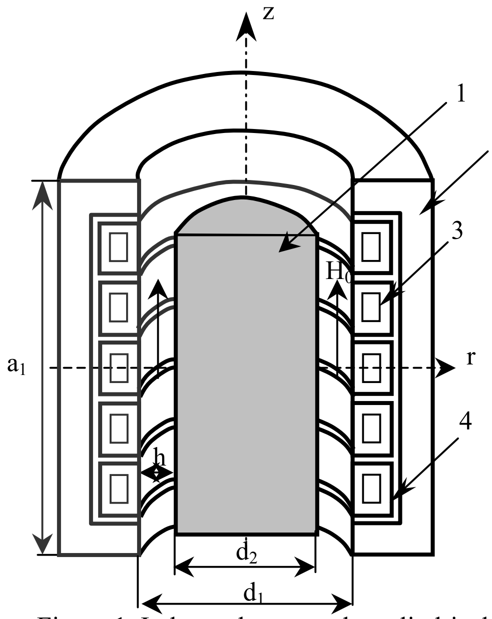

Figure 1. Inductor heater and a cylindrical piece, where: 1- piece; 2- magnetic coil; 3 – water; 4- turn

The heater is double sided, heating in longitudinal field without magnetic core. The phisical model of the heater is presented in Figure 1, [3].
The mathematical models of the electromagnetic and thermal fields, based on the specific laws of these phenomena are described by partial equations.
The mathematical model of the cuasi-static harmonic field is represented by the equation (1):

$$(j)\omega\mu_0\sigma A + \mathrm{rot}((1/\mu_r)\mathrm{rot}(A)) = \mu_0 J \tag{1}$$

The mathematical model of the transient thermal field θ(r, z, t) is described by equation (2):

$$\rho c * d(T)/dt + \mathrm{div}((-k) * \mathrm{grad}(T)) = QH \tag{2}$$

where ρc is the specific heat.

III. OPTIMISATION METHODS

In practice, the optimal design problem is usually formulated in terms of constrained optimization of a multi-objective scalar function, typically constructed as a weighted sum of different objective functions, which usually are competitive.
The obtained scalar function is then minimized or maximized inside a suitable search domain, taking into account the imposed constraints. Among the deterministic optimization methods that could be applied, it is to be mentioned the simplex algorithm and the golden selection method. But generally, the obtained scalar function has multiple local optima scattered in the admissible solutions space. Therefore optimization calls for global techniques able to explore the search space.
Evolutionary strategies are a family of algorithms widely used in global optimization problems and particularly; Genetic Algorithms (GAs) have a particular relevance, for their simple implementation and for their efficiency.

III.1. Genetic algorithms

Genetic algorithms (GAs) are global random search methods widely employed in optimization problems, or in problems where the gradient of a given objective function is not available. They compensate the lack of gradient information through a random exploration of the search space that evolves in analogy to the evolution in nature, [6].
The power of GAs consists in only needing objective function evaluations to carry out their search.
GAs consist in having a population of candidate solutions (individuals, chromosomes) to an optimization problem that evolves at each iteration *t* of the algorithm, called *generation*. The evolution of the species is simulated through a *Fitness* function and some genetic operators such as selection, crossover, and mutation.
Fitness function is a scalar value that combines the optimization objectives and is obtained in the evaluation step, when a problem specific routine returns its value. The fittest individuals will survive generation after generation while also reproducing. At the same time the weakest individuals disappear from each generation.
Individuals must be encoded in some alphabet, like binary strings, real numbers, and vectors.

