CHAPTER 1 *Introduction*

1.1 Design optimization

The objective of design optimization is to find the set of parameters that minimize an objective function subject to a set of behavioural constraints. The importance of design optimization was first recognised by the aerospace industry where aircraft structural designs are controlled by weight. In other industries like civil or mechanical engineering, cost may be the primary consideration although weight has a large influence in the performance.

The group of parameters that can be varied to improve the design are called *design variables* and can describe cross-sectional dimensions, material properties, shape of structure, etc. They are denoted by the vector

$$\boldsymbol{x} = (x_1, ..., x_N)^{\mathrm{T}} \tag{1.1}$$

The optimization process implies that there is an objective function $F_0(\boldsymbol{x})$ that can be improved and that provides a basis for choice between alternative acceptable designs. Typically it reflects the cost of a system or some simplified criteria such as cost or weight.

The constraint functions impose limitations on the behavioural characteristics, and can be various response quantities such as stresses, aerodynamic drag, etc.. Constraints can be introduced as *inequality constraints* and/or *equality constraints*. Since any equality constraint can be reformulated as two inequality constraints, only the latter are considered. The constraints that impose lower and upper limits on the values of design variables are called *side constraints*.

Once the design variables, objective and constraint functions have been defined, the optimization problem can be stated as:

Minimize

$$F_0(\boldsymbol{x}), \quad \boldsymbol{x} \in R^N \tag{1.2}$$

subject to the constraints

$$F_j(\boldsymbol{x}) \leq 1 \quad (j = 1, \ldots, M) \tag{1.3}$$

and the side constraints

$$A_i \leq x_i \leq B_i \quad (i = 1, \ldots, N) \tag{1.4}$$

It is preferable to consider the constraints in a dimensionless form, e.g. stress constraints can be formulated as

$$\frac{\sigma_{eq}(\bar{\boldsymbol{x}})}{[\sigma]} \leq 1 \tag{1.5}$$

where $\sigma_{eq}(\bar{\boldsymbol{x}})$ is an equivalent stress and $[\sigma]$ is the allowable stress.

1.2 The problem

Numerical techniques became usual tools for the analysis of engineering systems. After being able to analyse the behaviour of the system modelled, for example, with a finite element method, an important goal of the designer is to improve and to optimize its performance. However, the application of these techniques to real-world problems suffers, in practice, from two difficulties:

- high computational cost of the response analysis,
- presence of noise in the response data.

The first problem has been mitigated in the mid-seventies by introducing approximation concepts (Schmit and Farshi, 1974) to replace the objective function and/or the constraints of the problem with less expensive models. Response surface methodology is a technique that traditionally uses polynomial models (to be used as approximations) created by performing a least-squares fit into a set of data, thus reducing the negative effect of numerical noise in the response function values.

One of the major problems in the application of approximation techniques is the necessity to select the structure of the approximation function. The most typically used linear (e.g. polynomial) and intrinsically linear functions (e.g. multiplicative) are simple and easy to use, but the quality of approximations can be low, so the overall convergence of the technique can be slow.

The problem of selection of individual regression components in the empirical model building is a combinatorial one (Box and Draper, 1987). The search through all possible combinations would result in prohibitive computational effort. The objective of this thesis is to develop a Genetic Programming algorithm for the creation of an approximation function structure of the best possible quality.

1.3 Genetic Programming Methodology

An evolutionary algorithm (EA) is a computer-based system that evolves a solution to a problem by simulating processes found in nature. One relatively new EA is genetic programming (GP). The GP paradigm deals with the problem of representation in genetic algorithms (GA) by increasing the complexity of the structures undergoing adaptation.

The use of genetic programming for symbolic regression was first proposed by Koza (Koza, 1992). The advantage of symbolic regression over standard regression methods is that in symbolic regression the search process works simultaneously on both the model specification problem and the problem of fitting coefficients, as opposed to conventional linear or nonlinear regression which involve finding the coefficients of a prespecified function form.

The genetic programming paradigm is an approach that seeks to automate the process of program induction. Different problems can be reformulated as requiring the discovery of a computer programme that produces some desired output when presented with a particular input, i.e. in this thesis finding a model that fits a given sample of data.

The basic idea is to use a computer programme to evolve the solution. Other existing paradigms like machine learning, artificial intelligence or neural networks involve specialised structures (e.g. weight vectors for neural networks). Many applications use neural network to realise self-learning models. The disadvantage of neural models is that input/output is approximated by a black box approach and no understanding of the underlying relationship can be gained.

The flexibility of genetic programming comes from its hierarchical structure, the possibility to manipulate intermediate calculations and the fact that we do not specify the size, the shape and the structural complexity of the solution in advance, but these characteristics evolve as part of the solution to the problem.

1.4 Structure of the thesis

In this thesis, a genetic programming methodology for empirical model building is presented. A review of basic approximation techniques used in design optimization is given in Chapter 2. This techniques can be local, global or mid-range depending of the range of their applicability. Focus is given on global approximations.

The response surface methodology to construct approximation models is described in Chapter 3. Two main aspects are treated: the selection of the approximation model and the plan of experiments where the response has to be evaluated. Several most useful plans of experiments are reviewed.

Chapter 4 presents the theory of genetic algorithms and genetic programming. The implementation of the methodology is detailed with the justification for the choice of an evolution strategy and the control parameters. The evaluation of the coefficients of the approximate function and its integration in the solution process is explained.

Applications of the approximation model building using genetic programming are presented in Chapters 5 and 6.

Chapter 5 applies the GP methodology to problems with numerically simulated response functions. These are responses obtained from computer experiments. Several examples are shown ranging from the approximation of known expressions to the detection of damage in steel structures modelled with a finite element method.

Applications to problems with experimental responses are described in Chapter 6. Three examples with responses measured in physical experiments are presented: approximation of design charts, prediction of the shear strength capacity of reinforced concrete deep beams and the multicriteria optimization of Roman cement.

Chapter 7 gives a general conclusion followed by recommendations for future work.