Two Applications of Genetic Algorithms to Component Design

Phil Husbands and Giles Jermy and Malcolm McIlhagga and Robert Ives

School of Cognitive and Computing Sciences,
University of Sussex, Brighton, BN1 9QH, UK

Abstract. This paper describes work on two different aspects of the application of genetic algorithms to component design. Namely structural design optimisation and the evolution of free-form 3D shapes. On the first aspect, a thorough comparison of ten different search techniques applied to a wing-box design optimisation problem is described. The techniques used vary from deterministic gradient descent to stochastic Simulated Annealing (SA) and Genetic Algorithms (GAs). The stochastic techniques produced as good solutions as the best found by the deterministic techniques. However, only the stochastic techniques consistently produced very good solutions every run. Significantly, only a distributed genetic algorithm (DGA) and hybrid methods (SA with gradient descent, DGA with gradient descent) had a reliable fast decent to good regions of solution space. On the free-form 3D shape aspect, an interactive systems for exploring the evolution of 3D shapes is described. An important element of the systems is its use of a shape description language based on superquadric primitives and global deformations of these primitives.

1 Introduction

This paper reports on recent work on two different aspects of the application of genetic algorithms to component design. Namely structural design optimisation and the evolution of free-form 3D shapes. The first section describes an industrially-based structural design optimisation problem and then outlines a comparative study of ten different optimisation techniques used to tackle it. One of the main conclusions of this study was that a distributed genetic algorithm, and a distributed genetic algorithm hybridized with a gradient descent technique, had significant advantages over the other techniques tried.

Following that there is a description of work on the evolution of 3D shapes using a superqudrics-based shape description language. The shape description language developed is capable of describing arbitrary combinations of global deformations, Boolean set operations and superquadric modelling primitives. A genetic encoding was devised in which genomes could be translated in a non-direct manner to produce expressions in the shape description language. The reasons for choosing such a description language, and for using a non-direct genetic encoding are explained. Examples of shape evolution are given.

The final section of the paper draws conclusions from both aspects of the work.

2 A Comparative Study Using a Structural Design Optimisation Problem

Many engineering design optimisation problems involve search spaces with a highly complex structure. Stochastic techniques such as simulated annealing and genetic algorithms have been successfully applied to a number of these [4, 12], and folklore states that these are among the sorts of problems to which these techniques are best suited, having advantages over other methods [7, 4]. However, with a few notable exceptions [9], there has been a tendency to neglect large-scale comparative studies of many optimisation techniques applied to problems of this nature. We believe that studies of this kind are necessary in order to gain greater insight into the relative merits and weaknesses of different techniques, and to form an understanding of what kinds of problems they are most appropriate for. This is particularly important for genetic algorithms, where the body of empirical and theoretical work is still relatively small.

As a contribution to knowledge in this area, we made a thorough (at least 5×10^7 objective function calls per technique) comparative study of ten optimisation techniques applied to an industrially-based structural design optimisation problem. The search techniques used were: random search; brute force search; local gradient descent (moving in the first down hill direction encountered); gradient descent (moving in the best single move down hill direction); Powell's method [13]; simulated annealing [10]; a distributed genetic algorithm (DGA) [3]; the Alopex method [15]; hybridized simulated annealing and gradient descent (gradient descent to a local optima after every SA move); and hybridized DGA and local gradient descent (local gradient descent on each new individual in the population – altering their genetic material).

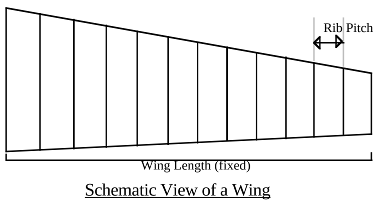

Fig. 1. Schematic view of wing

The wing box problem can be simply described as the task of finding the number of panels needed in an aircraft wing topskin and the thickness of each of those panels while minimising the mass of the wing and ensuring that none of the panels buckle under maximum operational stresses. Figure 1 shows a schematic plan view of the wing discussed in this section.

