

Searching for the best optimizer for an automated CAD system

Abstract. It is recommend not to exceed the abstract length more than four lines. It is recommend not to exceed the abstract length more than four lines. It is recommend not to exceed the abstract length more than four lines. It is recommend not to exceed the abstract length more than four lines.

Streszczenie. Streszczenie nie powinno przekraczać czterech linijek tekstu. (Wybór procedury optymalizacyjnej dla systemu CAD)

Keywords: CAD, optimal design, deterministic algorithms, evolutionary algorithms
Słowa kluczowe: systemy CAD, optymalizacja, algorytmy deterministyczne, algorytmy ewolucyjne

Introduction

Automated Computer Aided Design (CAD) systems are currently commonly used in many areas of engineering. Use of such tools allows to design and prototype final products with better performance more effectively. One of the crucial components of many CAD systems is an automated optimization procedure. We assume here, that the user of such a system is responsible for general design decisions, and the system's role is to find the best parameters for the chosen design. Such tasks are usually formulated in form of one or many cost functions and the best values are found in process of minimization of these functions. Nowadays the most common choice of cost function minimizer is one of the stochastic algorithms. Such methods are simple to implement, they search for globally best solution and they may be applied for practically any cost function: even discontinuous and multimodal.

Here we shall present a case study of process of selecting such minimizer for an example CAD system. The stochastic optimizer will be compared to the deterministic one in terms of the simplicity, robustness, quality of the results and speed.

Example CAD system

In the paper we use a CAD system for electromagnetic flow meter (EFM) design as an illustrative example. In the EFM externally induced magnetic field \mathbf{B} acts on a moving conducting liquid and creates creating electric field \mathbf{E} which can be measured as a potential difference on the electrodes \mathbf{e}_1 - \mathbf{e}_2 situated on the banks (see Fig. 1). It can be shown, that

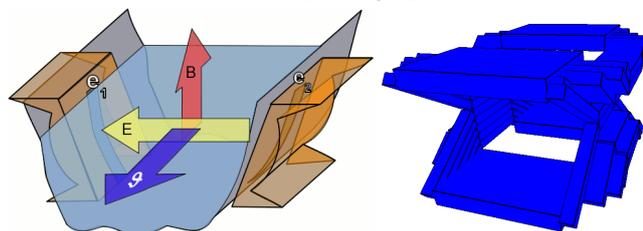


Fig. 1. An electromagnetic flow meter: principle of operation (left) and sample shape of the coil (right)

for the proper distribution of the magnetic field \mathbf{B} (for example uniform perpendicular component) the signal can be made proportional to the total flow [1]. Thus the optimal design of the EFM is usually formulated as a search for the coil shape.

A dedicated CAD system—Flow Meter Design Tool (FMDT) [3] is aimed at helping to design and test EFM prototypes. A designer can use FMDT to manage several alternative projects, design and optimize coil shapes and test their performance by numerical simulation.

From our perspective the most interesting part of the FMDT is the *coil designer*. It can be divided into two components: the *pre-designer* which helps to propose starting topology of the coil, and the *optimizer* which controls the optimization routine to calculate the best dimensions for the proposed topology.

To explain the optimization task it is necessary to present how the coil is described in FMDT. Figure 1 presents a 3D view of the optimized coil. It can be noticed that the main, working part of the coil is formed of rectangular blocks. This assumption comes from the technological requirements for realization of the designed shape. The designer selects a basic topology, number of sections, initial lengths and widths of sections. The initial coil may be evaluated with field simulation to check the magnetic field strength and predict signals for preselected flow(s).

The optimization task

The following design parameters were used in experiments: widths of all sections, length of the working part, total depth of the coil and shape (width or height) of the side-to-side connections. The objective function can be defined in several ways [2]. We have used a postulate of uniform distribution of the vertical component of magnetic flux density. Thus the cost function is defined as

$$(1) \quad f(\mathbf{d}) = \frac{1}{N} \sqrt{\sum_N (B_{y,i} - \bar{B})^2},$$

where \mathbf{d} is a vector of the design parameters, N is the number of control points within the channel, $B_{y,i}$ is the vertical component of magnetic flux density at the i -th node and \bar{B} is the desired value of B_y . The control points are spread on a uniform lattice grid in the channel.

