TITLE PAGE

Title: A Voxel Based Representation for Evolutionary Shape Optimisation

Peter Baron and Robert Fisher

Department of Artificial Intelligence, University of Edinburgh, 5 Forrest Hill,

Edinburgh EH1 2QL, Scotland, UK.

Email: {peterba, rbf}@dai.ed.ac.uk

Andrew Tuson (Corresponding Author)

Department of Computing, City University, Northampton Square,

London, EC1V 0HB, England, UK.

Email: andrewt@soi.city.ac.uk, Tel: ++44 (0)171 477 8164, Fax: ++44(0)171 477 8587

Frank Mill and Andrew Sherlock

Department of Mechanical Engineering, University of Edinburgh, Sanderson Building,

The King's Buildings, Mayfield Road, Edinburgh EH9 3JL, Scotland, UK.

Email: {F.Mill, A.Sherlock}@ed.ac.uk

Number of Pages: 30 (inc. refererences + this page, excl. figures)

Number of Tables: None

Number of Figures: 13

ABSTRACT AND KEYWORDS PAGE

Title: A Voxel Based Representation for Evolutionary Shape Optimisation

Abstract

A voxel-based shape representation when integrated with an evolutionary algorithm offers a number of potential advantage for shape optimisation. Topology need not be predefined, geometric constraints are easily imposed and, with adequate resolution, any shape can be approximated to arbitrary accuracy. However, lack of boundary smoothness, length of chromosome and inclusion of small holes in the final shape have been stated as problems with this representation. This paper describes two experiments performed in an attempt to address some of these problems. Firstly, a design problem with only a small computational cost of evaluating candidate shapes was used as a testbed for designing genetic operators for this shape representation. Secondly, these operators were refined for a design problem using a more costly finite element evaluation. It was concluded that the voxel representation can, with careful design of genetic operators, be useful in shape optimisation.

Keywords: shape optimisation, evolutionary algorithms, voxel representation.

1. Introduction

Shape optimisation attempts to find an optimal shape for a component subject to design constraints. Typical problems that are of interest to the research community in this area have been concerned with structural load bearing components and aerodynamic profiles. Some work has also been reported in areas such as thermal conduction for heat sinks and manufacturing cost minimisation. In structural shape optimisation, often these studies aim to minimise the amount of material (and hence perhaps cost and weight) needed to support a given load. In aerodynamic optimisation, often the aim is to minimise drag subject to constraints on lift and geometry. Almost all of the work to date has described shape representations for single criterion optimisation although many researchers are interested in multi-criteria problems.

Structural shape optimisation can be usefully characterised as the integration of geometric modelling, structural analysis and optimisation algorithms (Hsu 1994). The finite element (FE) method is popularly used to analyse candidate shapes. In early research in shape optimisation the FE mesh itself was used as the geometric model to be manipulated by the optimiser. Optimisation techniques then available were based on mathematical methods of function optimisation, typically gradient based. The nodal co-ordinates of the FE mesh were used as design variables. However it soon became apparent that use of the mesh as the geometric model was impractical due to difficulties in ensuring that the mesh could adequately calculate stresses and in keeping the shape's boundary smooth. Researchers moved to separating the geometric modeller and the FE mesh. Commonly the boundary of the component is modelled using splines, with control point co-ordinates used as design variables. Splines have the useful property of smoothness and local shape control. Mesh generation techniques then generate an adequate mesh given a description of the candidate shape's boundary.

Gradient-based optimisers can find optima with very few design evaluations. This is often extremely important in engineering problems, where the time taken to perform one design evaluation is often many orders of magnitude greater than the time taken to produce candidate designs. However, such optimisers can often have difficulties in dealing with local optima, discrete design variables and with noise generated when small changes in the design variables cause changes in mesh topology. Recently, to address these problems, the use of stochastic optimisation techniques, such as genetic algorithms (GAs) due to Holland (1975), and simulated annealing (Kirkpatrick, 1983), in shape optimisation (Chapman, 1994; Smith, 1995b) has been a popular area of research. Generally, this research has still retained a parameterised description of the shape's boundary as the geometric model.

The work described in this paper investigated the possibility of replacing this boundary representation of the shape with a cellular representation. The cellular representation chosen in this work used voxels which partition the design space into rectangular regions or boxes that are then assigned a binary full or empty value. This approach was motivated by a number of potential advantages (Smith, 1995b):

- any shape can be represented to an arbitrary accuracy by increasing resolution,

- it is straightforward to convert existing engineering solutions into voxels,
- they map naturally to the representations frequently used by GAs,
- domain knowledge can be readily incorporated,
- geometric constraints can easily be applied, and,
- the topology of candidate shapes is not predefined.

