This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Genetic algorithms in computer aided design

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

Department of Industrial Engineering, National University of Ireland, Galway

VERSION

VoR (Version of Record)

PUBLISHER STATEMENT

This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at:
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Graham, Ian J., Keith Case, and R.L. Wood. 2019. "Genetic Algorithms in Computer Aided Design". figshare.
<https://hdl.handle.net/2134/18073>.

GENETIC ALGORITHMS IN COMPUTER AIDED DESIGN

I.J. Graham, K. Case and R.L. Wood

Department of Manufacturing Engineering, Loughborough University, Loughborough,
Leics., LE11 3TU, UK

ABSTRACT

This paper describes progress in research into the development of a computer aided design (CAD) tool that aids designers in generating the form of a product by the use of evolutionary techniques. Genetic Algorithm (GA) software has been developed and combined with a commercial CAD solid modelling system. The system creates objects that initially have the appearance of being random in form, but which can be subjected to a user-directed selective breeding programme which is also guided by pre-set internal, or environmental, factors. User scoring of each object, or an objective function, determines which objects are considered to be the 'fittest', and thus likely to become parents of the next generation.

Through the co-operation of the user and the pre-set environmental factors, the forms on the screen progressively become more than an abstract collection of geometric primitives. It is believed that this can provide useful inspiration with regard to the aesthetics and functional characteristics of products, and the potential exists for this approach to be the basis of a new design methodology.

Early work demonstrated that the software had the ability to evolve interesting shapes in line with a user's particular criteria for rating objects. However it was also obvious that the objects generated needed enhancement to convincingly represent some of the geometric complexities of real products. Using the CAD software's blend function within the evolutionary process has provided that complexity, producing excellent results and greatly widening the field of application. In addition to simply creating secondary geometry through the smoothing of sharp edges or creation of curved fillets between adjoining solids, more significant and complex primary geometric forms have been generated by allowing relatively large blend radii.

The current challenge is to combine the existing ability to predictably evolve simple geometric shapes, with the added complexity arising from the use of blends, to make the concept genuinely useful. The outcome of genetic manipulation needs to be predictable, to the extent that desirable features from objects are reproduced in subsequent generations. The key to this is the way the genetic shape defining data is stored and processed, especially the way the blend instructions are integrated into the existing genetic structure, and this is the major focus of continuing research.

KEYWORDS: Genetic Algorithms, Computer Aided Design

1. INTRODUCTION

The collaborative method of user selection is often used in conjunction with genetic techniques [1] for producing computer art. In other cases the evolutionary process is entirely automatic, with existing genetic algorithm (GA) based systems finding mechanical and structural

design solutions to meet general user-specified constraints [2]. A novel cell division modelling system [3] has been developed specifically to remove problems associated with using boundary representation (B-Rep) and constructive solid geometry (CSG) representations for GA manipulation. In common with these examples of related activity, this research has the aim of taking a step forwards from traditional design techniques [4].

The inspiration for this research arises from an interest in evolutionary computer programming, genetic algorithms and the like [5], and in computer aided design, especially as a concept modelling and development tool for consumer products. At present, CAD techniques concentrate on the detail stages of design following conceptual design. In contrast to this, the industrial designer uses a sketchpad, a practised hand and a selection of pencils and markers, perhaps moving on to cardboard, clay, and other physical media, to establish the shape of products. Although a designer may draw inspiration from other objects, natural forms, art etc., it is very much down to the individual designer to create pleasing forms using experience and artistic ability. Existing CAD modelling systems can be useful during this process, allowing a designer to experiment with form quickly and easily, but the aim of this research is to develop an interactive tool, which will assist designers to a greater extent, particularly during the conceptual stage of aesthetic design.

2. THE EVOLUTIONARY FORM DESIGN SYSTEM

The evolutionary form design (EFD) system at the centre of this work is both evolutionary and user driven, with the balance between the two being adjustable. For example, the system can calculate the relative volume of objects (using the inherent capabilities of the underlying solid modeller) and could automatically penalise ‘heavy’ objects through the operation of the genetic algorithms. Geometric models are initially constructed from a set of primitives related by boolean operators in a manner that will be familiar to the users of any conventional solid modeller. However, some of the consequences of the resulting constructional properties are considered automatically. For example, an object that consists of several non-attached parts is given a low objective function, in order to encourage coherent objects in future generations. These and other multi-objective factors combine to create the environment in which the objects evolve. User defined constraints, such as the object and primitive size limits, the number of primitives per object etc., are set using control files. Once environmental factors and initial limits have been set, the user has only to provide ratings on each new generation of objects. However, reproductive controls, such as mutation probability and crossover type (the way in which parent’s genes are combined during reproduction) can be altered during an active session. For example, reducing the mutation probability lessens the chance of losing a desirable feature that has evolved over a period of time [6]. Early work on the EFD system has been reported previously [7], and this paper emphasises recent enhancements to the technique.

The screen shot in figure 1 shows an example of a first generation of objects. These objects can be viewed individually - rotated etc., before the scores are entered in the box to the left.

The CAD software used is Unigraphics from EDS. Unigraphics provides a ‘User Function Development Environment’ known as UG-Open API which consists of user callable ‘C’ functions that access the Unigraphics system functionality. The research software is essentially a hybrid, based around a GA written in ‘C’, which manages the data structuring and genetic manipulation, and which calls Unigraphics functions to generate the geometry, display the results, and carry out user interface tasks.

Figure 1 - Screen shot showing first generation objects

3. DATA STRUCTURE

The way in which data is stored is fundamental to the behaviour of a genetic algorithm. This, and the types of reproductive techniques selected for crossover and mutation, dictate how genetic data is recombined in subsequent generations and, along with the parent selection method, control the behaviour of the system as a whole. The information that defines a geometric primitive and its interaction with surrounding primitives is made up of seven parts. These seven parts, which correspond to seven chromosomes, are summarised in Table 1.

