



HAL
open science

Optimization of the boundary conditions by genetic algorithms

Jean-Luc Marcelin

► **To cite this version:**

Jean-Luc Marcelin. Optimization of the boundary conditions by genetic algorithms. International Review of Mechanical Engineering, 2012, 6 (1), pp.50-54. hal-00700103

HAL Id: hal-00700103

<https://hal.science/hal-00700103>

Submitted on 22 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimization of the boundary conditions by genetic algorithms

J.L. Marcelin

Abstract – *This work examines the possibility of using a stochastic method, called the genetic algorithm for the optimization of boundary conditions in finite elements calculations. The examples show that using genetic algorithms in order to optimize boundary conditions is an efficient way. Copyright © 2012 Praise Worthy Prize - All rights reserved.*

Keywords: *optimization, boundary conditions, genetic algorithms.*

I. Introduction

In numerous application cases (e.g. taking workpieces in machining, vibrations of a mechanical structure), the optimization of boundary conditions (location and nature of the boundary conditions) can bring an interesting improvement of the studied structure's mechanical behavior. These structures are most often calculated with the finite element method, and within this framework, the calculation of the sensitivities with regard to the boundary conditions remains enough complicated due to the discrete nature of the problem, and for example, in shape optimization, the boundary conditions problem is even most often eluded. Besides, the deterministic methods of optimization, called gradient methods, need a reliable calculation of these sensitivities. Some other stochastic or probabilistic methods of optimization are currently in vogue, like the simulated annealing method or that of genetic algorithms, which main benefits are they assure convergence without the use of derivatives, and can be used with possibly discrete variable and non-derivable functions. The detractors of this method point up without reason the high number of calculations, especially in the case of an analysis method of finite element type.

The author has a great experience in genetic algorithms ([1] to [6]). Few researches have been made on the boundary conditions optimization, and in the structural mechanics field, there are almost none. The [7] work, described below, deals with the optimization of the boundary conditions in electromagnetism. In [7], a methodology based on the genetic algorithm is proposed to determine the equivalent impedance boundary condition for corrugated material coating structures. We have found only two papers at our knowledge in the structural mechanics field [8] and [9], and only one [9] with the use of genetic algorithms. In [8], optimization of boundary conditions for maximum fundamental frequency of vibrating structures is done. In [9] the use of genetic algorithms for the selection of optimal

support locations of beams is presented. Both elastic and rigid supports are considered. The approach of adapting the genetic algorithms into the optimal design process is described. This approach is used to optimize locations of three supports for beams with three types of boundary conditions.

This article relates to fixed geometry and is to show that the optimization of boundary conditions is feasible by combining a genetic algorithm and the finite element method (the optimization of boundary conditions in shape optimization will be dealt with in future research). Indeed, in the case of a fixed shape and because of the characteristics of the finite element method, the calculation volume can be considerably reduced. The main reasons that are to be explicated in this work are the following: the stiffness matrix is calculated and assembled once and for all; in the case of a structure for which some boundary conditions can be fixed and other can be variables (i.e. entering in the optimization framework), it would be possible to triangularize the stiffness matrix once and for all, and to take into account the variable boundary conditions thanks to a penalization process of the energetic functional, to be minimized by the boundary conditions. In such conditions, even if the number of analyses is still important, the calculation time will remain reasonable because the analyses won't be systematically complete. Various examples will aim at showing that the implemented process helps in optimizing the boundary conditions and is fairly efficient.

II. The methods used

II.1. Genetic algorithms

The genetic algorithm method has been used several times within the various problems of mechanics. These algorithms were found to be very efficient, as in the case of the damping maximization of composite beams or plates or as somewhat diverse issues. The interest of these algorithms has also been showed in the difficult

case of the optimization of gears. The genetic algorithms are now well known and this article is not to introduce them in details nor generally.

Although it may seem so, the genetic algorithm method is not magic at all. It is part of the methods called "stochastic". The most famous of this kind of method is the already old simulated annealing. The main benefit of these methods is that they operate simultaneously on a sample of the solution space. The genetic method differs from simulated annealing due to the operators used to make this population sample evolve. The convergence is always ensured toward an extremum which is not necessarily the absolute extremum, but which is more likely to be absolute than if the conventional gradient method is used. Actually, a stochastic method explores more largely the solution space.