However, in contrast to the successful application of this technique in (Farrell, 98) for the inversion of geographical and potential-field data, earlier work by Watabe and Okino (1993) states the following objections to the scalability of voxel representations:

- the occurrence of small holes in the final shape,
- the long length of the chromosomes,
- the expectation that crossover operators would be ineffective, and,
- the lack of smoothness in the shapes' outlines.

Given the potential advantages of a voxel representation, the Authors considered it worthwhile to address these difficulties. Specifically, the aims of this work were:

- to determine the suitability of voxels as a geometric model for use in shape optimisation and any difficulties, such as those outlined above, that may arise;

- to design suitable operators for a GA optimiser to use with such a representation to overcome such difficulties;

- to investigate and identify issues that will have to be confronted by the practitioner in scaling up this representation to real-world problems.

Therefore this work does not aim to produce a system that returns a usuable, improved solution to a real-world problem. Instead it concerns itself with the more strategic and scientific question of investigating and, where possible, resolving issues that pertain to *how* a practitioner is to construct such a practical system.

1.1 Experiments

Two experiments were devised in order to investigate the voxel representation. Firstly, a simplified beam design problem was formulated for which the cost of evaluation would be small. Using this problem as a test-bed, a number of operators were designed. Secondly, an annulus design problem was tackled using a finite element analysis. The computation cost of evaluation in this case was thus much greater. The usefulness of the operators designed in the first experiment could then be evaluated with a more difficult design problem and related scalability issues investigated. Baron (1997) gives comprehensive details of all experiments undertaken.

Finally, this investigation will restrict itself to examples where two-dimensional voxels (pixels) are used. This is for reasons for convenience and speed of solution evaluation as FE analyses in three dimensions are more computationally demanding. However, no assumptions are made in this study regarding the dimensionality of the problem and so the results presented here should be generalisable to higher dimensional problems.

2. Simplified Beam Design

A prototypical mechanical engineering problem is that of optimising a beam to support various loads with a minimal amount of material. Evaluation of the candidate cross-sections was made using bending theory for symmetrical beams, considering only normal stresses (Gere and Timoshenko, 1984). This is an oversimplified model, but is sufficient to test whether the potential problems with a voxel representation outlined above do pose a problem in practice. The maximum stress constraint imposed by the physics model used in these experiments is summarised below.

$$\left|\frac{My_i}{I}\right| < \sigma_{\max} \qquad \textit{for all voxels}$$

where σ_{max} is the maximum stress allowed within any given area (voxel); M is the bending moment; y_i is the distance of the voxel *i* from the neutral axis of the shape; I is the second moment of area of the candidate cross-section. The neutral axis of a shape is defined as a horizontal line which passes through the centre of mass of the shape. As a voxel representation uses areas which are all of uniform size and density, the centre of mass can be found by taking the average of the positions of all occupied voxels. The second moment of area is approximated in the discrete representation by summing the moments of each voxel, that is:

$$I = \sum_{i=0}^{n} a y_i^2$$

where a is the area of a voxel.

In the real world, the solution to this problem would correspond to an I-beam, but that also requires a web to connect the two flanges of the beam together. In a design based on a full calculation with shear stress, the web would be necessary so to counteract this additional stress. However, as shear stress is not represented in this problem, a connectivity requirement in the form of a repair step was added, whereby all pixels must be connected to a seed pixel in the centre top edge of the beam. In addition, all vertically central voxels were enabled to provide a straight web before the connectivity repair step. This was found, in formative experiments, to prevent the formation of a crooked web (as the physics model used does not prevent this), and improve slightly the results obtained.

