APPENDIX C

Genetic algorithms for structural design

Rajan Filomeno Coelho, Tomás Méndez Echenagucia, Alberto Pugnale and James N. Richardson

Genetic algorithms are a subclass of evolutionary algorithms. A Genetic Algorithm (GA) is defined by Goldberg (1989) as a 'search algorithm based on the mechanics of natural selection and natural genetics', but another definition, more focused on its way of functioning, is provided by John R. Koza in his 1992 book, *Genetic Programming: On the Programming of Computers by Means of Natural Selection:*

> The GA is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects [...], each with an associated fitness value, into a new population (i.e. the next generation) using operations patterned after the Darwinian principle of reproduction and survival of the fittest and after naturally occurring genetic operations (notably sexual recombination).

In other words, in a random population of potential solutions, the best individuals are favoured and combined in order to create better individuals at the next generation. In the 1980s, genetic algorithms received an increasing recognition by scientists, and studies in fields ranging from biology, artificial intelligence, engineering and business to social sciences began to appear. At present, GAs provide a robust and flexible tool to solve complex problems, including air-traffic programming, weather forecasts, share portfolios balance and electronic circuits design, in which a consolidated analytic way of resolution is unknown. Moreover, as far as the world of construction is concerned, GAs are increasingly being used to deal with the optimization of bridges and large-span structures, the morphogenesis of shells and membranes, and the spatial configuration of reciprocal structures.

C.1 Main characteristics

With respect to other traditional optimization and search procedures, GAs differ in four fundamental aspects, described by Goldberg (1989):

- GAs search using a 'population' of candidate solutions, and not a unique solution;
- GAs work with a coded version of the parameter set, and not the parameters themselves;
- GAs are based on stochastic transition rules (they use randomized operators);
- GAs are blind to auxiliary information; they only need an objective function (fitness function).

All these characteristics contribute to the typical GA's robustness.

C.2 Terminology

The specific terminology for GAs derives from natural systems as well as from computer science technical vocabulary. For this reason, it is possible to find in technical literature the same concept expressed with two different, but equivalent, terms (see Table C.1). Many of these terms specifically refer to the world of natural systems and cannot be found in other optimization procedures. Examples are terms such

Meaning	Natural systems	Computer science
genetic codes	chromosome	string
genetic constitution of an individual	genotype	structure
observable characteristics of an individual	phenotype	parameter set, solution alternative, point
basic unit of a genetic code	gene	feature, character, detector
possible settings of a gene	allele	feature value
the position of a gene in a genetic code	locus	string position

Table C.1 Comparison of natural and artificial GA terminology

as 'individual', which represents a candidate solution to the evaluated problem, 'population', indicating a set of individuals considered at the same iterative step of the evolutionary process, and 'generation', synonymous to iteration, to refer to a specific step of the algorithm procedure. This is also the case with the three main operators of GAs – selection, reproduction and mutation – which are described in Section C.3.2.

C.3 Elements of a genetic algorithm

As with any systematic approach to search problems, GAs require two main elements in order for them to provide an effective and reliable result:

1. A representation scheme, describing each possible solution (individual) with a set of variables (chromosome), as well as the limits in which these variables can operate. This is done by means of parameterization and the definition of the parameters domain.
2. A fitness function, measuring the generated solutions (individuals) on the basis of a well-defined performance parameter, and the respective evaluation criterion.

The definition of a problem for a GA implementation requires also several parameters and termination criteria to control the algorithm. The parameters include the number of generations, the population size, the number of parents and various coefficients that are applied on the GAs operators (selection, crossover, mutation, elitism). A termination criterion can be related to the fitness function (the algorithm can stop once a minimum required fitness value has been reached), it can be related to the number of solutions considered, number of generations, or even calculation time.

C.3.1 Procedure

The conventional procedure of a GA can be summarized by the following steps:

1. Generate an initial, random population of individuals, or candidate solutions to the problem.
2. Evaluate the performance (fitness) of each individual.
3. Generate a new population of candidate solutions applying the following three genetic operators (or at least the first one):
   a. Selection: select best individuals for the reproduction to the new population.
   b. Crossover or reproduction: recombine genetic codes of selected individuals, creating new candidate solutions.
   c. Mutation: apply random mutations to the genetic codes of new individuals.
4. Repeat steps two and three to evolve the population over a number of generations, until a satisfactory result is achieved.

C.3.2 Operators

Successive populations are generated from the previous population by way of three genetic operators: the 'selection' of best individuals, their 'reproduction' or 'crossover', and 'mutation'. In order to further improve the general efficiency of the algorithm, some other secondary operators could be added to the main procedure, most commonly the 'elitism' operator. It is worth noting that the various user-defined parameters controlling these operators will be of great importance to the efficacy of the algorithm. As yet no universal

method exists to optimally choose these parameters and the experience of the user will play a dominant role in the choice of values for the parameters.

Selection

The selection operator retains individuals from the previous population. Different methods have been developed which all employ the fitness values in their selection, while some methods also incorporate a degree of randomness. Individuals with higher fitness generally have a greater probability of making a contribution to the new generation of individuals. The end result of the selection operator is the 'mating pool'; it is a list of pairs of individuals which are to be used in reproduction (and crossover).