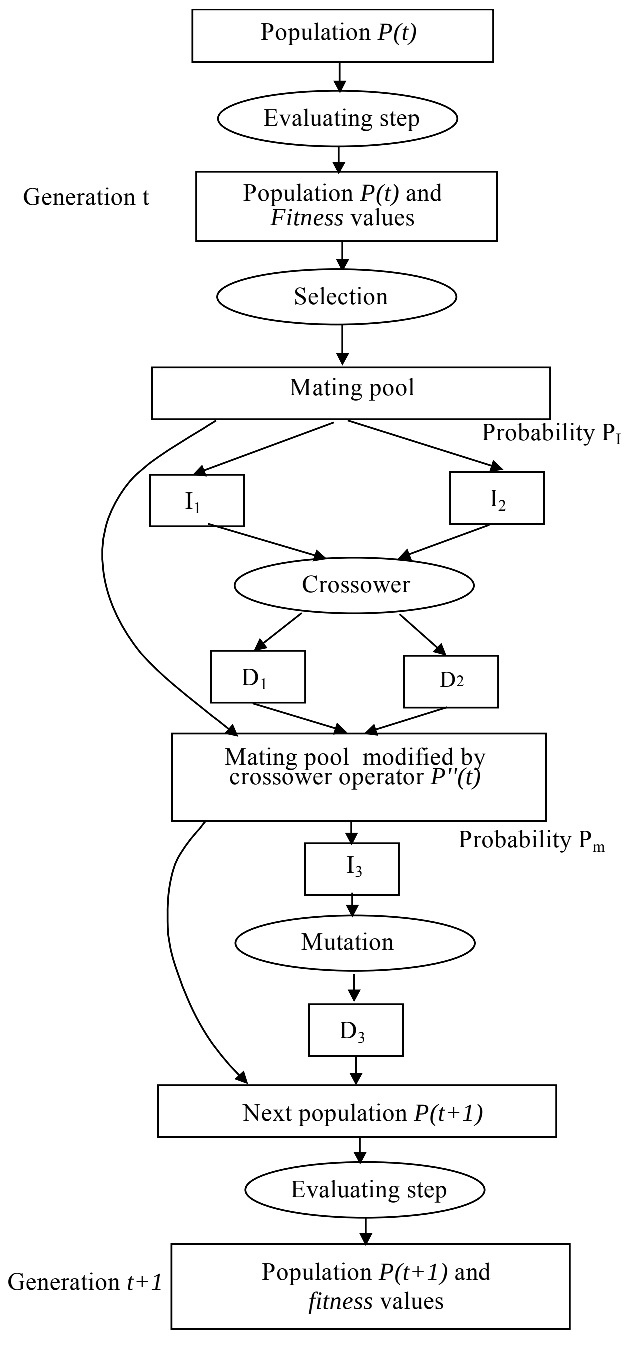

Figure 2. Creating the next generation in GAs

In a practical application of GAs, a population pool *P(t)* of chromosomes has to be installed and they can be randomly set initially. In each cycle of genetic evolution, a subsequent generation is created from the chromosomes in the current population, shown in figure 2, [8].

In the evaluation step, the Fitness value for each individual in the population is computed.
During the selection stage, a temporary population denoted "Mating pool" is created in which the fittest individuals have a higher number of instances than those less fit.
During the evolution, GAs employ *genetic operators* like crossover and mutation, [9].
The crossover is applied with the probability P_I. It randomly mates the individuals and creates offsprings D_1 and D_2 from two parents I_1 and I_2 by combining the parental genes and transferring them to the next generation.
The mutation operator is applied with a low probability P_m. It creates new individuals D_3, which are inserted into the population, by randomly changing the parent I_3.
The algorithm establishes the next generation P(t+1) and by this way, individuals of the original population are substituted by the new created individuals.
In essence, the procedure of a GA is given as follows:

1. *Generate randomly a population of chromosomes.*
2. *Calculate the fitness for each chromosome in the population.*
3. *Create offspring's by using genetic operators.*
4. *Stop if the search goal is achieved. Otherwise continue with Step 2.*

GAs can take into account the constraints by using different methods. The most used approach is to reject the infeasible individuals. Another approach is to use penalty functions. In this case, violation of constraints takes the form of penalties. The basic idea of this approach is to "punish" the fitness value of an individual whenever the solution produced violates some of the constraints imposed by the problem. In this paper, the first approach is used.

III.2. Fmincon optimisation

By using the Fmincon function from the Matlab Optimization Toolboxe [10], a function with several variables is to be minimized, by taking into account the constraints of the problem, as stated in relation (3).

$$\min\left\{\begin{array}{l} f(x)\middle|c(x)\le 0, ceq(x)=0 \\ A\cdot x\le b, Aeq\cdot x = beq, l\le x\le u \end{array}\right\} \quad (3)$$

The Fmincon algorithm.
For large scale optimization problems, the algorithm uses the confidence region procedure. Each iteration of the algorithm implies approximating the solutions of a large linear equations system by using the conjugate gradient method with preconditioning. fmincon uses a user-defined Hessian, or Hessian information for the objective function.
It approximates the Hessian using finite differences
For medium scale optimization problems, fmincon uses a Sequential Quadratic Programming method. In this method, a Quadratic Programming subproblem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration and a line search is performed using a merit function. The Quadratic Programming subproblem is solved using an active set strategy

Limitings.