It was assumed that the topskin panels can be modelled as flat plates, and that a panel undergoes uniform compressive stresses. It was also assumed that the spacing between ribs (rib pitch) is uniform and that the thickness of all ribs is the same. Hence the aim is to find the rib pitch, dictated by the number of panels, and the thickness of the top panels, such that the structure has minimum weight and will not buckle under the compressive stresses produced by the bending moments of a 2g manoeuvre.

2.1 The Objective Function

The objective function (to be minimised) used throughout is given in Equation 1:

$$C = \sum_{i=1}^{i=n} M_i(P_i + 1) \; + \; RM \tag{1}$$

Where n is the number of panels, M_i is the mass of the ith panel, P_i is the penalty term described in Equation 2 and RM estimates total rib mass.

$$P_i = \begin{cases} \frac{\sigma_i}{\sigma_t^i} \text{ if } \; \sigma_i > \sigma_t^i, \\ 0 \;\; \text{otherwise.} \end{cases} \tag{2}$$

Where σ_i is the compressive stress in the ith panel and σ_t^i is a threshold stress that must not be exceeded in that panel. For full details of the stress and rib mass calculations see [11].

2.2 Solution representation - the delta encoding

A solution to the problem is represented by defining for each section, apart from the one nearest to the fuselage, a term by which the thickness of that section should be increased/decreased from the previous one.

Figure 2 illustrates the encoding. $\delta_{n-th\ panel}$ is the amount by which the thickness of the n-th panel is bigger or smaller than that of the (n-1)th panel.

Number of Ribs	Thickness of 1st Panel	$\pm\partial_{\text{2nd Panel}}$	...	$\pm\partial_{\text{Nth Panel}}$

Fig. 2. Delta Encoding used by all Search Techniques

Most of the search techniques used in this study could not handle variable dimensionality search spaces, so they were run on separate problems defined for each possible number of panels in the range 45–65. However, the GA easily handled variable numbers of panels in a single run by having a fixed length encoding allowing a maximum of 65 panels, but specifying how many panels to consider as one of the variables. The thickness of the initial panel was defined to be in the range 5-15 mm (discretized into multiples of 0.125 mm), and $\delta_n \in \{-0.25, -0.125, 0.0, 0.125, 0.25\}$. This gives a search space of size $\sum_{n=45}^{n=65} \frac{10}{0.125} \times 5^{n-1}$, where n is the number of panels.

2.3 Results

This section presents the data generated by the various search techniques. Each technique was run for 5×10^7 evaluations to allow water-tight statistics. Note that a test with this number of evaluations for a single technique comprised many runs of the method; each run was started from a new random position in the search space. Table 1 summarises the results of the comparative study. For details of the implementations of the various search techniques see [11].

Search Technique	**Legal Solution ?**	**Best Solution (Kg)**	**Mean Solution (Kg)**	**SD Solutions (Kg)**
Random search	YES	13345.8	13663.8	336
Brute Force Search	NO	9.6×10^8	2×10^{11}	6.6×10^{11}
Local Gradient Descent	YES	12761.9	1.9×10^9	7.1×10^{10}
Gradient Descent	YES	12761.6	3×10^9	9.8×10^{10}
Powell	YES	13015.8	13456	145.7
SA	YES	12763.2	12770	9.1
DGA	YES	12859.5	12921	26.6
Alopex	YES	13077.3	5313578.3	36588508
Hybrid SA and Gradient Descent	YES	12759.5	12764.4	5.3
Hybrid DGA and Local Grad. Desc.	YES	12815	12856	40.6

Table 1. Comparison of all techniques

All of the deterministic methods (with the exception of Brute Force search) found comparatively good solutions within the liberal time-constraints allowed. However, the very large standard deviations for most of these techniques indicates that although they can find good solutions, this is very unlikely and in general these techniques have to be run many many times for very large numbers of evaluations before a high quality solution is found. It should be noted that the random search results are for one single run of 5×10^7 evaluations. The mean and standard deviation refer to the set of best-solution-so-far found during the course of the run, which necessarily monotonically decreases. Hence these figures should not be compared with those given for the other techniques. Results from the gradient descent methods give empirical credence to the belief that the search space is highly convoluted with many local minima.