To try to ensure that the alterations and improvements made to the GA here will also prove beneficial to the real-world problem, it was decided not to concentrate on fine-tuning any of the various parameters available but rather to focus on the design and operation of various new operators. Therefore, parametric variations were restricted to an absolute minimum and were used only to determine the approximate values required to gain reasonable advantages from the new operators. Therefore in the following experiments, the following parameter settings remain constant unless mentioned otherwise:

Beam Dimensions	$= 0.05 \times 0.10$	m
Bending Moment	= 13000	Nm

Voxel Grid	$= 32 \times 64$	voxels
Max. Stress Allowed	$= 2 \times 10^8$	Nm^{-2}

2.1 Experiments Using the Naïve GA.

The first set of experiments with a 2D representation treated the chromosome as a long one-dimensional binary string which wrapped around at the vertical edges onto new lines to form the two-dimensional cross-section. Standard two-point crossover (p_c = 0.35) and bitwise mutation (p_m = 0.001) were used in conjunction with a generational GA with a population of size 20. GENITOR-style rank-based selection (Whitley, 89) was used throughout. From the above, the fitness function, F, to be minimised was of the following form:

$$F = V + S/(1000 \times \sigma_{max}) + k \times \max\{(S - \sigma_{max}),0\}$$

where V was the count of active voxels (proportional to weight), S the maximum stress of any voxel, σ_{max} the value of the maximum stress constraint, and k the constraint penalty multiplier (set to 5×10^{-5} according to the results of formative experiments).

With this particular optimisation problem, the difficulty lay not in getting a valid solution, but in getting a near optimal-mass solution. The first experiments were relatively unsuccessful in this regard: the results after 2000 generations were full of small holes and had extremely uneven inner edges. This can be seen in the typical end-of-run results shown in Figure 1 (the numbers represent the fitness values of each individual).

[Figure 1: Typical End of Run Results from the Naïve GA]

The stresses were concentrated at the vertical extremes of the beam, so the material in the middle contributes less towards the beam's ability to withstand the load, and therefore as we are trying to minimise the mass of the beam, the material is more usefully employed at the extremes of the beam. The GA, even in this simple standard form, rapidly removed material from the middle of the cross-section, and in the later stages of the experiments was observed to be moving material from low stress areas into high stress areas where holes were left near the extremities.

However this first naïve GA approach took an extremely large number of evaluations in order to make significant progress, and this is not acceptable as later experiments would have a greatly increased evaluation time due to the integration of the FE package. The rate of improvement was also seen to decrease as the run continued, levelling off to almost none at all by the end of the run. This means that the GA was not finding any further improvements to the chromosome and, as the results are visibly poor, it indicates a general weakness in the operators being applied.

Attention was therefore concentrated towards improving the GA operators, in order to achieve greater benefits during the early search period, and to produce better quality final results.

2.2 The Smoothing Mutation Operator

The smoothing operator experiments were an attempt to address directly some of the weaknesses of the voxel representation by devising a new specialised operator, which should aid the search by reducing the number of small holes and ragged edges produced by the GA. The new operator was intended to be capable of easy expansion from two-dimensions to n-dimensions in order that it would continue to be useful in the case of higher dimensional problems using the voxel representation.

This operator selects a rectangle with both random position and size ranging from 2 pixels to 1/4 of the dimensions of the grid. The most common value for the pixels in the area selected was then found and written to all of the pixels in that area (Figure 2).

[Figure 2: The Smoothing Operator]

The GA parameters used were the same as before and the new operator was applied in addition to the previous mutation and cross-over operators – application of this operator to 60% of the chromosomes in the population was found, in formative experiments, to give the best results. The GA configuration was otherwise unchanged, though the number of generations was limited to 1500 in this case.

Comparing Figure 3 which displays some typical end-of-run population members with earlier results (shown in Figure 1), shows just how effective this domain specific approach to operator design has been, especially at eliminating isolated holes and reducing ragged edges.

[Figure 3: Typical End-of-Run Results with the Smoothing Operator]

2.3 UNBLOX: An N-dimensional Crossover Operator.

The two-point crossover operator which had been used up to this point treated the chromosome as a one-dimensional string of bits and therefore suffered from a problem with linkage - voxels which are adjacent in a two-dimensional grid are not necessarily adjacent in the one-dimensional string. This separation increases the possibility that useful building blocks (areas of the grid which contribute to a higher overall fitness evaluation) will be disrupted during the crossover procedure.

Cartwright and Harris (1993) describe the use of the UNBLOX crossover operator, which was specifically designed to overcome these limitations with conventional two-point crossover. This operator swaps a rectangular area of the grid instead of the sub-string swapped by two-point crossover. If the area overlaps an edge of the grid then it is made to 'wrap-around' to the opposite side – this convention was adopted from the orginal paper, though its effect on edge smoothing is somewhat unclear. The size and location of the area to be swapped are both selected at random, and in this implementation the area was restricted to a minimum size of two voxels per dimension in order that the operator would always have some effect when applied.