Crossover

The reproduction or crossover operator acts on the selected individuals, creating a new population of individuals. Crossover involves the 'mating' of individuals to produce offspring with characteristics of its predecessors. The offspring individual is created by copying part of the genetic code of one parent, and the rest from the other parent (Fig. C.1). Different crossover operators perform this operation in diverse ways, introducing some level of randomness. The probability of a crossover operation taking place is set by the user. This operator is traditionally considered as the 'core' of the GA, because it is the main cause of variation and innovation of candidate solutions.

Mutation

The mutation operator performs genetic variations in chromosomes of the individuals of a population. The mutation operator is also implemented with a probability chosen by the user. Similarly, the scale of the mutation can also be set. Mutation consists of replacing entries in the chromosome by random values within the permitted range of the variable. The mutation rate is usually relatively small compared to the crossover rate, as is the mutation scale, the relative number of mutations in the chromosome. While crossover is generally considered as a constructor of new candidate solutions, mutation works as a disruptor of existing configurations; for this reason, it plays a secondary role with as main purpose avoiding genetic drift (when all individuals become identical, evolution is no longer possible).

Elitism

The 'elitism' operator, introduced by Kenneth Alan De Jong in his 1975 doctoral thesis 'An analysis of the behavior of a class of genetic adaptive systems', is the most common secondary operator. It works in addition to the selection method, forcing the GA to retain a fixed number of best individuals at each generation in order to save their chromosomes from destruction due to crossover and/or mutation, therefore avoiding a maximum performance decrease during the evolutionary process. Generally, it significantly improves the algorithm's efficiency.

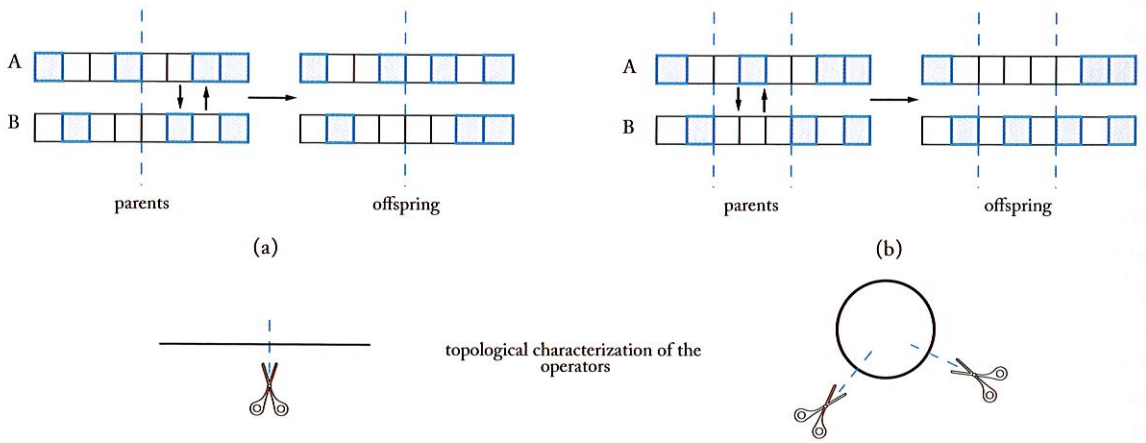

Figure C.1 Simple (a) single-point and (b) two-point crossover operators

C.4 Multi-objective genetic algorithms

When we study multiple and contrasting objective functions, with any search method, we have to consider the fact that the solutions will not be optimal for all functions. Therefore, the final result of a multi-objective search is inherently a set of solutions, not a single individual. This group of solutions is called 'trade-off set', 'Pareto front' or 'non-dominated set'. It comprises solutions that are said to be not dominated. The concept of dominance and non-dominance is defined as follows:

> In order for solution A to dominate solution B, solution A has to outperform, or equal B in all functions, as well as outperform B in at least one function. If solution A outperforms or equals solution B in all objective functions except in one in which solution B outperforms A, then A and B are non-dominated solutions.

GAs can easily be adapted to multi-objective search and optimization thanks to their inherent handling of multiple potential solutions, thereby leading to various trade-off designs. Different ways of transforming a GA into a Multi-Objective Genetic Algorithm (MOGA) by modifying some of the operators have been proposed (Deb, 2001). Various MOGAs employ different genetic operators in order to introduce the characteristics of multi-objective search and optimization, but one of the most important modifications in most MOGAs is done in the selection operator. The main differences in the selection operator (see Section C.3.1) are:

- Pareto-optimality: solutions are not ranked by their performance in the objective functions directly, but are ranked by their level of domination in the population.
- Clustering: how similar is this design to other designs in the population? Promoting diversity in the population leads to better exploration of the design space.

For a full explanation on MOGAs, the reader is referred to Deb (2001).

