International Journal of Applied Electromagnetics and Mechanics 15 (2001/2002) 219–223
IOS Press

Object-oriented genetic algorithms for two-dimensional design optimization of the magnetic circuit of a mobile magnetic resonance device

Hartmut Popella* and Gerhard Henneberger
RWTH Aachen, Department of Electrical Machines, Schinkelstraße 4, 52056 Aachen, Germany

Abstract. The mobile usage of a magnetic resonance imaging device commonly known as MRI requires a handy kind of magnetic circuit device for production of the main magnetic field B_0 . This device is called a Mobile Universal Surface Explorer (MOUSE). The requirements of homogeneity and field strength for MRI applications as well as low weight make great demands on the geometrical shape of the magnetic circuit. By genetic optimization of several FEM models of the MOUSE a suitable solution shall be found which meets the demanded requirements. The paper presents a genetic optimizer for static computation of the MOUSE' field distribution and describes the implementation using object-oriented design patterns.

1. Introduction

The resolution of magnetic resonance imaging experiments depends on the homogeneity of the externally applied magnetic field $\vec{B}_0$ [2]. For mobile usage this field can be produced by highly remanent magnets as NdFeB. The homogeneity shall be inuenced by the MOUSE' geometrical shape. The basic principle of the NMR-MOUSE is described in [4]. An analytical solution of this field problem is hardly possible. So one must think of an intelligent empirical FE examination of this arrangement. The use of optimization techniques presents itsself. In [4] a gradient based optimization tool was used. This tool is implemented in the FEM program ANSYS. Solving the MOUSE problem by a gradient technique leads to very fast convergence of the FEM computation. An expanded examination of different design sets is prevented. So an optimization tool based on genetic strategies was implemented. During the evolution process genetic algorithms use more effciently the provided search domain depending on population diversity and selective pressure [3].

2. Theory of genetic optimization

2.1. Principles of genetic strategies

The following function $f(\underline{x}) > 0$ shall be maximized:

$$f(x_1, \ldots, x_k) \;:\; R^k \rightarrow R \;\; x_i \;:\; D_i = [a_i, b_i] \subseteq R$$

*Corrseponding author: Tel.: +49 241 807636; Fax: +49 241 8888270; E-mail: popella@iemrwth-aachen.de.

$$f(x_1, \ldots, x_k) > 0 \ \vee \ x_i \in D_i \tag{1}$$

Normally genetic optimization bases on binary coding. Here a method of using oat point arithmetic for genetic optimization is introduced. In genetic optimization theory each variable x_i is called a gene. The number of genes for a certain object function defines the size of search domain. K genes form a chromosome, a potential solution of $f(\underline{x})$. The chromosome structure is as follows (if using binary coding, each gene's oat point value must be replaced by its corresponding binary or gray code value):

$$\underbrace{\text{gene}_1|\text{gene}_2|\ldots|\text{gene}_{k-1}|\text{gene}_k}_{\text{chromosome}} \tag{2}$$

Several existing chromosoms v_i form the population of the actual iteration step.

$$v_1 \ldots v_i \ldots v_{\text{populationsize}} \tag{3}$$

The fitness value of the whole population is defined as follows:

$$F = \sum_{i=1}^{\text{populationsize}} \text{eval}(v_i); \ \text{eval}(\nu_i) := f(x_i) \tag{4}$$

$$\overline{F(t)} = \frac{F(t)}{\text{populationsize}} \tag{5}$$

The fitness value is an indicator for the optimization progress during the computation. The average fitness value should become greater in case of maximizing the object function or smaller for minimization problems. In order to achieve optimization performance chromosomes with a strong fitness must be taken over into the next population and allowed to reproduce. Recombi- nation and thus genetic variety is forced by genetic operators crossover and mutation.

2.2. *Genetic operators using oat point arithmetic*

In most optimization cases multidimensional search domains must be examined (e.g. the geometrical finite element shape of the MOUSE is a multidimensional search domain which is spanned by the keypoint coordinates of the FE model). Then binary coding is unsuited for these kinds of problems, may result in poor convergence and might be better replaced by oat point arithmetic. One must define genetic operators for arithmetic use.

2.3. *Arithmetic crossover*

- Chromosomes at iteration steps t and $t+1$:
  Chromosomes v, w at iteration step t: s_v^t, s_w^t
  Chromosomes v, w at iteration step $t+1$: s_v^{t+1}, s_w^{t+1}
- Definition of arithmetic crossover:

$$s_v^{t+1} = a \cdot s_w^t + (1-a) \cdot s_v^t$$
$$s_w^{t+1} = a \cdot s_v^t + (1-a) \cdot s_w^t \quad a \in [0;1] : \text{crossover factor} \tag{6}$$

The crosssover, factor might be a fuction of iteration step. Then it is called non-uniform crossover, otherwise uniform crossover [3].

