Application of Genetic Programming to the Choice of a Structure of Global Approximations

Vassili V. Toropov
Dept. of Civil & Environmental Engineering
University of Bradford
Bradford BD7 1DP, West Yorkshire, UK
V.V.Toropov@bradford.ac.uk
http://www.brad.ac.uk/staff/vtoropov

Luis F. Alvarez
Dept. of Civil & Environmental Engineering
University of Bradford
Bradford BD7 1DP, West Yorkshire, UK
L.F.Alvarez@bradford.ac.uk
http://www.student.brad.ac.uk/lfalvare

ABSTRACT

Genetic Programming methodology is applied to the creation of the structure of global approximation functions used in the solution of design optimization problems. The approximations are obtained by the least-squares surface fitting (the response surface methodology). Attention is paid to the appropriate choice of the plan of experiments and to the model tuning using the nonlinear least-squares method. A test example is presented where the technique is applied to a simple truss optimization problem.

1. Introduction

Nowadays methods based on approximation concepts take dominant position in the treatment of complex design optimization problems, and the development of new high quality approximation functions is considered a high priority problem (Barthelemy and Haftka, 1993) (Sobieszanski and Haftka, 1997). The choice of the structure of approximation functions is the subject of this study.

Depending on the range of their applicability, the approximations can be classified as local (valid in a vicinity of a design point), global (valid in the whole region defined by side constraints) and mid-range (also called multipoint approximations (Toropov, 1989) (Toropov, Filatov and Polynkin, 1993) (van Keulen and Toropov, 1997)) which is the combination of the two basic approaches. Global and mid-range approximations are normally based on the response surface methodology (Roux, Stander and Haftka, 1996) (Schoofs, 1987), which is a method of constructing approximations of the system behaviour using results of the response analysis carried out at a series of points in the design variable space. The approximation functions are obtained by the least-squares method. One of the major problems in the application of such techniques is the necessity to select a structure of the approximation function.

This study attempts to develop and use a Genetic Programming (GP) methodology for the creation of an approximation function structure of the best possible quality, and use it within a global or multipoint approximation technique.

2. Response Surface Methodology (Global Approximations)

According to this technique, the original optimization problem

$$F_0(x) \rightarrow min, \quad F_j(x) \le 1 \quad (j = 1,\dots,M),$$
$$A_i \le x_i \le B_i \quad (i = 1,\dots,N) \tag{1}$$

is replaced by a much simpler mathematical programming problem:

$$\tilde{F}_0(x) \rightarrow min, \quad \tilde{F}_j(x) \le 1 \quad (j = 1,\dots,M),$$
$$A_i \le x_i \le B_i \quad (i = 1,\dots,N) \tag{2}$$

The functions $\tilde{F}_j(x)$ $(j = 1,\dots,M)$ present global approximations of the corresponding original functions $F_j(x)$, and the solution of the problem (2) is considered as a current approximation of the solution of the problem (1). The same basic procedure can be repeated in a relatively small neighbourhood of the obtained point in the design variable space in order to improve the solution.

It is proposed to choose the structure of the functions $\tilde{F}_j(x)$ using the Genetic Programming methodology.

3. Genetic Programming

Genetic Programming (Koza, 1992) is a branch of Genetic Algorithms (GA) (Goldberg, 1989). Their basis is the same Darwinian concept of survival of the fittest. The innovation of GPs is the use of more complex structures. While GAs use a string of numbers to represent a solution, the GP creates computer programs.

In our case of design optimization, the programs represent an empirical model to be used for approximation of response functions in the original optimization problem. The programs are composed of elements from a *terminal set* and a *functional set*. The terminal set (variable nodes) consists of N design variables $x_1, x_2, \ldots, x_N$. The functional set contains the mathematical operators that will be used to generate the regression model, e.g. {+, *, /, power, square, square root, negation, ...}.

3.1. Fitness Function

When selecting randomly a tree to perform any genetic operation, the so-called *fitness proportionate* method is used here. This method specifies the probability of selection on the basis of the fitness of the solution. The fitness of a solution shall reflect (*i*) the quality of approximation of the experimental data by a current expression represented by a tree and (*ii*) the length of the tree in order to obtain more compact expressions. In problems of empirical model building, the most obvious choice for the estimation of the quality of the model is the sum of squares of the difference between the simplified model output (2) and the results of runs of the original model (1) over some chosen plan (design) of experiments. In a dimensionless form this measure of quality of the solution can be presented as follows:

$$Q = \frac{\sum_{p=1}^{P} \left(F_p - \tilde{F}_p\right)^2}{\sum_{p=1}^{P} \tilde{F}_p^{\,2}} \qquad (3)$$

where, for a given tree, $\tilde{F}_p = \tilde{F}(\mathbf{x}_p)$ is the simplified function value corresponding to the p-th point of the plan of experiments and $F_p = F(\mathbf{x}_p)$ is the original function value at the same point.