The 'sign' chromosome controls the interaction with the other primitives that make up an object. The four operators are as follows: A created primitive is displayed, but does not physically join with the other primitives in the object. An added primitive is united with the primitives it touches. A subtracted primitive is removed from the existing primitives. An intersected primitive causes only the material that is common to the current and the existing primitives to be kept. The order in which primitives are introduced is therefore very influential. Incidentally, the first primitive is always 'created'; this is achieved by overriding the individual's sign gene.

<i>Chromosome</i>	<i>Segments</i>	<i>Numeric range</i>	<i>Decoded to :</i>
Type	1	0 to 3	block, cylinder, cone or sphere
Origin	3	0 to 50	local x, y, z co-ordinates
Sign	1	0 to 3	create, add, subtract or intersect
Direction	3	-1 to 1	x, y, z vectors, restricted to integers
Shape	3	0.01 to 1	proportions, e.g. length, width, height
Size	1	1 to 100	multiplier value
Blend	20	0 to 100	Edge blend radius and probability

Table 1 - Summary of genetic data structure

3.1. Objects formed from ‘Teams’ of members

Two ways of organising this genetic information have been examined. In one method, each member’s genes contain one set of the seven chromosomes described above, and are decoded to produce a single primitive. In the language of biology, the interaction of the genotype with its environment forms the phenotype, in this case, an individual geometric primitive. Several members are then grouped together to form each object. So far no analytical method has been devised to define the way these ‘teams’ group together; the first five primitives produce the first object, the second five produce the second object etc. Hence, to produce a set of twelve objects, sixty members are required. The individual members share the fitness function assigned to an object. Without any team-forming control, very little visual inheritance could be identified in the offspring and analysing trends and patterns was difficult. To be effective, this method will need a set of rules and routines, enabling meaningful collaboration between members. These rules will also be subject to the assigned and internal fitness functions causing the interactions to evolve in parallel with the individuals. These are advanced concepts, even within the field of specific GA research, and so will form the subject of future investigations. Figure 2 illustrates this approach.

Figure 2. (left) team objects and (right) single member objects

3.2. Single member objects

In order to display inheritance and continuity, a second method can be employed (figure 2). The storage method can be simplified to make each object the result of one member's genetic data (each genotype contains five sets of the seven chromosomes described above, making 35 chromosomes in all). In this case, the whole object is therefore the phenotype. This method has immediate benefits, making the whole system conceptually simpler, and allowing direct comparisons between parents and offspring. Figure 3 shows a family tree for a fourth generation object displaying the inheritance of features. Each generation normally consists of twelve objects, but the figure shows only the objects involved in the generation of the featured object 'g4p8-olive'. This particular example shows a high degree of variability, which may or may not be desirable. This can be altered to suit the application by changing variables like the mutation probability and crossover type.

This type of data structuring works well, as long as the user supplied scoring is the primary fitness consideration. However, when more environmental factors are included and the reliance on user ratings is lessened, as is envisaged for future work, the large amount of genetic data per member will create a lot of evolutionary inertia within the population, making evolution slow.

Figure 3. A family tree of a fourth generation object

4. CONCLUSIONS

Complex and aesthetically interesting objects such as those illustrated in figure 4 can be evolved using the evolutionary form design (EFD) system. Once evolutionary properties can be re-established, the aim will be to make the system more efficient, with more selection work going on ‘behind the scenes’ (at the moment, out of a group of 12 objects, there will often be quite a number of very small shapes that have suffered from ‘over-enthusiastic’ subtraction operations). This will mean giving more control to the user, in addition to just scoring each object. For example if a good object is produced, the user may wish to take this one object, and produce an entire set of offspring from this one object, using just slight mutations.

Figure 4. Six examples of evolved objects

Only very limited and informal measures of effectiveness of the EFD system's capabilities have been investigated to date. A quick evaluation can be carried out by deciding on a shape, or product, and then trying to produce the shape in mind. So far, trying to evolve a real product (except for perhaps a ‘sculpture’) is unrealistic. However, a descriptive specification, like ‘spindly’ or ‘flat’ etc. can be realised.

Further work will also involve creating more internal environmental objectives, allowing user defined functional and mechanical design constraints. This would move the work on from an aesthetic tool to a system also capable of application to a wider range of design problems. For example, figure 5 illustrates how the geometry related to the functionality of a design (the box model) might be aesthetically enhanced by the evolution of more complex shapes whilst maintaining the functionality.

Figure 5. Adding pre-set functional 'black boxes'

5. REFERENCES

1. Rowley T, A Toolkit for Visual Genetic Programming, University of Minnesota, <http://www.geom.umn.edu/~trowley/genetic/report/report.html>, 1994.
2. Bentley P J, Generic Evolutionary Design of Solid Objects using a Genetic Algorithm (PhD thesis), University of Huddersfield, 1996.
3. Taura T, Nagasaka I, Yamagishi A; Application of evolutionary programming to shape design, *Computer Aided Design*, vol 30 no 1, pp 29-35, 1998.
4. Ulrich K T, Eppinger S D, *Product Design and Development*, McGraw Hill International Editions, New York, 1995.
5. Goldberg D E, *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley Publishing Company, Inc, Reading, Massachusetts, 1989.
6. Wood R L, Genetic Algorithm Based Inverse Analysis (Internal report), Loughborough University, 1996.
7. Graham, I J, Case, K and Wood R L, Evolutionary form design: the application of genetic algorithmic techniques to computer aided product design, *Advances in Manufacturing Technology XIII*, Proceedings of the 15th National Conference on Manufacturing Research, Professional Engineering Publishing, pp 345-350, 1999.