- The objective function that is to be minimised and the constraints must be continuous functions.
- The obtained result is a local minima.
- The objective function and the constraints functions must return real values.

.

III.3. Fminimax Optimization

The fminimax algorithm uses a Sequential Quadratic Programming method. Modifications are made to the line search and Hessian. In the line search an exact merit function is used together with the merit function, [10].
The line search is terminated when either merit function shows improvement. A modified Hessian that takes advantage of special structure of this problem is also used. Using optimset to set the MeritFunction parameter to singleobj uses the merit function and Hessian used in fmincon.
Limitations

- The function to be minimized must be continuous.
- The constraints must be continuous functions.
- The objective function and the constraints functions must return real values.

IV. OPTIMIZATION OF THE INDUCTION HEATERS

IV.1. Optimization of the induction heaters by means of simple GAs

A simple GA implemented in Matlab is used. It has the following main features:

- Individuals can be encoded like binary strings, real numbers, and vectors of binary strings or of real numbers, or like permutation strings;
- The genetic operators are implemented according with the encoding scheme used;
- Randomly generates the initial population, but allows the use of an initial population specified by the user;
- Performs the GA's specific iteration;
- Includes the best performing individual of the parent generation in the new generation in order to prevent a good individual being lost by the probabilistic nature of reproduction;
- Allows the user to establish the GA's parameters: the size of the population, the type of selection scheme, crossover and mutation and the probability of applying the genetic operators.

The design variables of the problem are: the frequency *f*, the *Li* and *h*.
In order to apply genetic algorithms, it is necessary to use a coding scheme to the solutions of the problem. In this case, we used vectors of real numbers, having the form of relation (4) and presented in Figure 3.
[f Li h]   (4)

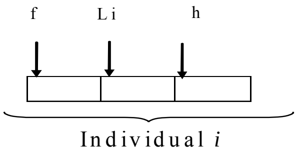

Figure 3 Individual encoding scheme

GAs implicitly perform maximisation of the Fitness function, which is a scalar value. While in this optimization problem two objectives are stated, maximization of the efficiency and minimization of the temperature ecart, the scalar Fitness value is obtained by using the weighting coefficients method. [7].

$$\text{Fitness}=w_1 Ob_1+w_2 Ob_2$$

or

$$\text{Fitness}=w_1 Ob_1+(1-w_1) Ob_2$$

Where Ob_1 and Ob_2 are both to be maximized, we will have a single w_1.
Since the efficiency takes values in the interval 0.1...0.9 [%], and ecart in 80....1000 [°C], the weighting coefficients are to be established in such a manner that both objectives contribute in equally sized quantities to the Fitness function.
Firstly the objectives were adimensionalised in order to remove the influence of the variation domain of the two objectives as in the following relation:

$$F_{obj} = w_1 \cdot \frac{(ecart - ecart_{min})}{(ecart_{max} - ecart_{min})} + (1 - w_1) \cdot \frac{(rand_{max} - rand)}{(rand_{max} - rand_{min})}$$

In the above relation, the value of ecart is to be minimized while the value of rand is to be maximized. By using the negative sign in the second term of the relation, F_{obj} is to be minimized.
Since GAs perform implicilty a maximization of the Fitness value, the minimizing optimization problem must be transformed into a maximizing one, for exaple by inverting the objective function expression, as in the folowing relation:

$$\text{Fitness} = \frac{1}{F_{obj}}$$

In order to establish the Pareto set that is the solution to the multiobjective optimization problem, the optimization run was repeteadly applied for different values of the weighting colefficient w_1=0...1.

The first step to start the optimization is to define the initial population. This is carried out, by generating individuals having the form of relation (4) and the randomly picked parameters as defined by relation (5).

$$\begin{aligned} &\text{frequency:} && f \in [\,250 \quad 3000]\ \text{Hz} \\ &\text{inductor length:} && Li/2 \in [\,0.10 \quad 0.145]\ \text{m} \\ &\text{the air-gap:} && h \in [\,0.001 \quad 0.015]\ \text{m} \end{aligned} \qquad (5)$$

Once the parent population is available, recombination allows for the creation of new individuals, based on the previous generation. In this work, we have chosen to use arithmetic crossover. This technique consists in taking two individuals from the parent population and using a random factor from the interval [0,1], so that the new offspring's parameter might be at any point between its parents' parameter values. This process of recombination is mathematically expressed in relation (6).