The GA and SA methods proved much better at avoiding the many local optima around the 13000−14000 Kg region of the search space. Both the DGA and SA consistently found very good solutions, i.e. had very low standard deviations of best solution. The DGA and SA methods were also able to handle variable numbers of panels rather than assuming a fixed number as was the case with the other techniques.

The Alopex method did not fare well. This appears to have been because there are complex interactions between the large number of variables in the problem. Always changing all variables at once is unlikely to retain the valuable parts of previous solutions.

The DGA descended into the region of very good solutions much faster than SA; it typically reached solutions within 5% of the best found after only 100,000 evaluations, whereas SA needed several million calls to the objective function to proceed that far. Hybridized SA and Gradient Descent was as reliable as SA and found comparably good solutions. However, it produced them much faster. The performance of the hybridized DGA and Local Gradient Descent was very similar to that of the hybrid SA method in terms of reliability and quality of solution found. However, it was significantly faster again at descending into very low cost regions of the search space.

Clearly the GA and SA based methods were head and shoulders above the other techniques tried, with the hybrids providing quality, reliability and speed.

3 Evolution of 3D Objects Using a Superquadric-based Shape Description Language

3.1 Introduction

This section briefly describes a preliminary exploration of an application of the genetic search paradigm to three dimensional object design. A computer program has been developed to enable the interactive design of various interesting three dimensional objects. The advantage offered by this program over other design systems is that the user need have no familiarity with computer aided design techniques, no extended training and no knowledge of the underlying algorithms; potentially allowing the exploration of designs which would be very difficult, if not impossible to generate by traditional methods. The devised genetic representation of the three dimensional objects and the genetic operations employed, result in a program with which a novice user may explore the vast space of three dimensional objects available by simply acting as critic to the program's efforts. At any time a collection of objects are simultaneously displayed and the user indicates preferences from amongst these. The program then generates and displays a further collection of objects whose underlying descriptions are based on those chosen, through genetic operators such as mutation and crossover recombination. Differences between this work and previous work [5] include the type of three dimensional objects used. The objects generated by this work are constructed from deformable superquadrics combined using constructive solid geometry. This is the first use of superquadric primitives [1] or global deformations [2] employing a genetic design approach. Global deformations were implemented in order to extend the range of shapes representable. The deformations taper, twist and stretch objects as if modelling with clay. The parameters controlling the deformations alter the effects smoothly, again suggesting the parameters could successfully be put under genetic control. The few parameters needed for a given deformation extends the range of shapes representable significantly whilst retaining the compactness of representation.

3.2 Superquadrics

The primitive shape chosen for this work was the superquadric. All other shapes depicted are combinations and deformations of superquadrics. Superquadrics were chosen for the range of shapes they encompass simply by altering their two shape parameters. Superquadrics are an extension of the superellipse, a curve discovered by D. Gardiner in 1965 [6]. The implicit equation of the superellipse is given in equation 3.

$$\left|\frac{x}{a}\right|^{\frac{2}{e}} + \left|\frac{y}{b}\right|^{\frac{2}{e}} = 1 \tag{3}$$

For $e = 1$ this is an ellipse (circle at $a = b$), for $0 < e < 1$ the shape becomes nearer a square and for $e > 1$ the shape becomes progressively more pinched. Barr introduced the superquadrics, a 3D extension of the superellipse, for use in computer graphics [1]. In fact superquadrics embrace the superellipsoids, supertoroids, and superhyperboloids of one and two sheets. Superellipsoids only were considered in this project. The implicit equation of a superellipsoid is given in equation 4.

$$F(\underline{r}) = \left(\left|\frac{x}{a_1}\right|^{\frac{2}{e_2}} + \left|\frac{y}{a_2}\right|^{\frac{2}{e_2}}\right)^{\frac{e_2}{e_1}} + \left|\frac{z}{a_3}\right|^{\frac{2}{e_2}} - 1 = 0 \tag{4}$$