For experiments analyzed in this paper we have designed a coil for rectangular channel 15 centimeters wide and 11 cm deep. We have assumed, that the controlled part of the channel is 5 cm long. The internal width of the coil was 16 cm. The working part of the coil contained 20 blocks and working parts were connected above and below the channel. The initial width of all blocks was 1 cm. Current density was set to $1e6 \text{ A/m}^2$.

Results of the magnetic field simulation for the initial coil shape is shown in Fig. 2. The absolute value of B_y is in the range of 3.89 mT to 7.223 mT. After optimization process we would like to obtain possible uniform field close to 7 mT.

We have used test problems with the same initial coils, but slightly different constraints. We have started with the initial coil, presented above, and allowed to change the shape and the total length of the coil. This resulted in 21 design

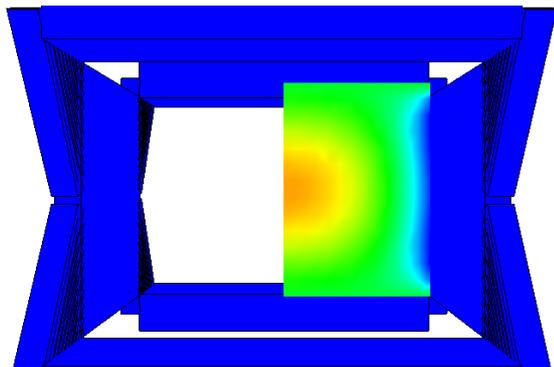


Fig. 2. Initial coil (3D view) and B_y distribution at the central cross-section of the channel

variables: 20 representing the widths of sections and the last one representing the length of the working part of the coil. In the first problem (Problem A) the widths were allowed to change in range of 1 mm to 3 cm, in the second one (Problem B) the upper bound was extended to 5 cm. Constraints of the coil length were the same: from 5.01 to 15 cm.

Experiments with Levenberg-Marquardt algorithm

The deterministic Levenberg-Marquardt (LM) optimizer, which was used in FMDT, was implemented with help of the public domain library `levmar` [4]. The Jacobian of the cost function was calculated numerically. Two examples of optimized shapes are shown in Fig. 3.

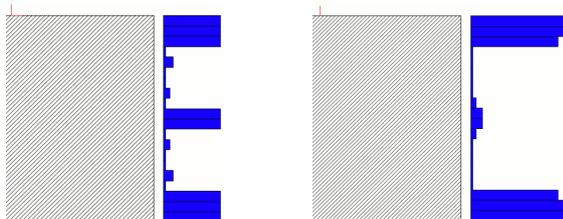


Fig. 3. The results of the deterministic optimization: solution of Problem A on the left side, Problem B on the right side

Experiments with different starting points (initial shape of the coil) are illustrated in Fig. 4. It may be noticed, that differences in the coil shape are minor, and the resulting field distributions are as well very similar. In our opinion this variation originates from numerical calculation of the Jacobian.

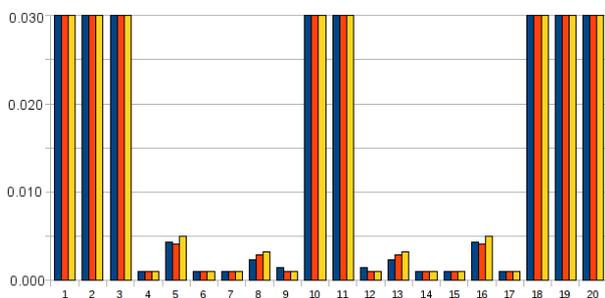


Fig. 4. Comparison of results of LM optimizer for Problem A with different starting points: blue – algorithm started from upper bound, orange – from the lower bound, yellow – from the midpoint of lower and upper bounds

Giving more freedom to the optimizer in the Problem B resulted in a different coil shape (presented in the right side of Fig. 3) and better (more uniform) distribution of B . Changes of the starting point resulted in small changes of the optimal shape. The effect is similar to the one which is presented in Fig. 4.

Experiments with genetic algorithms

The base for the genetic algorithm optimizer was the open source library `GAlib` [5] which contains a set of C++ “genetic” objects. On the basis of these objects the typical genetic algorithms was implemented. We have used real coding, roulette wheel selection method and elitism to preserve the best solution found. We have tested generational, steady-state algorithms and deterministic crowding niching. On the basis of all conducted experiments (few hundreds of optimizations) we are not able to say, which version of GA performs best.