If $Q(S_i)$ is the measure of quality of the solution S_i, Q_u is an upper limit value of the quality for all N_t members of the population, ntp_{max} is the maximum allowed number of tuning parameters, ntp_i is the number of tuning parameters contained in the solution S_i and c is a coefficient penalizing the excessive length of the expression, the fitness function $\Phi(S_i)$ can be expressed in the following form:

$$\Phi(S_i) = Q_u - Q(S_i) + c * \left(ntp_{max}^2 - ntp_i^2\right) \qquad (4)$$

The probability that the solution S_i will be selected is

$$\frac{\Phi(S_i)}{\sum_{j=1}^{N_t} \Phi(S_j)}$$

Programs with greater fitness values $\Phi(S_i)$ have a greater chance of being selected in a subsequent genetic action. Highly fit programs live and reproduce, and less fit programs die.

3.2. Genetic Operators

Model structures evolve through the action of three basic genetic operators: reproduction, crossover and mutation. Figure 1 shows a flowchart of the process.

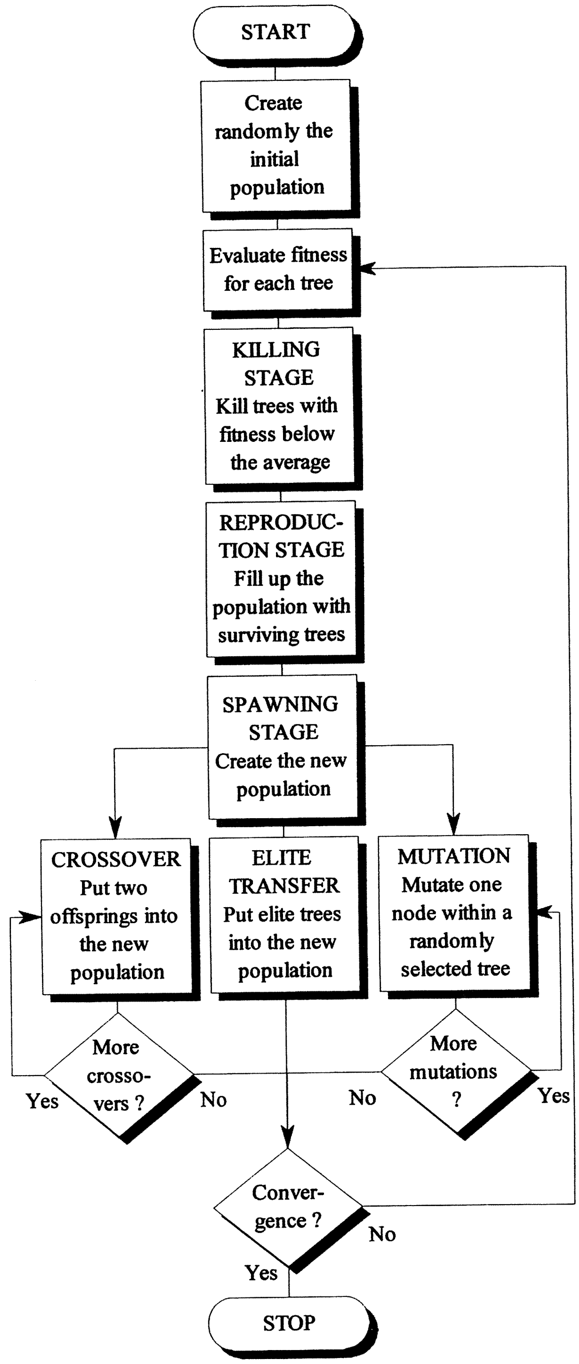


Figure 1. Flowchart of GP methodology

Crossover and *mutation* provide diversity of the population. In the *reproduction* stage, a strategy must be adopted as to which programs should die. In this simple implementation, trees with fitness below the average are killed. The population is then filled with the surviving trees according to fitness proportionate selection. Finally, a relatively small number of the fittest programs, called the elite, is selected to be transferred unchanged to a next generation. As a result, a new population of trees of the same size as the original one is created, but it has a higher average fitness value.

4. Implementation

The algorithm has been implemented in C++ following recommendations given in (Kinnear, 1994) and a sample program presented in (Kuhlmann and Hollick, 1995). In the process of finding the structure of approximations by the genetic search as described above, it is necessary to address two important problems: the choice of the plan (design) of experiments and the model tuning (evaluation of tuning parameters) prior to the fitness evaluation.