C.5 Application to structural design and optimization

For structural design and optimization, the use of GAs is very attractive, for the following reasons:

- The nature of the variables: structural optimization problems may be characterized by mixed variables (continuous, discrete, integer and/or categorical). GAs handle these variations naturally, whereas gradient-based algorithms, for instance, are mainly devoted to problems with continuous variables and differentiable functions.
- The nature of the functions: as the functions involved in structural optimization (e.g. the maximum stress among all elements of a truss) may be non-differentiable, and sometimes discontinuous, gradient-based techniques are excluded. Only algorithms requiring only the values of the functions, and not their derivatives, are applicable.
- Exploration of the search space: as they work on a population of solutions instead of a single point (at each iteration) – even while blind to any specific knowledge about the problem – GAs are less likely to be trapped in a local minimum, and effectively find the optimal global value. The schemata theorem has shown that the way recombination of individuals is performed allows the algorithm to explore widely the whole design space. GAs are thus very well suited for noisy and multi-modal functions.

C.5.1 Chromosomes

When considering structural problems, the variables describing each individual, that is, its chromosome, depend on the type of optimization considered. For instance, the chromosome can consist of member sizing, nodal position (shape) or topology variables, or of some combination of these three types. It can consist of bit string representation or real-valued representation, though mutation and crossover operators are typically a bit more involved for real-valued representation. The capability to handle variables of different types at once is one of the great strengths of GAs as an optimization procedure. GAs allow, for example, for combining continuous shape

variables with discrete topology variables in a single chromosome representation.

Figure C.2 shows a chromosome representation of the shape and topology variables for a two-dimensional, four-node structure. Each entry in the chromosome refers to one of the variables considered. In the example, the first four entries correspond to the shape variables x_1, y_1, x_2 and y_2, the coordinates of nodes 1 and 2. The remaining entries in the chromosome correspond to the elements defined by the nodes they connect. The binary 0/1 topology variables refer to the existence or non-existence of the element.

C.5.2 Fitness function

Besides the abstract definition of the set of potential solutions, a measure of the structural performance (used in step two of the procedure described in Section C.3.1) must be chosen in order to drive the optimization process. In single objective optimization, basic choices could be measures of mass or stiffness. The maximum vertical displacement of a structure is a suitable structural parameter to have a raw evaluation of its structural behaviour. However, analogous results could be obtained by calculating the fitness function from other integral parameters of the structure, such as total strain energy or buckling multiplier.

Furthermore, there may be structural properties associated with the chromosome that we wish to know in order to ensure that the structure adheres to certain structural requirements. For example, the maximal stresses in the structure should not exceed a certain limit, ensuring satisfaction of the stress constraint. In general, violation of the constraints will either be used to augment or penalize the fitness function, so as to influence the selection of unfeasible solutions, or eliminate them as candidate designs. The evaluation of the individual's fitness often occurs by way of structural analysis such as the force method or displacement method (e.g. finite element analysis). Whatever method of structural analysis, the evaluation will be called at least once per iteration of the genetic algorithm. It is noted though, that for very complex problems, alternative methods to evaluate the objective functions exist; for example, response surface methods, surrogate modelling. Since the GA itself is a black-box type procedure, the operators (selection, crossover and so on) are not specific to structural design and optimization.

Further reading

- *An Introduction to Genetic Algorithms*, Mitchell (1998). A simple and complete introductory book to genetic algorithms, implementing the main concepts of the work and research developed by Holland, Goldberg and Koza during the last four decades. It should be used to program a basic GA procedure, as well as for the development and tuning of the routines related to the three main GA operators.
- *Genetic Algorithms in Search, Optimization, and Machine Learning*, Goldberg (1989). The initial reference book on genetic algorithms.
- *Multi-Objective Optimization using Evolutionary Algorithms*, Deb (2001). Widely regarded as the definitive book in its field.

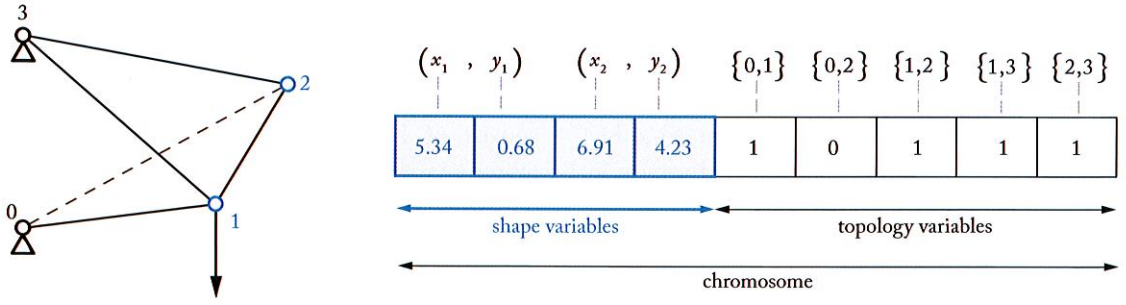

Figure C.2 Chromosome representation of a four-node, pin-jointed structure. Entries in the chromosome correspond to shape variables (coordinates of nodes) and topology variables (existence or non-existence of elements)