Genetic algorithms and finite element coupling for mechanical optimization

G. Corriveau, R. Guilbault & A. Tahan
Department of Mechanical Engineering,
Ecole de technologie superieure, Canada

Abstract

Optimization of mechanical components is an important aspect of the engineering process; a well-designed system will lead to money saving during the production phase and better machine life. On the other hand, optimization actions will increase the engineering investment. Consequently, and since computer time is inexpensive, an efficient design strategy will tend to transfer the effort from the staff to the computers. This paper presents an efficient design tool made to carry out this task: a new optimization model based on genetic algorithms is developed to work with commercial finite element software. The objective is to automate optimization of static criteria (stresses, weight, strength, etc.) with finite element models. In the proposed model, the process acts on two geometric aspects of the shape to be optimized: it controls the position of the vertices defining the edges of the volume and, in order to minimize stresses concentrations, it can add and define fillet between surfaces. The model is validated from some benchmark tests. An industrial application is presented: the genetic algorithms-finite element model is employed to design the fillets at the crown-blade junctions of a hydroelectric turbine. The results show that the model converges to a very efficient solution without any engineer intervention.
Keywords: genetic algorithms, shape optimization, finite element, hydroelectric turbine.

1 Introduction

The design process of any mechanical part controls its global cost. A well-designed system will lead to money saving during the production phase and better machine life. Incorporate an optimization cycle into the design process is

WIT Transactions on The Built Environment, Vol 91, © 2007 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/OP070091

then primordial. On the other hand, the optimization of mechanical components could increase the delays and cost related to the design. As much as 70 to 80% of the final production cost can result from the design process [1]. Consequently, and since computer time is inexpensive, an efficient design strategy will tend to transfer the effort from the staff to the computers. Powerful calculation approaches, such as finite element method (FEM) and numerical optimization schemes are then required.

This paper includes a brief description of the genetic algorithms in section 2. Section 3 is devoted to the coupling method while section 4 discusses the application case.

2 Genetic algorithms

Genetic algorithms (GA) can be considered as a controlled random walk, they efficiently exploit information from previous configurations to generate new configurations with improved performances expected [2]. GA are formed principally with three operators; selection, crossover and mutation. Numerous operator types are described in the literature depending on the problem to be solved and the coding used to represent the configurations. Imagination is the only limit to the development of new operators. Michalewicz [3] gives a detailed description of the different selection, crossover and mutation types.

2.1 Description of the genetic algorithms process

Genetic algorithms use a population of configurations, called individual, to evolve over a number of generations. Each individual is represented by its genetic material, called chromosome. For optimization purpose, the chromosome is described by the design variables. Different kinds of coding are possible. However, this paper will deal with binary coding (Figure 1).

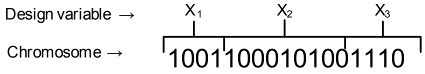

Figure 1: Individual chromosome representation.

The process starts with an initial population of *n* individuals. The first individual has the default configuration; while the others are randomly generated. The performance of each individual is then evaluated in regard to the objective function and the handling of constraints (if some are considered). Using the performance of these individuals, a selection is done in the population to identify valuable parents. Higher is the performance of an individual, higher is its probability to become parent. Two parents are match randomly to exchange their genetic materials to form the offspring for the next generation. This exchange process is called crossover. The crossover process is associated to a probability (p_c). If this process doesn't happen, the parents are directly transferred to the next generation meaning the cloning of these individuals. After the crossover operator and before forming the next generation, all the individuals are forced to undergo a mutation process. A probability (p_m) dictates if the mutation occurs. The evolution procedure is repeated until the population converges to a certain level or simply if the maximum number of generation is reached.

The efficiency of the genetic algorithms has been proved experimentally for a wide range of scientific field [4,5]. Genetic algorithms have a theoretical background mostly developed by Holland [6] and well described by Goldberg [2].

2.2 Genetic operators

Despite de fact that no general conclusion can be brought, some guidelines have been proposed to determine the best type of operator and genetic algorithms parameters like p_c, p_m and the size of the population. These guidelines are mostly based on empirical experiments, where different kinds of problem can lead to different conclusions. Srinivas and Patnaik [7] expose some empirical evidences regarding the choice of the operators and the parameters. Section 2.4 discusses in more details the selection of genetic operators and parameters.