The crossover operators were used with the standard probability of 0.3 per chromosome and no changes were made to the standard algorithm or to any of the other parameter settings described earlier. The graph in Figure 4 shows the results of ten trials using three alternative crossover operators, including the UNBLOX operator. The other two crossover operators were the standard two-point crossover and uniform crossovers (Goldberg, 1989).

[Figure 4: The Effectiveness of Various Crossover Operators]

The results confirm that the UNBLOX operator does indeed perform better than either the two-point crossover or the uniform crossover techniques on this problem. The rate of descent of the UNBLOX line is quicker, indicating that the population converged to good solutions faster with this approach than with the other operators, and the eventual end result after 1500 generations had a slightly better fitness value than those produced by the other techniques

2.4 Two Dimensional Mutation Operators

A new mutation operator was designed which scrambles the contents of a randomly selected rectangular area of the voxel grid, it is referred to here as the ‘two dimensional’ operator. This operator can be easily modified to work in N-dimensions, and affects a relatively small area of the chromosome rather intensively in the selected rectangular selected area in the same way as for the smoothing mutation operator. A second, somewhat altered version of this mutation operator was also designed and tested in these experiments called the ‘two by two’ area mutation operator. This operator uses a fixed mutation square of two by two voxels and was designed to be applied only if at least one voxel in the mutation area is already active. The theory behind this operator is that most of the modifications need to be made to the surface or interior of the evolving shape and that very little benefit will result from flipping isolated voxels in the middle of the void areas. The choice of a fixed two by two area was motivated by the observation that most of the irregularities on the surfaces would fit into such an area and that with only sixteen permutations possible (four binary bits), the probability of mutating a poor quality area into a more fit variation would be reasonably high.

The new operators were again applied in addition to the original bitwise mutation operator, with a probability of 0.25 per chromosome of being applied. After each application there was a decreased probability of the same operator being applied again, with the probability of a further application being decreased to one half of its previous value each time. The experiments were performed ten times for each of the three alternative mutation combinations, over a period of fifteen hundred generations.

The graph in Figure 5 shows the effect of the two new mutation operators alongside the results obtained when neither of them was applied. The generation number is plotted along the horizontal axis and the average fitness of the best individual from the population at each generation is plotted vertically.

[Figure 5: The Effectiveness of Various Mutation Operators]

The addition of the two dimensional operator generally results in better performance than the bitwise operator alone, though the two lines do meet between generations 300 to 400. The steeper descent of the two dimensional operator line indicates that early performance was especially improved, and the final result after fifteen hundred generations is significantly better than previously. The two by two operator offers a similar rate of improvement during the early stages of the trial, a slightly better performance between generations 100 to 600 and finally converges with the two dimensional operator's line at about generation 1000. This seems to indicate that although offering early benefits to the optimisation, it is not better than the two dimensional operator in the long run.

In conclusion, two new mutation operators were designed with the particular intention of directly addressing the perceived problems with the prior optimisations. Both of the new operators were found to be more effective than the previous uninformed bitwise mutation, producing benefits to both the rate of early improvement and the final quality of solution generated.

In the absence of any other clearly distinguishing features, the two by two operator will be used during the further experiments as it offers a speed advantage over the two dimensional mutation operator outlined above.

2.5 Conclusions about the beam design problem

The results have shown that although a naïve GA does indeed suffer from the problems suggested by Watabe & Okino (1993), a small selection of operators informed only by domain knowledge about the representation, will effectively solve each of these difficulties.

[Figure 6: Typical End-of-Run Results for the Complete System]

To see whether the above improvements can be usefully combined to produce the desired behaviour, and improve further upon Figures 1 and (especially) 3, Figure 6 depicts a number of typical end-of-run results for the complete system with all operators active. Comparision with the earlier results shows that the complete system produces superior results with no holes or large protrubances. In addition, the dramically improved performance of the final system in terms of the solution quality-time tradeoff surface it exhibits is shown clearly by Figure 7.

[Figure 7: Performance Comparison Between the Naive and Final GAs]

In summary, the final system uses a normal bitwise mutation operator in addition to the two new mutation operators, smoothing, and two by two. The smoothing operator rapidly cuts away unwanted areas of material during the early stages of the optimisation and can help to smooth ragged edges and fill small holes later on. The two by two mutation operator is highly effective at both smoothing off ragged edges and at filling in small holes in the material if they occur in undesirable places. Finally, the two-point crossover operator has been replaced by the n-dimensional UNBLOX operator, to fully exploit the 2D structure of the problem.