The standard parameterisation as a spherical product of two superellipses is given in equation 5.

$$\underline{r}(\eta, \omega) = \begin{bmatrix} a_1.cos^{e_1}\eta.cos^{e_2}\omega \\ a_2.cos^{e_1}\eta.sin^{e_2}\omega \\ a_3.sin^{e_1}\eta \end{bmatrix} \tag{5}$$

The parameters $-\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}$, $-\pi \leq \omega \leq \pi$ correspond to deformed latitude and longitude as the surface of the superellipsoid is traced out; a_1, a_2 and a_3 are scale parameters controlling how stretched the shape is in the x, y, and z directions; e_1 and e_2 are the shape parameters controlling how round, square or pinched the superellipsoid is. By varying e_1 and e_2, spheres, cuboid and cylindroids can be produced [1]. To overcome difficulties encountered with surface sampling of superquadrics, a new reparameterisation of the superquadric modelling primitive was devised and is presented in [8]. This reparameterisation samples the surface of superquadrics more evenly than the standard parameterization [1].

Inside / outside function A modified form of the superellipsoid inside / outside function is used - (see equation 6).

$$F(\underline{r}) = \left(\left(\left(\frac{x}{a_1}\right)^{\frac{2}{e_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{e_2}}\right)^{\frac{e_2}{e_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{e_2}}\right)^{\frac{e_1}{2}} - 1 \tag{6}$$

Where,

$$F(\underline{r}) \begin{cases} < 0 \Rightarrow & \underline{r} \text{ is inside superellipsoid,} \\ = 0 \Rightarrow & \underline{r} \text{ is on surface of superellipsoid,} \\ > 0 \Rightarrow & \underline{r} \text{ is outside superellipsoid.} \end{cases} \tag{7}$$

This modified function does not change the position of the surface, but does result in a better behaved function, being closer to the standard Euclidean metric.

3.3 Global Deformations

Adaptations of the global deformations described in [2] were implemented, namely a smooth tapering deformation and a smooth twisting deformation. Also new bending and cavity deformations were added to the set of possible global deformations. The following transformations are currently implemented: translation, rotation, scaling, reflection, tapering and twisting. All transformations are functions taking a point in 3D space as input and outputting a point in 3D space. Figure 3 shows a smooth tapering, offset from the z axis, of a cylindroid superquadric primitive.

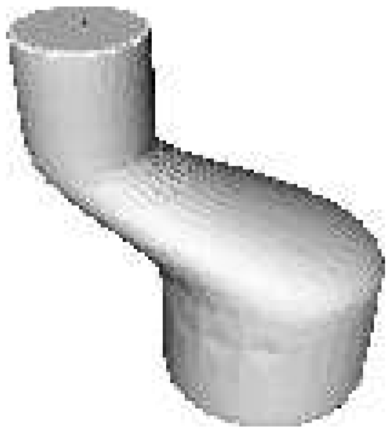

Fig. 3. Smooth tapering, offset from z axis, of a cylinder

3.4 Shape Description Language

A description language was developed capable of describing arbitrary combinations of global deformations. This is achieved through Boolean set operations of superquadric modelling primitives and meta-constructs, allowed through a recursively defined description language. The shape description language is an adaptation of standard constructive solid geometry representation of objects [14]. The adaption permits global space transformations and deformations to be inserted in the expression or tree that represents an object.