2.2.1 Selection

Different types of selections are implanted in the optimization model described in section 3, but only the method called tournament selection is used for the application presented in section 4. The tournament selection randomly identifies some competitors from the population to compete against each other. The one with the highest performance win a parent status.

The tournament selection permits to control the selective pressure put on the population. The population diversity is adjusted by modification of the competitor number. Greater competitor numbers in the tournament increase the chances to focus the search over the best individuals - meaning a greater selective pressure. On the other hand, with only two competitors, the possibility of becoming a parent remains open to a larger band of the population - meaning a lesser selective pressure.

2.2.2 Crossover

Four types of crossover are used for the application presented in section 4; 1-point, 2-point, uniform and weighted crossover. The 1-point crossover randomly determines a cross-point in the length of the chromosome, combines the left part of the chromosome of the first parent with the right side of the chromosome of the second parent to form the first offspring. A second offspring is inversely generated.

The 2-point crossover implies two cross-points. The first offspring has the beginning and the last parts of the chromosome of the first parent and the middle portion of the second one. The second offspring is again inversely generated.

The procedure is quite different for the uniform crossover. With this type of crossover, a random 0 or 1 is selected for each bit of the chromosome and for the bit where a 0 is chosen; the first offspring uses the bit of the first parent at this position. On the other hand, when a 1 is selected, the first offspring takes the bit of the second parent at this position. For the second offspring, the random 0 and 1 are inversely used. The name uniform comes from the fact that the random 0 or 1 has the same probability to be selected (50%).

The weighted crossover is similar to the uniform crossover. However, the probability of selecting a random 0 or 1 is not fixed at 50%. Also, it is important to sort the two parents to make sure that the first parents correspond to the best ones, in regard to their fitness. Than, the probability allocated to select a random 0 is fixed between 50% and 100% with these limits excluded. By using this method the first offspring will have a greater contribution from the best parent. Again, the second offspring is inversely generated.

2.2.3 Mutation

Mutation acts as an insurance policy against premature loss of important notions when it is used with selection and crossover operators [2]. With binary coding, the mutation proceeds by changing a bit indicating 0 by 1 or vice versa. The mutation operation progresses over each bit of the chromosome with a probability p_m of being applied. The p_m probability is normally very small (<1%). Over a certain level, the mutation could turn the genetic algorithm into a simple random walk, meaning a lost in the efficiency related to the search strategy.

2.3 Convergence of genetic algorithms

The definition of some convergence criteria allows the genetic algorithms to stop the search process without attainment of the global optimum. On the other hand, the evolution (not the process), ends when the best configuration reaches the global optimum of a given environment. Different kinds of convergence criteria could define an acceptable solution. The criteria could be based on the best individual or on the average of the population. A maximum number of generations or a maximum allowable time for the evolution process could also be specified.

2.4 Parameters in genetic algorithms

As indicated in section 2.2, there is no strait way to determine which type of operators or what are the best parameters. Nevertheless, Eiben et al. [8] give a good review of this topic and propose a classification. This classification is used here to illustrate the operators and parameters setting. The discussion is presented for the parameters, but the same can be applied to operators.

Two categories divide the way the parameters can be set. The first is a static setting and the second is a dynamic setting. The static setting, also called parameters tuning, is the simplest way to define parameters, but does not lead to the optimal evolution. Parameters tuning relies on tests made before starting the experiments in order to find the best combination of parameters (e.g. p_c, p_m and the population size). The parameters remain constant over the generations. Parameters tuning could be done with trial-and-error method, with design of experiment (DOE), by using other heuristic algorithms or simply with experience on similar problems. Due to the fact that these settings cannot change the balance between the exploration and the exploitation of the search domain during the evolution process, it can be said that they do not correspond to optimal settings.

During the searching, it could be appropriate to modify the exploration/exploitation balance. Normally, at the beginning it is important to explore the domain, whereas at the end it is preferable to exploit the best domain's region to reach the optimum. This can be achieved with a dynamic setting of the parameters, also called parameters control. The control could be deterministic or stochastic.