4.1. Design of Experiments

The selection of points in the design variable space where the response function is to be evaluated is commonly called design of experiments. The choice of the design of experiments can have a large influence on the accuracy of the approximation and the cost of constructing the response surface.

In this paper, the approach suggested by (Audze and Eglais, 1977) (Rikards, 1993) is used. It considers a non-traditional criterion for elaboration of plans of experiments which is not dependent on the mathematical model of the object or process under consideration.

The plan of experiments is characterized by a matrix, which contains the levels of factors N (number of design variables) for each of K experiments. For example, for $N = 2$ and $K = 10$, the matrix is

$$\begin{vmatrix} 8 & 10 & 4 & 6 & 2 & 3 & 9 & 5 & 7 & 1 \\ 1 & 7 & 10 & 6 & 8 & 5 & 4 & 2 & 9 & 3 \end{vmatrix} \qquad (5)$$

The corresponding plan is shown in Figure 2.

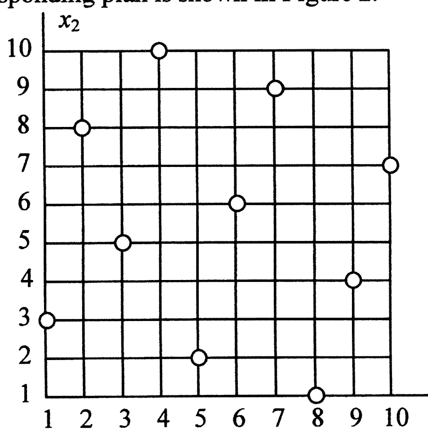


Figure 2. Plan of experiments for $N = 2$ and $K = 10$

4.2. Model Tuning

The simplified model is characterized not only by its structure (to be found by the GP) but also by a set of tuning parameters $\boldsymbol{a}$ to be found by the model tuning, i.e. the least-squares fitting of the model into the set of values of the original response function:

$$G(\boldsymbol{a}) = \sum_{p=1}^{P} \left(F_p - \tilde{F}_p(\boldsymbol{a})\right)^2 \quad \rightarrow \quad min \qquad (6)$$

The allocation of tuning parameters $\boldsymbol{a}$ to an individual tree follows basic algebraic rules. To identify the parameters of the expression by the nonlinear least-squares fitting, i.e. to solve the optimization problem in (1), a combination of a genetic algorithm (GA) and a nonlinear mathematical programming method (Madsen and Hegelund, 1991) is used. The output of the GA is the initial guess for the subsequent derivative-based optimization method which amounts to a variation of the Newton's method in which the Hessian matrix is approximated by the secant (quasi-Newton) updating method. Once the technique comes sufficiently close to a local solution, it normally converges quite rapidly. To promote convergence from poor starting guesses the algorithm uses the adaptive update of the Hessian and, consequently, the algorithm is reduced to either a Gauss-Newton or Levenberg-Marquardt method.

5. Example

In order to test the genetic programming algorithm, a three-bar truss optimization problem described in (Haftka and Gurdal, 1993) was used. The two design variables x_1 and x_2 describe the cross-sectional areas of individual bars (Figure 3), the objective function is the volume of the material and the constraints limit the stresses in all bars and the displacement of the free node. The set of response data was generated using the plan of experiments (5), which was then used to build the global approximations of the objective function and the constraints.

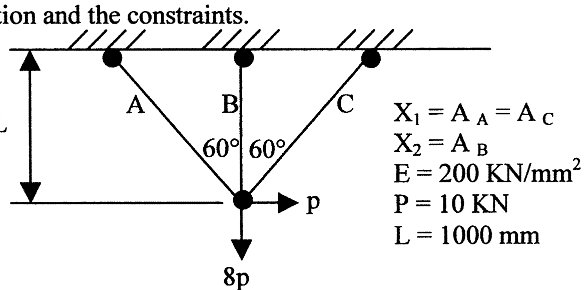


Figure 3. Three-bar truss optimization problem

The following parameters have been used:

- population size: N_t=100
- proportion of the elite: P_e=0.2
- probability of mutation: P_m=0.001
- functional set: binary functions +, *, /, ^ ; unary functions $(...)^2$, $\sqrt{(...)}$, -(...)
- terminal set: design variables x_1, x_2.

The output of the algorithm still needed some manual post-processing in order to get rid of those terms in the expression that give a null or tiny contribution, for example when the same value is added and subtracted. It is then suggested to run the problem several times in order to identify, by comparison, the most likely components.