3. Annulus Design Problem using FE Analysis

The experiments undertaken with the simplified beam design problem outlined in section 2 led to the design of effective GA operators for manipulation of 2D shapes. This section details further experiments undertaken to apply these operators to a more difficult design problem. The problem chosen was to design a jet-engine annulus. The finite element method was chosen as the design evaluation/analysis technique. Initially, for ease of implementation, the voxel shape description was directly used as the finite element mesh.

3.1 The annulus design problem.

The full original specification of this problem was taken from (Smith, 1995a). The problem is to design a jet-engine annulus, that is subjected to loading due to rotation and due to the attachment of the turbine blades to its outer circumference. The part is axisymmetric around the axis of rotation, and consequently it reduces to the two-dimensional shape optimisation problem shown as Figure 8.

[Figure 8: Annulus Axisymmetric Cross-Section]

The optimisation involved reducing the mass of the annulus whilst observing a series of four separate stress constraints at discrete locations in the annulus. The constraints relate to the hoop stresses at the inner and outer circumferences and the radial stresses along the centre line of the annulus. The stress constraints to be observed were, in descending order of importance:

Hub hoop stress	< 1330 MPa
Rim hoop stress	< 396 MPa
Inner radial stress	< 741 MPa
Outer radial stress	< 334 MPa

3.2 The Fitness Function

The GA fitness function was defined as an objective (the weight of the annulus in kg, and a factor to minimise the *total* stress, in MPa) plus a sum of penalty terms if one of the 4 stress constraints was broken. The function maximised

$$F = \Sigma_i\ \sigma_{max(i)}/(\Sigma_i\ 1000 \times S_i) – annulus\_weight - \Sigma_i\ k \times i \times max\{S_i - \sigma_{max(i)},, 0\}$$

Constraint penalties were applied if any of the four constraints limits $\sigma_{max(i)}$ were exceeded by the stress, S_i, measured (in MPa). The constraints were ordered in importance by using 4 $\times$ *k* for the most important, 3 $\times$ *k* for the second most important, 2 $\times$ *k* for the next and 1 $\times$ *k* for the least important constraint, the (decresing) order of importance was as for the contraints limits listed above.

3.3 Results from the basic system.

Again, a generational GA with a population of size 20 and GENITOR-style rank-based selection was used. The UNBLOX, smoothing mutation, and 2-by-2 mutation operators were applied sequentially with probabilities 0.3, 0.8, and 0.8 respectively (on the basis of formative experiments). A 62 by 27 voxel grid was used to represent the annulus and the constraint penalty, k, was set to 0.00005. The settings used for the annulus were:

Dimensions of design space = 0.25 x 0.05 m

Radius of hole	=	0.10	m
Blade force	=	10×10^5	N rad^{-1}
Young's modulus	=	2.238×10^{11}	N m^{-2}
Material density	=	8.221×10^3	kg m^{-3}
Revolution speed	=	1571.0	rad s^{-1}

The basic system was first applied without further modifications to the annulus optimisation. However the problem as specified was very tightly constrained, which meant that the attempts to solve this problem using random population initialisation violated all of the stress constraints by large amounts. Also, the rate of improvement in the population, when extrapolated beyond the time period allocated to the experiments, indicated that a valid solution would not be found for some considerable number of generations.

To circumvent this problem, the population was instead initialised with a selection of variations on the annulus design supplied with the original specification, which were modified further by an aggressive random mutation operator that added and removed small areas of material over the surface of the annulus design. This kind of intelligent initialisation was thought reasonable as a user will often want to start the GA with existing designs in order to see what improvements can be made. Even when a totally new shape is being designed, the user would normally have some expectation about the final form, which could easily be used to initialise the population. The intelligent initialisation approach meant that the initial population was not unreasonably far outside of the stress constraints, yet supplied the optimisation with sufficient variation that the population did not rapidly converge onto a single solution. Some of the results from this basic system can be seen in Figure 9 which shows six members of the population after seventy-five generations.

[Figure 9: Results of the Basic Annulus Optimisation After 75 Generations]

The results shown in Figure 9 were poor. The lack of symmetry around the horizontal axis and the uneven edges were just the most visible failings in this set of results. A second problem was the occurrence of large stresses at the corners of elements on the edge of the shape. These failing need to be addressed if any claims as to this representation's scalability can be made.

3.4 Improvements made to the system

Attention was now turned to resolving the issues and shortcomings highlighted by the above investigation in turn.