In this paper, the operators and parameters are set with a static setting. This choice has been made considering that no dynamic setting taking care of all the interactions between operators and parameters is already available. On the other hand, the coupling proposed between genetic algorithms and finite element has to be validated on complex components (section 4) before developing a dynamic setting approach. Table 1 gives the static parameters used. The parameter values selection was based on the recommendations of De Jong [9] and Grefenstette [10] combined to tests on simple forms (beam in flexion, stress concentration in notches and cylinder in tension).

Table 1: Genetic parameters used.

Parameter	Value
Population size	50
p_c	0.8
p_m	0.005
Tournament competitors	2

2.5 Advantages and inconvenient of genetic algorithms

Genetic algorithms have numerous advantages over traditional optimization methods (Haupt and Haupt [11]). The most important are that they can optimize continuous and discrete variables, can treat analytical function, experimental or numerical simulation data, while dealing with large amount of variables. Moreover, the derivative information is not required. Genetic algorithms search from a wide sampling of the cost surface which can be very complex. Finally, genetic algorithms provide a list of potential solutions.

Nevertheless, genetic algorithms could be time consuming depending on the evaluation required to characterize the performance of each individual. For shape optimization, such as the case of section 4, where the finite element method is used to define the performance of a configuration, this could be an issue. Section 3 discusses the method used to minimize the impact of the finite element analyse. It is also important to mention that genetic algorithms do not guarantee that the global optimum will be found. However, with a good genetic setting, this method will provide improvement compare to the initial configuration.

3 Design optimization model

The optimization design tool created with Tcl/Tk and named AGC, allows to couple the genetic algorithm approach with commercial FEM software (ANSYS). AGC controls all the settings and the processing related to the genetic algorithms, all the information needed to optimize the mechanical component and all the parameters required to produce the evaluation by the FEM.

The vertices defining a line or a B-spline of the geometry can be used as design variables to modify the shape of the component. Also, the addition and the optimization of fillet radius between surfaces can be set as design variables.

Finally, AGC creates a list of the chromosome evaluated by the finite element software during an experiment. This list ensures not to reanalyse twice the same individual in successive generations. Because the FEM analysis part corresponds approximately to 95% of the total time elapsed during the processing of an individual, this list leads to great time saving. Figure 2 shows, for the experiment presented in the next section, the relative numbers of finite element configurations evaluated in each generation (50 individuals). It is interesting to note that less than 5 individuals (10% of the population) are evaluated after generation 20. For one individual, it took 4 minutes (in average) for the evaluation step - geometry construction, meshing, solving operations and reading of the objective result. This gave a total of 3.33 hours for the process of the generation 0 (100% of the population analysed). Consequently, it would take 166.5 hours for a complete evolution process. However, because of the already analysed chromosomes list, the total evolution process took 52.5 hours, which corresponds to a saving of 114 hours. The evolution process was run on an Intel® server with 2 Xeon® Dual Core 1.6GHz and 2.0 Go of RAM.

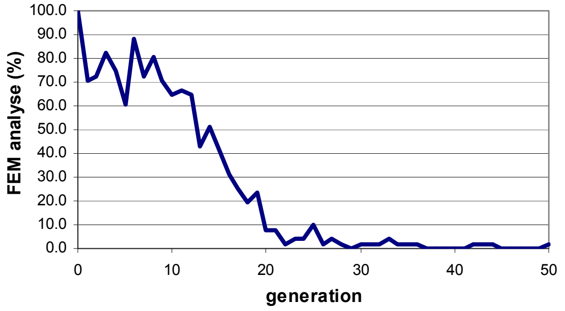

Figure 2: Percentage of finite element analyse.

4 Application

This section presents the optimization of the fillet between the crown-blade junctions of a hydroelectric turbine. Figure 3 details the turbine components. The objective is to reduce the maximum static load stress (Von Mises). As shown by

Sabourin et al. [12], the critical zone of such a structure is located at the crown-blade junction in the trailing edge vicinity (Figure 4).