2.4. Arithmetic mutation

The definition of arithmetic mutation is as follows whereas v_k^t is the actual oat value of the considered gene:

$$v_k^{t+1} = \begin{cases} v_k^t - (u_k - v_k^t) \cdot \left(1 - r^{(1-\frac{t}{T})\cdot b}\right) & : \text{random value } r < 0.5 \;\; r \in [0;1] \\ v_k^t + (v_k^t - l_k) \cdot \left(1 - r^{(1-\frac{t}{T})\cdot b}\right) & : \text{random value } r > 0.5 \end{cases} \tag{7}$$

- Upper interval limits of the considered gene: u_k
- Lower interval limits of the considered gene: l_k
- t : actual iteration step; T : max. iteration step
- b : correction value $b < 0.1$

2.5. Selection strategy of genetic algorithms

The implemented algorithm's (called ModifiedGA) selection strategy corresponds to the elitest model. All chromosomes of the actual population are divided into three sets according to their fitness value: weak chromosomes, average ones and strong ones. During the next iteration step weak chromosomes die off, average chromosomes are adopted and strong ones are even allowed to reproduce by the presented genetic operators. Another simple genetic algorithm is described in [1] and realized as SimpleGA in the optimization tool. Other implemented algorithms work with varying population sizes. Their functioning bases on SimpleGA or ModifiedGA by passing on those algorithms' characteristics and adding new information of chromosomes' lifetime.

3. Object-oriented implementation

The genetic optimizer is implemented in an object-oriented environment. By using object- oriented paradigms it is possible to design classes which realize genes, chromosoms and populations. The object-oriented approach submits the simple extension of different genetic algorithms which work on data structures for genes, chromosomes and populations. A static class diagram can be used in order to express the functionality of the realized genetic optimizer with constant population size. The object-oriented implementation simplifies the integration of new algorithms by specialization of existing classes for genetic optimization.

4. Performance of the implemented optimization tool

Before applying the developed genetic tools to design optimization of FEM computated MOUSE models they have to be applied to functions with known analytical solutions of existing extrema. The oscillating test $-x \cdot \sin(10\pi x) + 1$ function SimpleGA algorithm. ModifiedGA converges very fast and finds the global minimum.

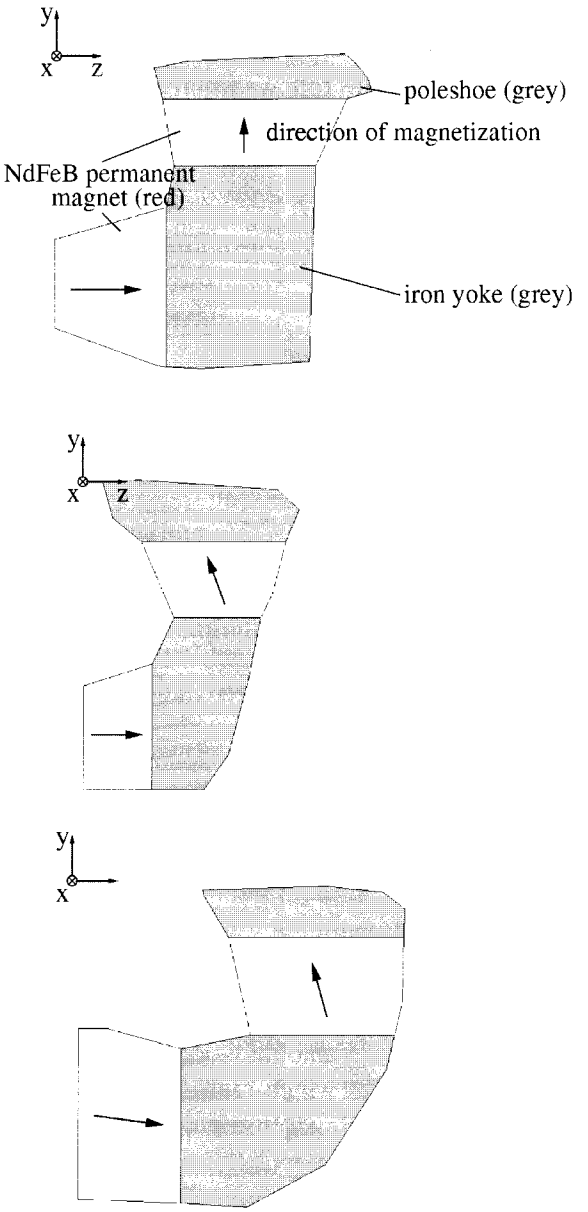

Fig. 1. A selection of genetically found FE MOUSE shapes.

5. Application of the optimization tool to the problem of the NMR-MOUSE

Finally the optimization tool shall be used to find suitable geometrical shapes for the NMR- MOUSE. The object function is defined as the gradient strength decline. Each iteration step yields a more or less best chromosome (a suitable geometrical shape for FEM calculation) per population. Figure 1 demonstrates a selection of MOUSE models consisting of four chromosomes which establish the start population.

6. Conclusion

In this paper the implementation of an object-oriented genetic optimization tool has been introduced. The application of genetic strategies to the optimization of the NMR-MOUSE geometry allows to examine a wide range of different FEM models which cannot be found by gradient based tools or simple parameter variation. The consequential object-oriented approach eases the management of large programming tasks.

References

[1] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, 1989.
[2] J. Jin, *Electromagnetic Analysis and Design in Magnetic Resonance Imaging,* CRC Press, 1999.
[3] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs,* Springer-Verlag, 1999.
[4] H. Popella, Design and optimization of the magnetic circuit of a mobile nuclear magnetic resonance device for magnetic resonance imaging, *Compel* **20**(1) (2001), 269–278.