II.2. Optimization of the boundary conditions

This kind of optimization consists in combining a standard calculation program by finite elements (FE) (called thereafter analysis program) and the genetic algorithm.

The analyze program is a standard FE code. This code is simply to be called each time the genetic algorithm must estimate the cost function for a given chromosome. This is done for all the individuals of the population; consequently, for example, for 20 individuals and 30 generations, there will be 600 finite element half-analyses (the total stiffness being calculated once and for all), which is relatively low compared to the 2^{20} possible solutions. On the opposite, for the various tests that are done, especially those introduced after, there was not necessary to implement a penalization strategy of the "total potential energy" functional by the imposed boundary conditions, because convergence was fast enough. The programmer work simply consists in drafting a "pre-analysis" program that can decode the chromosome in question and that can automatically modify the finite element data file accordingly, and then in creating a "post-analysis" program that can extract the cost function from the finite element result file. Both these programs, as the calling of the finite element code, are built in the genetic program that drives the process.

Choosing the coding and the objective function

The problem contains two difficulties: First, the implementation of a solution code in the form of a simple and efficient chromosome and then the development of an objective function. The most generally used code is simple and natural (it has variants that are to be set forth in the examples): It can use the often used code for the boundary conditions in finite element programs, 0 being a free freedom degree and 1

a fixed freedom degree. The various codes of the concerned nodes are arranged end to end in a chromosome that is made of n binary digits that correspond to the n degrees of freedom that can be fixed. When it comes to the objective function, it depends on the posed problem. The first two examples are static cases, where the aim is to minimize the maximum displacement, or to minimize a deformation or a stress; the third example is a dynamic case, and the objective is to maximize the first natural frequency; it is also possible to try to remove two resonance frequencies. A lot of other choices are possible, such as multi-objective functions or penalizing the objective function by limitations.

Obtained results

Before each use, the genetic algorithm asks the user to specify the values of the following parameters:

the number of individuals contained in a population,

the maximum number of generations,

the chromosomes length,

the crossover probability,

the mutation probability.

It is clear that the algorithm gives best results when the chosen values for the first two parameters are high (within the limits of capacity of the used hardware). Practically speaking, the number of individuals contained in a population will be around 1 to 5 times the number of digits contained in a chromosome. However, the crossover and mutation probabilities are more difficult to choose. It has already been said that mutation is a far less frequent phenomenon than crossover; in [2], it is recommended the following values:

$$P_{\text{crossover}} = 0,60 ; P_{\text{mutation}} = 0,03$$

These recommended values come from a numerical experimentation on numerous examples. In any event, the crossover probability must be clearly superior to the mutation probability because mutation is less frequent. For example, if any mutation is removed, the algorithm yet converges toward an extremum but it is unlikely to be the absolute extremum. Theoretically speaking, convergence is obtained when all the cost values of a population stabilize around a maximum value. Practically speaking, convergence is rather slow, with ebb and flow, due to the very nature of the algorithm. The user only has to stop the process when the maximum value of a population cost does not evolve anymore; he then manually selects the most interesting individual(s) of the final population to compare their

benefits.

III. Examples

Test 1

The first very easy, static test is made with the axisymmetric workpiece (of CL axis and z symmetry) illustrated on figure 1 and aims at verifying and making the implementation of the used techniques reliable. The stiffness has been calculated once and for all but no penalization has been applied to the boundary conditions; since the calculations are fast enough for the tests, this procedure has never been implemented.

For this test, the chromosome is a 10-binary-digit string, the first 5 digits are the codes of the boundary conditions of the 5 nodes that can be locked following z, and the following 5 are the codes of the boundary conditions of the 5 nodes that can be locked following y; therefore, the chromosome 1011001000 corresponds to the boundary conditions applied to nodes 1, 3, 4, and 7.