3.4.1 Use of Symmetry

It was known that a solution to the annulus design problem should be symmetric about a radial axis. It was therefore decided to utilise this domain knowledge and thus reduce the search space of the problem. The GA was modified to reconstruct the final shape in its entirety only when producing the element definition files to be accessed by the FE package. This simple modification reduces the search space from a typical size of 2^{2542} for a 62 voxel by 41 voxel grid, to 2^{1302} which represents a 62 voxel by 21 voxel half-grid. The central line of voxels along the axis of symmetry is not mirrored as it is now enforced by the GA to be always turned on – this also provides a guaranteed central line of elements for the stress measurements to be taken from.

3.4.2 Mesh Improvement

It was found in the initial experiments for the annulus design problem that directly using the voxel description of the geometry as the FE mesh caused problems with high stresses caused by corners in the mesh. It was therefore decided to separate the geometry model and mesh. There were several possible approaches that could have been taken. An approach which was considered was to use interpolation splines to form a smoothed edge. The voxels would then act as a 'skeleton' and the spline as a 'skin'. A mesh generator could then produce a mesh whose density could then be independent of the voxel model. However for this prototype system it was decided simply to add triangular elements at the corners. Whilst this was a far less elegant solution it was much simpler to implement.

These new triangular elements were created by specifying connections between groups of three nodes in the element connection file. These triangular elements were added to the shape at all suitable 'steps', which were identified by convolving the voxels in the shape against a series of four matching template masks. If each square in the mask matched the value of the voxels surrounding an empty voxel then the appropriate triangular element was created in the 'step'. The convolution masks and the triangles which they caused to be inserted are shown in Figure 10.

[Figure 10: Convolution Masks for Triangle Insertion Process]

3.4.3 Design of Operator to Remove Holes

The 2 x 2 mutation operator (which can either fix holes or cause them to appear) was modified to only mutate areas where, as well as at least one voxel being turned on, at least one of the four voxels is also turned off. The result of this modification is that the two by two mutation operator can now only mutate at the boundaries of the shapes being formed, and consequently it should also help reduce the number of small protuberances.

3.5 Results of Improved System

The improved GA for annulus optimisation used the same settings as the basic system for all parameters except that the chromosomal grid was set to 21 voxels high, which is mirrored due to the symmetry used to produce a voxel grid height of 41 voxels. The analysis was permitted to continue for 114 generations and this took approximately twenty-four hours in total. Some of the final population created by the improved GA are shown in Figure 11. This displays three of the twenty individuals and shows a clear improvement in quality over the results generated previously. The small protuberances have been totally eliminated and only a few members of the population contain small holes. The rate at which a valid solution was found is considerably faster than the basic implementation, and once found, the GA continued to improve upon this solution even to the very last pass of this trial.

The annulus shapes produced can be seen to be unusual. It is proposed that the ‘overhangs’ present at the cob and the thinness of the neck are due to the inadequate specification used for the annulus and the method used to penalise constraint violation. Stress constraints were defined for 4 discrete points in the specification which was intended to be used with a parameterised shape description. This specification would be adequate for such a representation. However, with the voxel representation the optimiser was able to remove material with greater flexibility. At an optimal solution one of the stress constraints is just inactive. Removing more material would then increase the stress to above the maximum value. However the GA could improve the fitness value if, by adding material elsewhere, the position of high stress was moved from the point at which the constraint was assessed, as long as the amount of material added was less than that removed. Given that this explanation is correct, the problems do not lie with the voxel representation and could be solved by improving the specification and method of penalising constraint violation.

[Figure 11: Final Annulus Cross-Sections From Improved GA]

After using the FE package to examine the solutions produced by this optimisation, it was possible to confirm that the use of the triangular elements to smooth the boundary worked as expected in reducing the amount of stress in the regions immediately surrounding a step. Figure 12 shows the stress values calculated by the FE package for the voxels surrounding steps in two typical runs and clearly shows how the triangles permit the excess stress to be distributed in a more even pattern. Darker shades indicate higher stress levels in both of these pictures.

[Figure 12: Results without and with Smoothing Triangles]

[Figure 13: The Best Annulus Design From the Final Set of Experiments]

3.6 Conclusions for the Annulus Design Problem

It was found that the use of unmodified operators from the beam design problem was unsuccessful. However when the operators were modified, taking into account knowledge held about the annulus design problem, the results were more successful.