Definition of a Valid Shape Description Expression A legal expression in the language is defined recursively as -

- EITHER A primitive - This currently must be a superquadric primitive, SPQD(e1,e2) , where e1 and e2 are numbers giving the two shape parameters of the superquadric. Note, as previously mentioned spheres, cuboids and cylindroids are all realisable as superquadrics. (The NULL( ) primitive is also valid describing an empty shape.)
- OR A transformation acting on a legal expression - *transformation(param1,param2,..)legal_expression*. Where *transformation* is one of the provided transformations and *param1,..* is the required parameter list for that transformation. The transformations currently implemented are SCALE, ROTX, ROTY, ROTZ, REFX, REFY, REFZ, TRANS, TAPERZ, TWISTZ, BENDY. See [8] for a full description of the transformations.
- OR A Boolean set operation of two legal expressions - *Boolean_operation(legal_expression1,legal_expression2)*. Where *Boolean_operation* is one of UNION, INTERSECT and DIFF, for set union, intersection and difference of the two shapes described by *legal_expression1* and *legal_expression2*.

For example, with this recursive definition the following expressions are all valid legal expressions:

1. SPQD(1,1) - Defines a single sphere.
2. ROTX(1.57)SCALE(1,1,2)SPQD(0.1,1) - Defines a cylinder like shape that has been scaled by two in the z direction and then rotated by 1.57 radians around the x-axis. Note transformations are applied right to left.
3. UNION(SPQD(1,1), ROTX(1.57)SCALE(1,1,2)SPQD(0.1,1)) - Is the union of (1) and (2).
4. REFX( ) DIFF(ROTX(1.57)SCALE(1,1,2)SPQD(0.1,1), SPQD(1,1)) - Demonstrates that any valid expression can have a transformation applied.

Internally such expressions are held as a linked tree for traversal and interrogation by the polygonisation process. Each node in the tree contains a pointer to its parent node, and one or two pointers to its children node(s). Transformations always have one child, Boolean operations always have two. Transformation nodes and primitive nodes also contain the necessary parameters and also contain pointers to their relevant functions for interrogation of the shape description tree.

3.5 Genetic Encoding

A genetic encoding was devised in which genomes could be translated in a non-direct manner to produce expressions in the shape description language. The translation from genome to phenome involves a recursive process to encourage interesting repeated shapes to emerge. Both the structure of expressions and parameter values within the expressions are determined by the genome. The genetic algorithm implemented uses mutation or crossover on the selected genome(s) respectively. Any genome produced by the genetic algorithm is translated into a valid expression in the shape description language. A genome is a string of nodes, each node stores several items of information. Principally pointers to one or two nodes in the string: this has an interpretation as a directed network. The network may contain cyclic elements including nodes pointing at themselves. Figure 4 indicates how a string of three nodes represents a directed network.

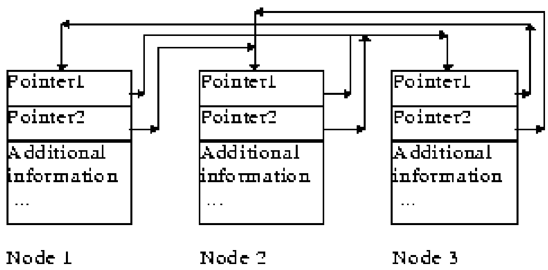

Fig. 4. Nodes in genome represent a directed network

The network is traversed recursively, starting from the first node in the string to produce a shape description expression. Each time the process descends through a node its recursive count is incremented by one. If the recursive count of a node becomes greater its recursive limit then that node does not get interpreted as an atom of primary node type, instead the node gets interpreted as a primitive of type default primitive type, with parameters default parameters. In this way the recursive traversal of the network will stop at the end of a branch.

Genetic Operators The genetic algorithm applies mutation or crossover at each breeding cycle as selected by the user. All genetic operations act on the string of bits that code for the genomes. If mutation is selected a new population is produces by producing imperfect copies of the bit string of the selected parent. Mutation rate is per bit and typically 0.005 is used. Mutation can affect one or more parameter values within a shape description expression, or can mutate the structure of the expression itself. If crossover is selected a new population is produced by applying two point crossover on the two parent's bit strings. The crossover points are uniformly distributed along the bit strings. No mutation is used on a crossover breeding. Crossover will result in a genomic network having some nodes from each of the two parents while also perhaps including nodes that are 'new' as a result of crossover points occurring in the middle of a node's bit string. The one parent selected for mutation and two for crossover, are preserved intact into the next generation.