The optimization problem is reduced to the approximated one shown in table 1.

Table 1. Results of the three-bar truss approximation.

ORIGINAL FUNCTIONS	APPROXIMATIONS
$F_0 = 1000 * (4 * x_1 + x_2) \rightarrow min$	$\tilde{F}_0 = 4000 * x_1 + 1000 * x_2 \rightarrow min$
Subject to:	
ORIGINAL FUNCTIONS	**APPROXIMATIONS**
$F_1 = \dfrac{400}{0.25 * x_1 + x_2} \leq 1$	$\tilde{F}_1 = \dfrac{106373.40}{66.48 * x_1 + 265.94 * x_2} \leq 1$
$F_2 = 25 * \left(\dfrac{(0.25 * \sqrt{3} + 6) * x_1 + \sqrt{3} * x_2}{3 * x_1 * x_2 + 0.75 * x_1^2} \right) \leq 1$	$\tilde{F}_2 = \dfrac{-17292.79 * x_1 - 4655.76 * x_2}{-(322.56 * x_2 + 80.65 * x_1) * x_1} \leq 1$
$F_3 = \dfrac{200}{0.25 * x_1 + x_2} \leq 1$	$\tilde{F}_3 = \dfrac{-41886.67}{-52.35 * x_1 - 209.41 * x_2} \leq 1$
$F_4 = 25 * \left(\dfrac{(-0.25 * \sqrt{3} + 6) * x_1 - \sqrt{3} * x_2}{3 * x_1 * x_2 + 0.75 * x_1^2} \right) \leq 1$	$\tilde{F}_4 = \dfrac{8170.86 * x_1 - 2542.18 * x_2}{(176.12 * x_2 + 44.03 * x_1) * x_1} \leq 1$

6. Conclusion

It is proposed to use the genetic programming methodology for the creation of the structure of global approximation functions when the response surface technique is adopted. It has been successfully applied to build high quality approximations for test problems, showing considerable potential for complex structural optimization and identification problems.

Acknowledgements

This research is supported by the Training and Mobility of Researchers (TMR) Program of the European Commission, contract ERBFMBICT961615. The first author would like to express his appreciation of support provided by Delft University of Technology, The Netherlands.

The authors thank Dr. Fred van Keulen of Delft University of Technology for his useful discussions.

Bibliography

Audze, P. and Eglais, V. 1977. New approach for planing out of experiments. *Problems of Dynamics and Strengths.* Riga, Zinatne Publishing House, 35, 104-107 (in Russian).

Barthelemy, J.-F.M. and Haftka, R.T. 1993. Approximation concepts for optimum structural design - a review. *Structural Optimization,* 5, 129-144.

Goldberg, D.E. 1989. *Genetic algorithms in search, optimisation and machine learning.* Reading, MA. Addison-Wesley.

Haftka, RT. and Gurdal, Z. 1993. *Elements of Structural Optimization.* 3rd ed., Kluwer Academic Publishers. Page 377.

Kinnear, K.E. 1994. *Advances in genetic programming.* MIT Press.

Koza, J.R. 1992. *Genetic Programming: On the programming of computers by means of natural selection.* Cambridge, MA. MIT Press.

Kuhlmann, H. and Hollick, M. *Genetic programming in C/C++.* Final Report CSE99/CIS899. May, 1995. URL: http://www.cis.upenn.edu/~hollick/genetic/paper2.html.

Madsen, K. and Hegelund, P. *Non-gradient subroutines for non-linear optimization.* Institute for Numerical Analysis, Technical University of Denmark, Report NI-91-05. June, 1991.

Rikards, R. 1993. Elaboration of optimal design models for objects from data of experiments. In: Optimal design with advanced materials, The Frithiof Niordson volume. *Proceedings of the IUTAM Symposium,* Lyngby, Denmark. Pedersen, P. (ed.). Elsevier. Pages 113-130.

Roux, W.J.; Stander, N. and Haftka, R.T. 1996. Response surface approximations for structural optimization. AIAA paper 96-4042-CP. *Proc. 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.* Bellevue WA. Part 2, pages 565-578.

Schoofs, A.J.G. *Experimental design and structural optimization.* Ph.D. Thesis. Eindhoven University of Technology, The Netherlands, 1987.

Sobieszanski-Sobieski, J. and Haftka, RT. 1997. Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization,* 14, 1-23.