There are 2^{10} possibilities. The objective is to minimize the d displacement of the node to which forces are applied. Since the genetic algorithm actually seeks the maximum of an objective function, the chosen objective is to maximize the 1-d function. The interest of this test is that the optimal solution is known: It is of course the 1111111111 chromosome, but the test helps in validating the process and in estimating how many steps are necessary for the genetic algorithm calculation to get this solution. We take here 40 individuals per population. The number of individuals in a population is usually around 1 to 5 times the chromosome's size (here the number of digits). The maximum is reached in only 5 generations (for the crossover and mutation probabilities provided in the last part), which corresponds to 200 half-analyses or a bit less (because a solution that appears several times during the process is calculated once and for all) and which is low compared to the possible 2^{10} combinations. Non-consistent convergence is characteristic of genetic algorithms because the best individual of each population may very unlikely be eliminated; besides, if the algorithm is forced to keep only the bests, the method is not probabilistic anymore and the algorithm may be more efficient or diverge in some cases. Besides, if optimization is launched again with the same parameters, the obtained convergence is not at all the same, because the process is totally random.

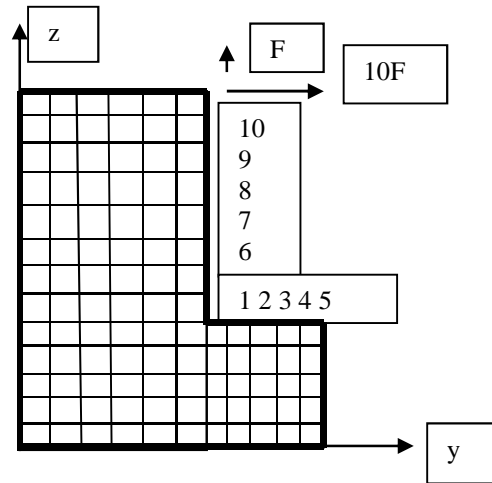


Figure 1

Test 2

This test illustrates a first industrial application in the taking of workpieces in machining, always with simple data that help in validating and checking the implemented strategy.

The quality of the workpiece depends on the deformations caused by the machining, because of the machining process itself or because of the holding process of the workpiece on its support. The chosen workpiece is of z axis revolution and is illustrated on figure 2. There are three surfaces to be machined, S1, S2, and S3. For this test, the calculations are only done for the S2 surface.

Contrary to the preceding test, where the 10 selected nodes could be locked, the clamping chuck can only be applied to one of the 8 possible nodes (nodes 9 to 16); the spindle stopper can be applied to one of the 8 nodes, numbered from 1 to 8. The test remains easy and calculates the genetic algorithm's behavior because the optimal solution can be forecasted and the number of possible solutions is limited, which would not be the case with a thinner mesh. The same type of code that in the previous test can be taken, that is to say that the first 8 chromosome digits concern the nodes 1 to 8, but the possible number of 1s in the algorithm is limited to 1 in this part of the chromosome; the 8 following digits are for the nodes 9 to 16, but any chromosome having a number of 1 greater than 1 in this part will be removed from the process. For example, 100000001000000 is an acceptable chromosome. This code type has not been kept for this example because it leads to 16-digit-long chromosomes and assumes the genetic algorithm is modified. Another type of possible coding is to build a 2-decimal-long chromosome; for example, 29 means that the nodes 2 and 9 are subjected to boundary conditions; the first digit varies between 1 and 8, and the second one between 9 and 16.