3.6 Example

Figures 5 – 10 show the evolution of a shape over a number of generations. The 'fitness function' was the eye of the beholder.

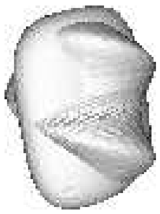

Fig. 5. Generation 0

Fig. 6. Generation 1

Fig. 7. Generation 2

Fig. 8. Generation 3

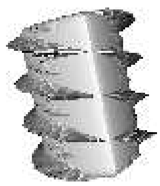

Fig. 9. Generation 4

Fig. 10. Generation 5

4 Conclusions and Discussion

As far as we are aware, the wing-box study is one of the most extensive comparative studies of different optimisation methods applied to a problem of this nature. However, because there are so many parameters associated with the various techniques that could be altered, it is inevitably far from exhaustive.

Nevertheless, it does suggest strongly that hybridized stochastic search is the correct route for problems such as the simplified wingbox design task. One obvious direction for future work, then, is a more thorough investigation of hybrid algorithms.

Past work with three dimensional shape design using genetic algorithms has been restricted to a small fixed set of primitives such as spheres and cylinders and only linear transformations. The use of superquadrics with global deformations was chosen as a powerful method of representing three dimensional shapes. Previous work with these tools has demonstrated some of the generality of this representation but has been restricted to hand-design of shapes. This research has combined these two methods of exploration and representation as the working system briefly described here. The creativity of the genetic algorithm was used to explore the expressiveness of the representational scheme. Various intriguing and sometimes suprising shapes have been produced by the program, some of which have been shown. The full potential of this approach wiil only be realised once extensive exploration and extension of the system have been fulfilled. Such a future system may eventually rival/complement traditional methods and advance the field of computer aided design.

Acknowledgements

This research was supported by EPSRC grant GR/J40812.

References

1. A. Barr. Superquadrics and angle preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
2. A. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–30, 1984.
3. R. Collins and D. Jefferson. Selection in massively parallel genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth Intl. Conf. on Genetic Algorithms, ICGA-91*, pages 249–256. Morgan Kaufmann, 1991.
4. L. Davis. *The Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1990.
5. R. Dawkins. *The Blind Watchmaker*. Harlow Logman, 1986.
6. M. Gardiner. The superellipse: a curve that lies between the ellipse and the rectangle. *Scientific American*, 213:222–234, 1965.
7. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts, USA, 1989.
8. G. Jermy. Evolutionary design of three dimensional objects. Master's thesis, School of Cognitive and Computing Sciences, University of Sussex, 1995.
9. A. J. Keane. Structural design for enhanced noise performance using genetic algorithms and other optimisation techniques. In R. Albrecht, C. Reeves, and N. Steele, editors, *Proceedings of ANNGA93, the Intl. Conf. on Neural Networks and Genetic Algorithms*, pages 536–543. Springer-Verlag, 1993.
10. S. Kirkpatrich, C. Gelatt, and M. Vecchi. Optimisation by simulated annealing. *Science*, 220:671–680, 1983.

11. M. McIlhagga, P. Husbands, and R. Ives. A comparison of search techniques on a wing-box optimisation problem. In *Proceedings of PPSN IV*. Springer Verlag, 1996.
12. D. Powell and M. Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In S. Forrest, editor, *Proc. 5th Int. Conf. on GAs*, pages 424–431. Morgan Kaufmann, 1993.
13. W. Press, W. Vetterling, S. Teukolsky, and B. Flannery. *Numerical recipes in C (2/e)*. CUP, 1992.
14. J. Snyder. *Generative Modelling for Computer Graphics and CAD*. Academic Press, 1992.
15. K. Unnikrishnan and K. Venugopal. Learning in connectionist networks using the alopex algorithm. In *Proceedings IJCNN 1992*, pages I–926 – I–931. IEEE Press, 1992.

This article was processed using the LaTeX macro package with LLNCS style