$$\begin{aligned} D_i &= \alpha \cdot I_i + (1 - \alpha) \cdot I_j \\ D_j &= (1- \alpha)\cdot I_i + \alpha \cdot I_j \end{aligned} \qquad (6)$$

where I_i and I_j are the parents, α is a random factor and D_i and D_j are the produced offsprings. The used crossover probability is 0.8.
Through the recombination operator a new population is created. All the parameter values have been calculated based on *inherited* values from the parent population. Therefore no *new* information has been inserted in the population but only *old* information has been recombined.
In order to introduce new information into the population pool, the mutation operator is used.
Mutation consists of slightly perturb ting the parameters of the offspring individuals. Uniform mutation was used, that randomly takes a parameter k of an individual and replaces it with a new value using a random factor from the interval [0.1]. This is expressed in mathematical term by relation *(7)*.

$$D_{ik} = L_{min} + \alpha \cdot |L_k| \qquad (7)$$

where D_{ik} is the k'th parameter of the produced offspring, α is a random factor and $|L_k|$ is the allowed domain for the given parameter. The used mutation probability is 0.1.
After applying the mutation operator, a mechanism that allows us to *select* which offsprings will conform the new population has to be implemented. The selection operator used is roulette wheel, the traditional selection function. The probability of surviving is equal to the fitness of a given individual, divided by the sum of the fitness of all individuals. In the implementation of the algorithm, simple elitism was also applied. This technique guarantees survival of the best individual.
The approached problem deals with two opposing objectives, one of them being the electrical efficiency denoted *rand*, and the other one, the uniform distribution of the temperature inside de heated piece, denoted *ecart*. The first one is to be maximized and the second one is to be minimized. Accordingly, the first objective becomes *rand* and the second objective is modified so that the obtained quantity is also to be maximized (8):

$$\begin{aligned} obj_1 &= rand \\ obj_2 &= \frac{1}{1 + ecart} \end{aligned} \qquad (8)$$

These objectives are normalized and adimensionalized in O_1 and O_2, so that finally we have to maximize these functions. The classical way of tackling such an optimization problem was used, which consists in converting multiple objectives into one objective, by the weighted sum approach, as given in (9).

$$\text{Fitness} = \alpha \cdot O_1 + (1-\alpha) \cdot O_2 \tag{9}$$

Since multiple objectives are converted into one objective, the resulting solution to the single objective optimization problem is usually subjective to the parameter settings chosen by the user. Moreover, only one solution can be found in one run.

Simple GAs were applied, with a population of 25 individuals and the genetic serch was run for 15 generations. Each run was performed repeadly with w_1=0; 0.1; ... 0.9; 1.

The results were stored in the Table 1 in the following form:

Col. 1, number of generation the solution was found
Col. 2, frequency f;
Col. 3, inductor lenght L_i;
Col. 4, air gap h ;
Col. 5, Fitness.

Usually, the best individual is the last returned by GAs.

Table 1. Solutions obtained with simple genetic algorithm

w_1	f [Hz]	L_i [m]	h [m]	ecart [°C]	rand [%]
0.0	1102.0975	0.1450	0.0010	204.8767	69.11
0.1	913.0529	0.1442	0.0010	164.6998	69.89
0.2	907.1050	0.1442	0.0010	163.0130	69.86
0.3	828.6229	0.1277	0.0012	132.6263	69.42
0.4	927.8151	0.1313	0.0010	129.2788	69.16
0.5	632.0527	0.1449	0.0010	116.9644	68.41
0.6	819.3478	0.1343	0.0034	114.4589	67.77
0.7	910.2839	0.1290	0.0021	116.3087	68.75
0.8	985.8491	0.1292	0.0055	112.0084	67.26
0.9	1420.4631	0.1224	0.0102	103.0639	63.79
1.0	2995.9575	0.1138	0.0115	93.1985	52.00

IV.2.Optimization of the induction heaters by means of fminimax and fmincon functions

This optimisation task is performed by using the fmincon and fminmax functions from the Matlab optimisation toolbox.

Theses functions are applied in two different ways. First, the starting point is randomly generated in the feasible search region. Secondly, we use as starting point a solution that was previously found by GAs and the other opimisation methods are applied in order to refine the search space.

The search domain of variables are given by relation (10).

The design variables are *f, Li* and *h*.

The two objective used in the design process are: e*cart* and *rand*.

- frequency $f \in [250\ 3000]$;
- inductor length $L_i \in [0.10...0.145]$; (10)
- the air-gap $h \in [0.001...0.015]$;

IV.2.a) Applying fminmax from randomly choosed starting point.

The starting point was randomly generated in the feasible search space. It was the following vector:
[1500 0.12 0.0012].