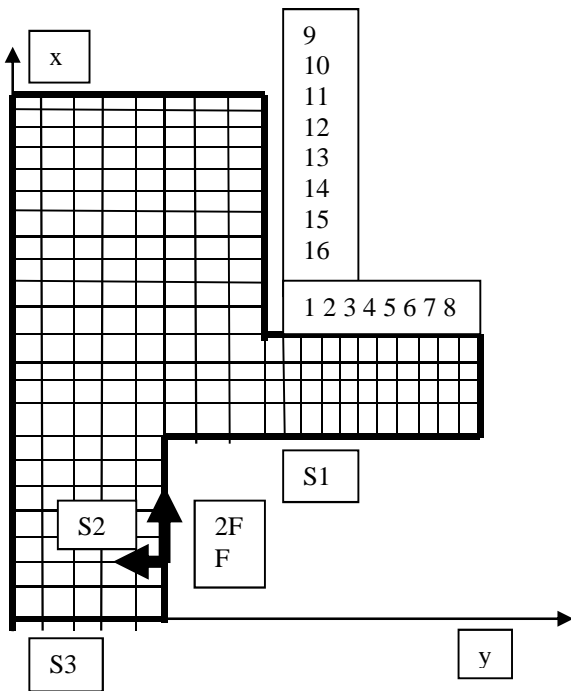


Figure 2

The chosen code in this example uses 6 binary digits, as in 100011. The decoding is done as followed (let's recall that the decoding program and that of modification of the finite element data file is to be designed by the user for each new example and must be placed immediately before the analysis): The first 3 digits give the code of the forced node 1 to 8, according to the following correspondence: 000 (node 1), 001 (node 2), 010 (node 3), 011 (node 4), 100 (node 5), 101 (node 6), 110 (node 7), 111 (node 8), and the following 3 digits provide the forced node 9 to 16 code, according to the same type of correspondence; therefore the example 100011 matches to the forced nodes 5 and 12. Of course, this example is still an easy test because only 64 combinations are possible and they can all be calculated to reach the problem optimum. The objective is to minimize the maximum equivalent deformation or the equivalent Von Mises stress that appears where forces are applied. The best solution found by the genetic algorithm is the combination of nodes 8 and 16, for which the Von Mises stress equals to 17.009 daN/mm². This result is found after a dozen finite element calculations (and from the second generation for a 6-individual population). In contrast, the genetic algorithm can be instructed to find the less good solution: the program is launched again with the objective of maximizing the main Von Mises stress; and this less good solution is the combination of nodes 1 and 9; for which the Von Mises stress equals to 17.195 daN/mm². The test remains easy because the mesh size

is limited. It could be more complicated if the mesh was thinner and if the genetic algorithm was instructed to find a compromise solution that would be valid to machine the S1, S2 and S3 surfaces.

Test 3

This test takes up the dynamic test offered in [8] and helps in validating the implemented strategy, once again on an easy case. The chosen example is that of a square plate, measuring 30.5 cm with a thickness of 0.328 cm in deflection vibrations (Young's modulus 73.1 GPa, density 2,821 kg/m³). This plate rests on 4 points that are located symmetrically on the diagonals (figure 3). The objective is to find the optimal location of the supports, maximizing the first fundamental frequency.

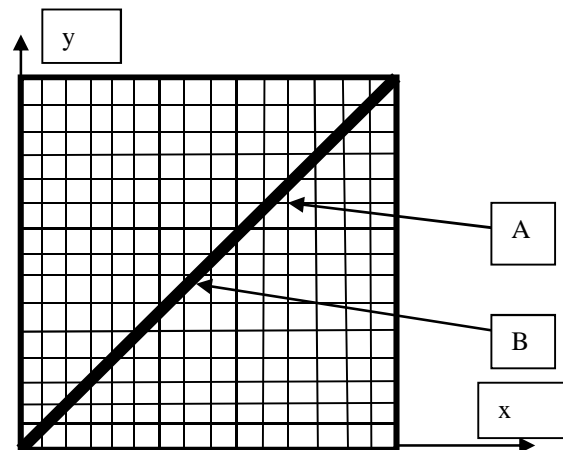


Figure 3

In [8], this problem is solved with a conventional gradient method, from a calculation of the frequencies' sensitivities with regard to the boundary conditions location. Since this is only the first symmetric mode of deflection, only a quarter of the plate is meshed. [8] finds two equivalent optimal points (A and B on figure 3) that correspond respectively to frequencies of 169.46 Hz and 169.67 Hz. Actually, In [8], it is only used a 36-element mesh for the whole plate and a study with thinner meshes has shown that the optimum is actually located between the A and B points. This test is often used in the literature; all the authors find that the optimal point is located between A and B.

Implementing a genetic algorithm strategy assumes that the support point location is coded under a chromosome on the main diagonal of the quarter of the plate. With the chosen mesh, that is 15 X 15 elements, there are only 16 possibilities that can all be calculated

to get the reference solution that actually corresponds to the points 7 and 8, with frequencies around 205 Hz. The code of the 16 possible points is simply a binary one: 0000 corresponds to the node 1, 0001 to the node 2, 0010 to the node 3 and so forth until the node 16 (1111). Let's recall that the objective is to maximize the first frequency. For a 4-individual population, a 2-individual population (0110, node 7 and 0111, node 8) is obtained after the thirtieth generation, which shows the genetic algorithm convergence, but its efficiency is more convincing with longer chromosomes (as in the example 1).