Difficulties were encountered in the direct use of the voxel shape representation as the FE mesh. These were to some extent alleviated by the use of smoothing triangular elements. However, the full decoupling of the primary voxel-based shape description and FE mesh would be desirable in future studies.

Unfortunately, due to the flexibility of the voxel representation in removing and adding material coupled with the GA's ability to exploit the whole search space, it was found that the specification of the problem needed to be more tightly defined as unwanted overhangs were present in the final solution. In response it should be noted that, in the authors' experience, there are often a number of possible problem formulations for a parametric approach, each with differing suitablity to the problem at hand and ability to represent only feasible solutions. Therefore, the above should not be taken to be a severe criticism of the voxel representation – for any approach, a significant amount of experimentation will be required to identify a suitably constrained problem formulation.

The unwanted overhangs aside, a comparison of the mass of the annulus produced by the voxel representation (41 kg), compares well against both the original annulus design (68.6 kg), and that produced by the parametric GA described in (Smith 1995a) which achieved an annulus of mass 40.9 kg. All of these annulus designs satisfied the stress constraints, though given that these designs were evaluated using different FE packages, a fine-grained comparision needs to treated with some caution.

Finally, and rather unfortunately, the voxel GA did not perform as well in regard to time to solution. The parametric GA found its solution in 400 evaluations compared to the 1000 evaluations required by the voxel-based GA – this was felt to be a result of the GA having to search a much larger and less constrained search space when using a voxel representation.

4. Conclusion

Voxels were found to be a viable representation for shape optimisation with an evolutionary algorithm in 2D problems. They have a number of potential advantages over other representations such as parameterised boundary descriptions. Topology is not predefined, domain knowledge is easy to incorporate, geometric constraints can be easily applied, and it is straightforward to convert existing solutions into such a description in order to 'seed' an initial population of shapes.

Experiments were undertaken on two design problems to investigate the effectiveness and scalability of this representation: a simplified beam design and a jet-engine annulus design using finite element analysis. During these experiments a number of difficulties inherent with this representation were addressed, primarily by use of specifically designed genetic algorithm operators which utilised domain knowledge held about the problems tackled. An N-dimensional crossover operator was used which provided linkage between adjacent rows of voxels and thus avoided the slow convergence found with a conventional crossover operator. An operator was designed to remove unwanted holes produced in candidate shapes and to smooth boundary edges.

On the annulus design problem, the direct use of the voxels as the finite element mesh was found to be inadequate, and a convolution mask based solution to this issue was devised. That said, further work in this regard will involve the further decoupling of the voxel representation and mesh.

Furthermore, the flexibility of the voxel representation, along with the GA's exploitation of of a much expanded search space uncovered deficiencies in the specification used for the annulus design problem, leading to unwanted `overhangs' in the solutions obtained. Though the results obtained were roughly equivalent in terms of the mass of annulus produced, they compared poorly in regards to the number of evaluations required to find such a solution.

Finally, it should be noted that GA optimisers can easily be modified to be used as interactive optimisation systems (Tuson et al., 1997). In this case the computer would rely on an engineer's practical experience and knowledge of the problem domain to direct key choices in the optimisation process. Given the diversity of possible shape optimisation problems such flexibility will be required to deal with the constraint handling issue noted above. The lack of initial assumptions in the voxel representation could be seen to be an advantage here as the engineer has, in effect, a *tabula rasa* to start work from, and constraints on the solutions obtained can be expressed directly. Given the amount of experimentation required to find a good problem formulation for both parametric and voxel approaches, such an interactive approach would be highly desirable in any case. Further research into principled methods for allowing the user to interact with such a system is therefore recommended.

Acknowledgements

We would like to thank the Engineering and Physical Sciences Research Council (EPSRC) for partly funding this work via two research studentships (95303677 and 95306458).

References

Baron, P.J. (1997). *A Voxel Representation for Evolutionary Shape Optimisation*, MSc Thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, UK.

Bentley P.J. (1996). *Generic Evolutionary Design of Solid Objects using a Genetic Algorithm*. Ph.D. Thesis, University of Huddersfield, Huddersfield, UK, 1996.

Cartwright, H.M. & Harris, S.P. (1993). Analysis of the distribution of airbourne pollution using genetic algorithms. *Atmospheric Environment*, 27, 1783-1791.

Chapman, C.D., Saitou, K., & Jakiela, M.K. (1994). Genetic Algorithms as an Approach to Configuration and Topology Design. *Transactions of the ASME, Journal of Mechanical Design*, 116, 1005-1012.