We have experimented with different parameters of the GA algorithm. It was quite obvious, that the size of population can not be small for 21 design variables. For practical reasons we wanted to keep acceptable (less than 1 hour) time of solution. This set upper limits for the size of population and the numbers of generations. In Fig. 5 the results of two example experiments are presented. Both results were obtained in 100 generations of 400-individuals population.

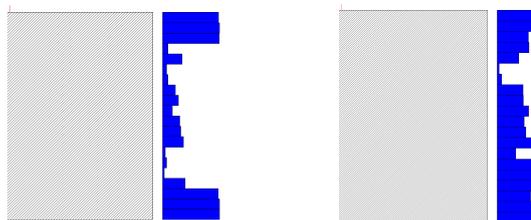


Fig. 5. The results of the genetic optimization of Problem A (left) and Problem B (right)

Comparison Fig. 3 and Fig. 5 shows clearly, that the genetic optimizer produces much worse results than the deterministic one. All shapes obtained with the genetic algorithm are less smooth and asymmetric. Change of the GA parameters usually leads to a different final result. The GA results differ much more than solutions obtained with multistart technique by the Levenberg-Marquardt optimizer. It can be seen in Fig. 6 where the results of three different runs of GA solving the Problem A are compared. One can notice that while

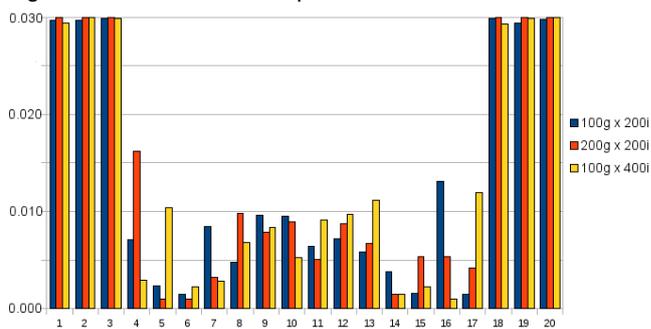


Fig. 6. Comparison of the results of GA optimizer for Problem A and different GA parameters: : blue – 100 generations of 200 individuals, orange – 200 generations of 200 individuals, yellow – 100 generations of 400 individuals

width of the upper and lower coil blocks stick to the constraints (what is similar to the results obtained with the LM optimizer), the widths of blocks in the inner part of the coil differ quite substantially. The final result can be improved by using more generations of bigger populations, but time of computations is then unbearable. The influence of the population size and number of generations on the quality of the final result is not obvious and it is very difficult to propose the minimal values which will guarantee satisfactory results of optimization. To illustrate this behavior one can study the results presented in Fig. 7. It is quite obvious than the result obtained with 50

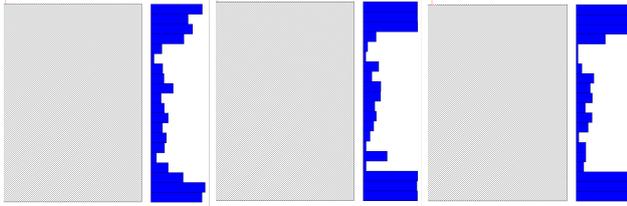


Fig. 7. Comparison of the best Problem A coil shapes obtained with the GA optimizer for different number of generations and same size of populations. From left to right: 200 individual population running for 50, 100, 200 generations

generations of 200-individual population is the worst one, but comparison between the results obtained with 100 and 200 generations is not simple at all. We shall compare them using field histograms, showing what proportion of the channel space is covered by the selected values of magnetic flux density. Such histograms for both discussed solutions are presented in Fig. 8.

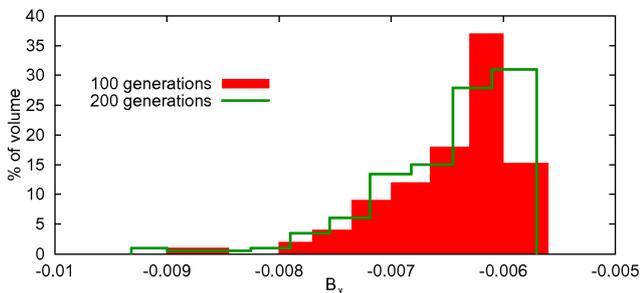


Fig. 8. Comparison of the field histograms for the best coils obtained with the GA optimizer for different number of generations and same size of population (200 individuals)

The good histogram should: a) be possibly narrow and b) exhibit the maximum for the flux density close to the desired value. For the desired value of B_y set to 7 mT the solution obtained with 200 generations could eventually be chosen as the better one, but it is not obvious at all.