The runing time of the algorithm is about 2 hours.

Table 2 presents the solutions obtained by applying the minimax function.

Table 2. Solutions obtained with fminimax function started from a random point:

f [Hz]	L_i [m]	h [m]	ecart [°C]	rand [%]
3000	0.0100	0.0010	266.1226	54.99

IV.2.b) The starting point is a solution obtained by GAs in a previously run.

The starting point is given by the following vector:
[632 0.1499 0.0010]

Table 3 presents the solutions obtained by applying the minimax function.

Table 3. Solutions obtained with fminimax function started from a solution found by GAs in a previously run.

f [Hz]	L_i [m]	h [m]	ecart [°C]	rand [%]
250	0.145	0.010	110.8395	53.28

IV.2.c). The same optimisation problem as used in the previous paragraphs is to be solved, this time by using the fmincon function. Since this function from the Matlab Optimisation toolbox is a single objective optimisation method, in order to apply this method to the design problem, the weighting coefficients method was applied. The two objectives were combined into one single objective by using a weighting coefficient w_1.

In order to obtain the Pareto set representing the solution to this problem, the optimisation task is repeatedly aplied with different values for the weighting coefficient w_1.

The obtained results are presented in Table 4.

Table 4. Solutions obtained with fmincon functions:

w_1	f[Hz]	L_i [m]	h [m]	ecart [°C]	rand [%]
0.0	3000	0.1	0.001	173.7602	54.91
0.1	601.56	0.10967	0.001937	336.1191	61.91
0.2	601.56	0.10966	0.0010938	193.4602	68.10
0.3	625.02	0.11	0.0011	231.1738	67.65
0.4	711.2	0.112	0.0011256	259.6993	66.74
0.5	613.28	0.10976	0.0010969	232.9850	67.61
0.6	2831.1	0.1241	0.0019	293.0236	65.98
0.7	601.56	0.10963	0.0010937	233.7253	67.54
0.8	2754	0.1254	0.001711	293.0217	66.01
0.9	823.99	0.11406	0.0011406	233.4940	67.53
1.0	625	0.1100	0.0011	247.0095	62.59

V. CONCLUSIONS

Analysing the obtained results, it can be seen that applying optimisation by means of GAs lead to better solutions, comparing with fmincon and fminmax started from a randomly choosen point. Anyway, the best solutions are those given by fminmax started from a solution of GAs since a refinement of the search space is performed.

Additionally, applying optimisation methods in the design of the electromagnetic devices lead to better designs that behave with higher efficiency and a more uniformly distributed temperature.

The disadvantage that belong to all applied methods consists in the fact that electromagnetic design problems are very demanding in terms of computing resources, by requiring the resolution of a complex electromagnetic problem for each evaluation. When the complexity of the adopted model increases, the development time of a new device configuration is high. But, in recent years, the increasing computation power of personal computers makes this disadvantage lesser, so that it is possible to use stochastic algorithms effectively in many applications, such as in electromagnetic design problems. The development of methods and programs for numerical simulation are very importan tasks, since these make possible the simulation of inductor behavior before the construction, and so we can avoid the design errors.

REFERENCES

[1] T. Leuca, E. Vladu, M. Popa, " Using Genetic Algorithms Optimal Design of Electromagnetic Devices", Ed. Academiei Române, pp.319-327.

[2] T. Leuca, "Elemente de teoria câmpului electromagnetic, Mãrimi fizice, Legi, Teoreme", Litografiat Universitatea din Oradea, 1998

[3] V. Fireţeanu,M. Popa , T. Tudorache , E. Vladu "Numerical analysis of induction throgh heating processes and optimal parametrs evaluation", EMF 2003 Aachen GERMANY 2003

[4] A. Antero, (1987). "Analysis of induction motors based on the numerical solution of the magnetic field and circuit equations" . Acta Polytechnica Scandinavica. Electrical Engineering Series, No. 59, Helsinki

[5] Janne Nerg. (2000). "Numerical Modeling and design of static induction heating coils".

[6] J. H. Holland, (1992)" Genetic Algorithms", Scientific American Ed, pp 66-72.

[7] E. Vladu ,(2003) "Contributions to the applications of genetic algorithms in engineering", Politehnica University of Timisoara.

[8] C.A. Coello, "An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design". PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, 1996

[9] E. Vladu „Multiobjective optimization using genetic algorithms" Proceeding of EMES 2003, Oradea

[10] Optimisation Toolbox Matlab.