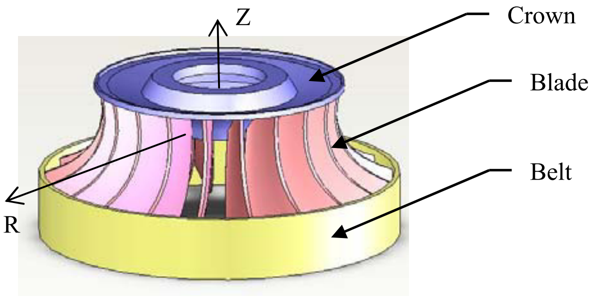

Figure 3: Hydroelectric turbine's detail.

The treated turbine has fictive dimensions. Nevertheless, the pressure defined on the blade was tuned to realistic power output (52.3 MW). The turbine is composed of 20 blades. Because of the symmetry, only one blade is analysed.

The crown-blade junction is defined by four design variables described in table 2 and Figure 4. The design variables are independent - variable fillet radii. The transition between two different fillet radii is also controlled to be smooth.

Because genetic algorithms are a stochastic process, five experiments were conducted. The evolution was limited to a maximum of 50 generations with a convergence criterion stopping the process when no more than 1% of average improvement was observed over ten generations. The other genetic parameters used are described in table 1. The results of these experiments are presented in table 3 below. Because the goal is to minimize the maximum stress, only the results from the best experiment (e.g. experiment 3) will be further discussed.

Table 2: Design variables.

Design variable (fillet radius)	Description	Lower bound (mm)	Upper bound (mm)	Precision (decimal)	Default value (mm)
X1	Suction side leading edge	12	63	0	38
X2	Pressure side leading edge	12	63	0	38
X3	Suction side trailing edge	12	63	0	38
X4	Pressure side trailing edge	12	63	0	38

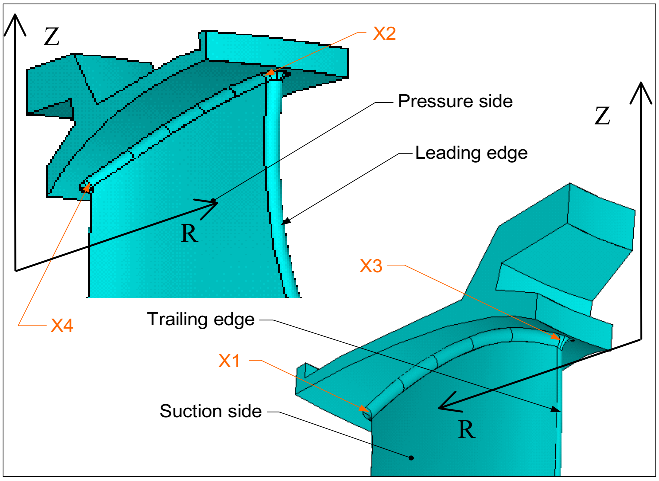

Figure 4: Description of the design variables.

Table 3: Experiment results.

Experiment (time-hour)	Crossover	Best value (mm)				Stop gen.	Best ind. (gen-ind)	Stress max (MPa)	Impro-vement
		X1	X2	X3	X4				
Original	-	38	38	38	38	-	-	295.34	-
1 (31.2)	1-point	62	60	44	53	40	18-27	280.65	4.97%
2 (31.6)	1-point	59	58	43	59	42	42-37	282.87	4.22%
3 (52.5)	Uniform	63	56	46	55	50	17-3	276.11	6.51%
4 (35.3)	Weighted 70%	56	62	42	50	25	10-46	283.87	3.88%
5 (26.9)	2-point	61	55	38	48	21	9-45	278.70	5.63%

Experiment 3 is the only one that did not stop with the convergence criteria. However, it's the experiment with the highest improvement (6.51%). Figure 5 presents the evolution process for experiment 3. The stress curve of the best individual and the stress curve of the population average are shown. The chart of Figure 5 shows that, even if the average population did not converge, the best individual converged at generation 17. An elite strategy was also incorporated into the genetic process to ensure that if no improvement appeared over the evolution, the original configuration live through all the process. In other words, the best individual can never get worst over successive generations. Figure 6 shows the differences between the original configuration and the best configuration found in these experiments. The stress distribution remains similar but the stress maximum value decreases from 293.34 MPa to 276.11 MPa.