Quality of the results

For comparison to the results presented in Fig. 8 let us show the field histograms for the best coils found by the deterministic optimizer, shown in Fig. 9. Comparing these histograms with those in Fig. 8 we can see that the coil found with the LM optimizer is better.

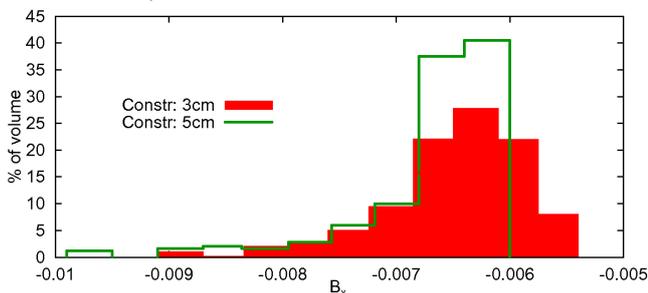


Fig. 9. Field histograms for the best coils obtained with the LM optimizer

Histograms presented in Fig. 9 confirm that Problem B,

with more loose constrains, allows us to find a coil which generates better field. As it could be expected, this better coil will be also more expensive in terms of production and application. Figure 3 shows that this coil needs more wire turns what causes more material and more feeding power.

Conclusion

Optimal design problem formulated in terms of a cost function may be solved by a deterministic or by a stochastic algorithm. The former approach is usually faster, but it requires continuous, differentiable cost function and it can converge to a local minimum. Stochastic algorithms in theory search for the global minimum and are much more tolerant. The minimized cost function may be multimodal and discontinuous. However, the biggest disadvantage of stochastic algorithms is their computational burden. In practice they require thousands of cost-function evaluations what makes them practically slow, comparing to deterministic methods. Speed-up of such algorithms can be obtained by reduction of the population size or number of generations, but this can seriously decrease the quality of obtained results.

The practical example presented in this paper suggests that application of the deterministic optimizer should be always considered. If the cost function is differentiable and its gradient may be calculated numerically then the implementation of the deterministic optimizer is not more difficult than the implementation of the evolutionary one. Moreover the optimal design will be obtained much faster with the deterministic optimizer and its quality will be much better.

Comparison of two example desing problems (A and B) and their solutions suggests that the optimality criterion should be forced not only by the field quality, but also by some additional criteria: costs of production and exploitation of the device. This, however, needs different, multiobjective approach to the optimal design.

Authors: Ph.D. Jacek Starzyński, Ph.D. Robert Szmurło, M. Sc. Piotr Rowiński, Prof. A. Michalski, Prof. S. Wincenciak, Institute of Theory of Electrical Engineering, Measurement and Information Systems, Faculty of Electrical Engineering, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warszawa, Poland, email: jstar@iem.pw.edu.pl, Prof. A. Michalski, Ph.D. Zbigniew Watral, Institute of Electronic Systems, Military Academy of Technology, ul. Kaliskiego 2, 00-489 Warszawa, Poland, email: zwatral@wel.wat.edu.pl

REFERENCES

- [1] Michalski A., Starzyński J., Wincenciak S.: Optimal design of the coils of the electromagnetic flow meter, IEEE Transactions on Magnetics, 34(5), pp. 2563–2566, Sep. 1998.
- [2] Michalski A., Starzyński J., Wincenciak S.: 3D Approach to Design the Excitation Coil of an Electromagnetic Flow Meter, IEEE Trans. Instrumentation and Measurement, 51(4), pp. 833–839, 2002.
- [3] Starzyński J., Szmurło R., Michalski A.: Computer-Aided Design Tool for Electromagnetic Sensors, IEEE Instrumentation and Measurement Magazine, 12(3), pp. 28-33, Jun. 2009.
- [4] M.I.A. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] <http://www.ics.forth.gr/~lourakis/levmar/>, Jul. 2004, Apr. 2009. [Accessed on 31 Jun. 2009].
- [5] Matthew Wall, GAlib A C++ Library of Genetic Algorithm Components [web page] <http://lancet.mit.edu/ga/>. [Accessed on 31 Jun. 2009].