IV. Conclusion

This study has shown the efficiency of genetic algorithms in responding to the problem of the optimization of the boundary conditions in finite elements. This study is above all a feasibility study and will soon be complemented by industrial examples. The study can easily be spread to other fields than mechanics; for example, in thermal science, it would be easy to design chromosomes containing not only the information on the boundary condition type, but also that regarding the condition value to be optimized (flow value, heat transfer coefficient value); it could also be applied in fluid mechanics. The efficiency can still be improved in the case of important calculations (e.g. shape optimization), using neural networks to analyze the problem, instead of using a conventional finite element analysis. Actually, the use of neural networks to model mechanical structures appears to give good results [10]. It would be possible to make the learning of a neural network in parallel with the first generations that would be calculated by finite elements (that is using the results of the finite element analyses). Once the learning stage is over, the neural network would completely replace the calculations by finite elements. The calculations would therefore become much faster, and the genetic algorithm method, contrary to the deterministic methods, does not need extremely precise calculations of the objective function.

Translated by Amandine MARCELIN, on behalf of AMTrad'gram (www.amtradgram.com).

References

- [1] J.L. Marcelin, "Genetic optimization of stiffened plates and shells", *Int. J. for Numerical Methods in Engineering*, Vol. 51, n°9, p.1079-1088, 2001.
- [2] J.L. Marcelin, "Genetic optimization of gears", *International Journal of Advanced Manufacturing Technology*, vol.17, n°12, p.910-915, 2001.

[3] J.L. Marcelin, "Genetic optimization of stiffened plates without the FE mesh support", *Int. J. for Numerical Methods in Engineering*, Vol.54, n°5, p. 685-694, 2002.

[4] J.L. Marcelin, "Using Genetic algorithms for the optimization of mechanisms", *International Journal of Advanced Manufacturing Technology*, vol. 27, n°1-2, p. 2-6, 2005.

[5] J.L. Marcelin, "Rayleigh-Ritz function approximations in computationally intensive genetic design", *International Review of Mechanical Engineering*, Vol. 1, n°2, p. 166 -173, 2007.

[6] J.L. Marcelin, "Optimization of vibration frequencies of rotors via Rayleigh-Ritz method and genetic algorithms", *International Review of Mechanical Engineering*, Vol. 2, n°1, p. 144 -148, 2008.

[7] T.Su and H. Ling, "Determining the equivalent impedance boundary condition for corrugated coatings based on the genetic algorithm", *IEEE Transactions on Antennas and Propagation*, Vol.48, no.3, pp. 374-382, 2000.

[8] J.H. Son and B.M. Kwak, "Optimization of boundary conditions for maximum fundamental frequency of vibrating structures", *AIAA Journal*, Vol. 12, n°31, p. 2351 -2357, 1993.

[9] B.P. Wang and J.L. Chen, "Application of genetic algorithm for the support location optimization of beams", *Computers and Structures*, Vol.58, no.4, pp. 797-800, 1996.

[10] J.L. Marcelin, "Cognitive optimization of mechanical structures", *International Review of Mechanical Engineering*, Vol. 5, n°1, p. 88 -91, 2011.

Author information



J.L. Marcelin Laboratoire G-SCOP (Grenoble, Sciences pour la Conception, Optimisation et Production) Grenoble-INP/UJF-Grenoble1/CNRS, UMR5272 38031 GRENOBLE France
e-mail : Jean-Luc.Marcelin@ujf-grenoble.fr

Jean-Luc Marcelin is born 31 march 1956. J.L. Marcelin is a doctor engineer of the "Ecole des Mines" of Paris (1983), and entitled to direct search to the University Joseph Fourier of Grenoble (1997). He is currently an associate professor at the University Joseph Fourier. He carries out search in optimization of structures and mechanical systems. He is an author or a co-author of 74 scientific publications, including 49 international publications, and 5 books of general interest.