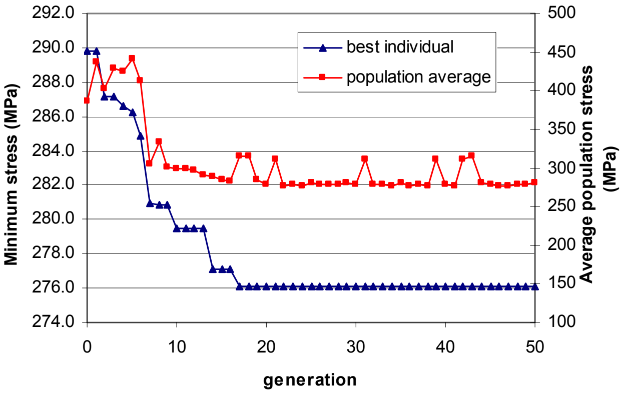

Figure 5: Evolution process of experiment 3.

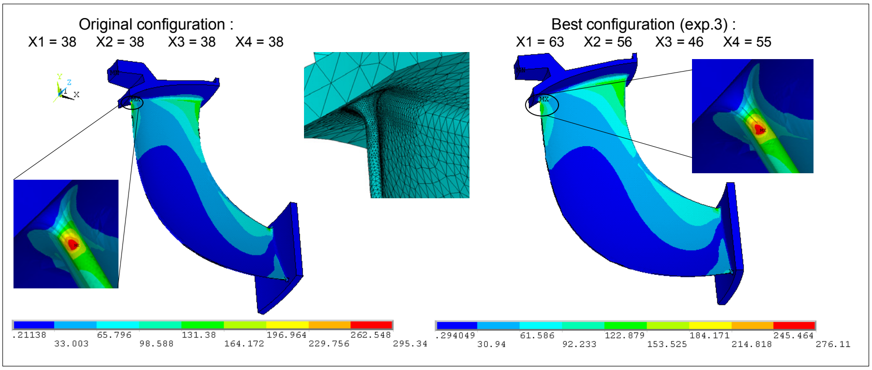

Figure 6: Von Mises stress.

5 Conclusion

This paper described a new model coupling genetic algorithms and finite element to optimize mechanical components. The goal of this optimization is to improve the shape of any structure to reduce the static stress in critical regions. The model was applied to the optimization of the crown-blade junction (fillet radius) of a hydroelectric turbine. In this application, the stress was decreased by a factor of 6.5%. The stress reduction could have reached a higher level with the inclusion of a few more design variables such as the thickness of the crown. However, the objective here was to illustrate the efficiency of the model even with very limited modifications of the initial form. In fact, such a fine-tuning of the fillet junction would be difficult to realize without a powerful numerical optimization method.

Nonetheless, it is clear that more work is required to reduce the time of the whole process. In the improvement to come, the focus will be put on the replacement, during the evaluation process, of the finite element simulation by a neural network, trained with the results generated during the first generations.

References

[1] Ertas, A. & Jones, J.C., *The engineering design process*, John Wiley & Sons: New York, 1996.

[2] Goldberg, D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley: Massachusetts, 1989.

[3] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag: Berlin and New York, 1994.

[4] Davis, L. (Ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold: New York, 1991.

[5] Renner, G. & Ekart, A., Genetic algorithms in computer aided design. *Computer-aided design*, 35, pp.709-726, 2003.

[6] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press: Michigan, 1975.

[7] Srinivas, M. & Patnaik, L.M., Genetic Algorithms: A Survey. *Computer*, 27(6), pp. 17-26, 1994.

[8] Eiben, A.E., Hinterding, R. & Michalewicz, Z., Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2), pp.124-141, 1999.

[9] De Jong, K., *The analysis of the behaviour of a class of genetic adaptive systems*. Ph.D. dissertation, University of Michigan, 1975.

[10] Grefenstette, J.J., Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1), pp. 122-128, 1986.

[11] Haupt, R.L. & Haupt, S.E., *Practical Genetic Algorithms*, Wiley-Interscience Publication: New York, 1998.

[12] Sabourin, M. et al., Mechanical Loads and Fatigue Analysis of a Francis runner. *HydroVision*, pp.1-13, 2004.