Advances in Engineering Software 41 (2010) 422–426

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Genetic algorithms and finite element coupling for mechanical optimization

Guillaume Corriveau, Raynald Guilbault *, Antoine Tahan

Department of Mechanical Engineering, Ecole de Technologie Superieure, 1100 Rue Notre-Dame Ouest, Montreal, Canada H3C 1K3

ARTICLE INFO

Article history:
Available online 9 April 2009

Keywords:
Genetic algorithms
Shape optimization
Finite element
Hydroelectric turbine

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.

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 then primordial. On the other hand, the optimization of mechanical components could increase the delays and cost related to the design. As much as 70–80% 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 (Fig. 1).

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 does not 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

* Corresponding author. Tel.: +1 514 396 8862; fax: +1 514 396 8530.
E-mail address: raynald.guilbault@etsmtl.ca (R. Guilbault).

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2009.03.008

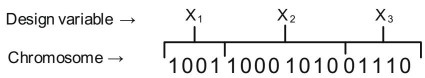

Fig. 1. Individual chromosome representation.

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