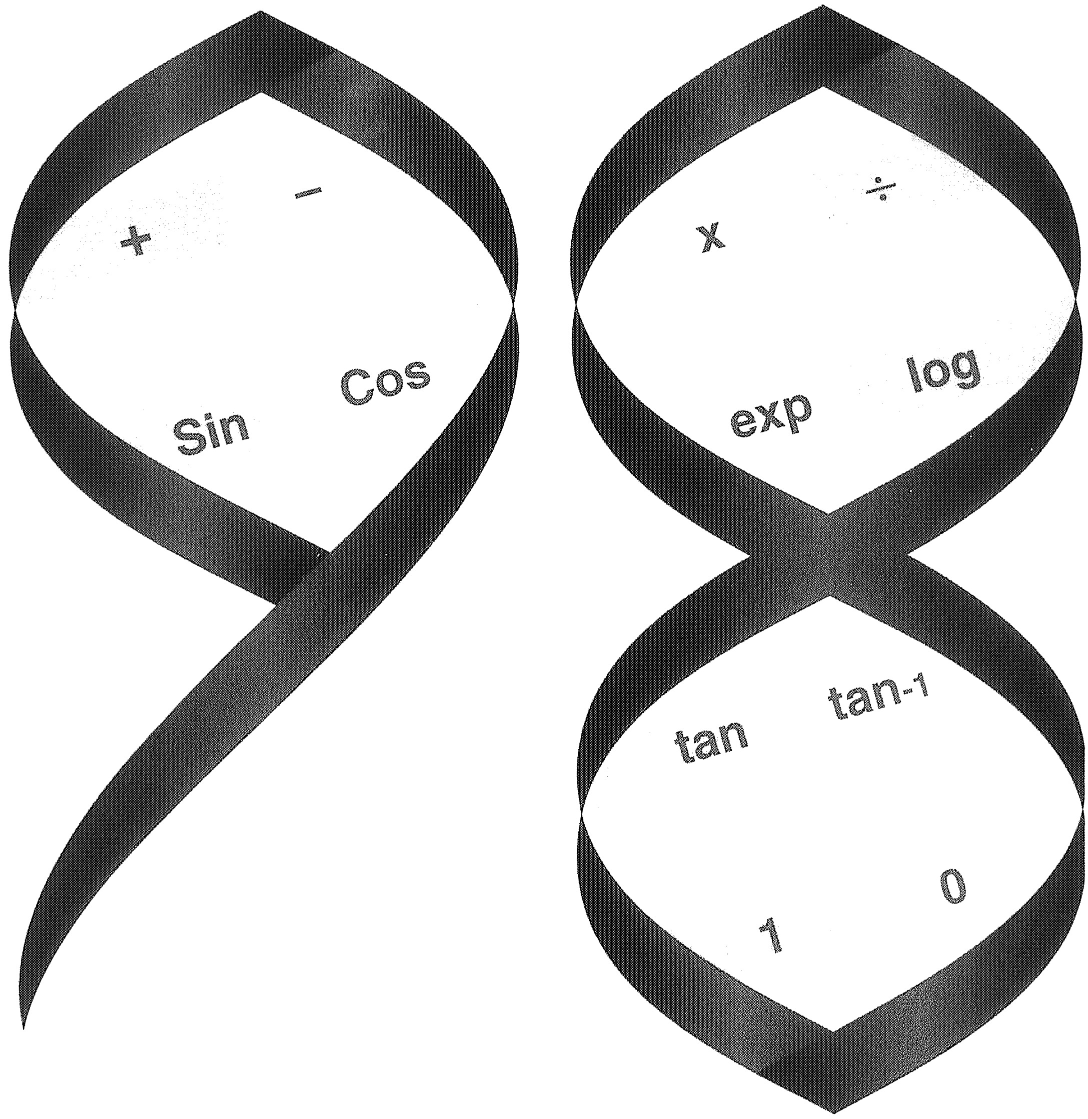
Toropov, V.V. 1989. Simulation approach to structural optimization. *Structural Optimization,* 1, 37-46.

Toropov, V.V.; Filatov, A.A.; Polynkin, A.A. 1993. Multiparameter structural optimization using FEM and multipoint explicit approximations. *Structural Optimization,* 6, 7-14.

van Keulen, F. and Toropov, V.V. 1997. New developments in structural optimization using adaptive mesh refinement and multi-point approximations, *Engineering Optimization,* 29, 217-234.

CONFERENCE PROCEEDINGS

Genetic Programming



edited by

John R. Koza
Wolfgang Banzhaf
Kumar Chellapilla
Kalyanmoy Deb
Marco Dorigo
David B. Fogel
Max H. Garzon
David E. Goldberg
Hitoshi Iba
Rick L. Riolo

Proceedings of the Third Annual Genetic Programming Conference

July 22–25, 1998
University of Wisconsin, Madison, Wisconsin