Farrell, S.M. (1998). Gamin: a new approach to the inversion of geographical and potential-field data. Ph.D. Thesis (submitted). Monash University, Monash, Austrialia.

Gere, J.M. & Timoshenko, S.P. (1984). *Mechanics of Materials*, 2nd Edition, Brooks/Cole Engineering Division, Monterey, California.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.

Hsu, Y-L. (1994). A review of structural shape optimisation. *Computers in Industry*. 25 (1), 3-13.

Husbands, P., Jeremy, G., Ilhagga, M., & Ives, R. (1996). Two Applications of Genetic Algorithms to Component Design. In *Selected Papers: AISB Workshop on Evolutionary Computing*, (Fogarty, T.C., Ed.), Lecture Notes in Computer Science 1143, pp. 50-61. Springer Verlag, Berlin.

Kirkpatrick, S., Gelatt, C.D. Jr., & Vecchi, M.P. (1983). Optimization by Simulated Annealing. *Science*, 220, 671-680.

Smith, R. (1995a). *A Parameterised Rolls Royce Annulus.* Internal Report No. 21, Manufacturing and Planning Group, Department of Mechanical Engineering, University of Edinburgh, Edinburgh, UK.

Smith, R, (1995b). *A First Investigation into a Voxel Based Shape Representation.* Technical Report, Manufacturing and Planning Group, Department of Mechanical Engineering, University of Edinburgh, Edinburgh, UK.

Tuson, A, Ross, P., & Duncan, T. (1997). On Interactive Neighbourhood Search Schedulers. In *The 16th Workshop of the UK Planning and Scheduling SIG,* 207-212.

Watabe, H. & Okino, N. (1993). A Study on Genetic Shape Design. In *Proceedings of the Fifth International Conference on Genetic Algorithms,* (Forrest, S., Ed.), pp. 445-450. Morgan Kauffman, San Mateo.

Whitley, D. (1989). The GENITOR Algorithm: Why Rank Based Allocation of Reproductive Trials is Best. In *Proceedings of the Third International Conference on Genetic Algorithms*, (Schaffer, J. D., Ed.), pp. 116-121, Morgan Kauffman, San Mateo.

Figure Legends and Figures

The figures are provided as camera-ready copy with each of the figures presented in the order given below:

Figure 1: Typical End of Run Results from the Naïve GA

Figure 2: The Smoothing Operator

Figure 3: Typical End-of-Run Results with the Smoothing Operator

Figure 4: The Effectiveness of Various Crossover Operators

Figure 5: The Effectiveness of Various Mutation Operators

Figure 6: Typical End-of-Run Results for the Complete System

Figure 7: Performance Comparison Between the Naive and Final GAs

Figure 8: Annulus Axisymmetric Cross-Section

Figure 9: Results of the Basic Annulus Optimisation After 75 Generations

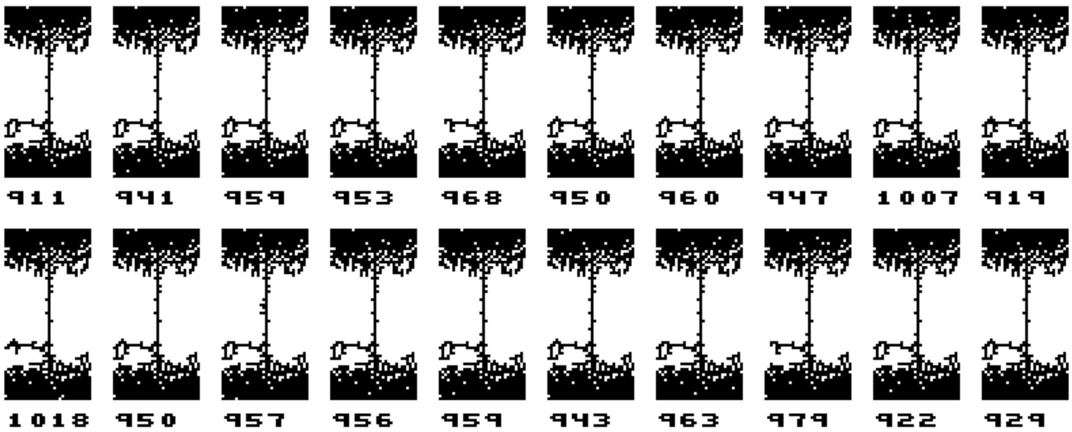
Figure 10: Convolution Masks for Triangle Insertion Process

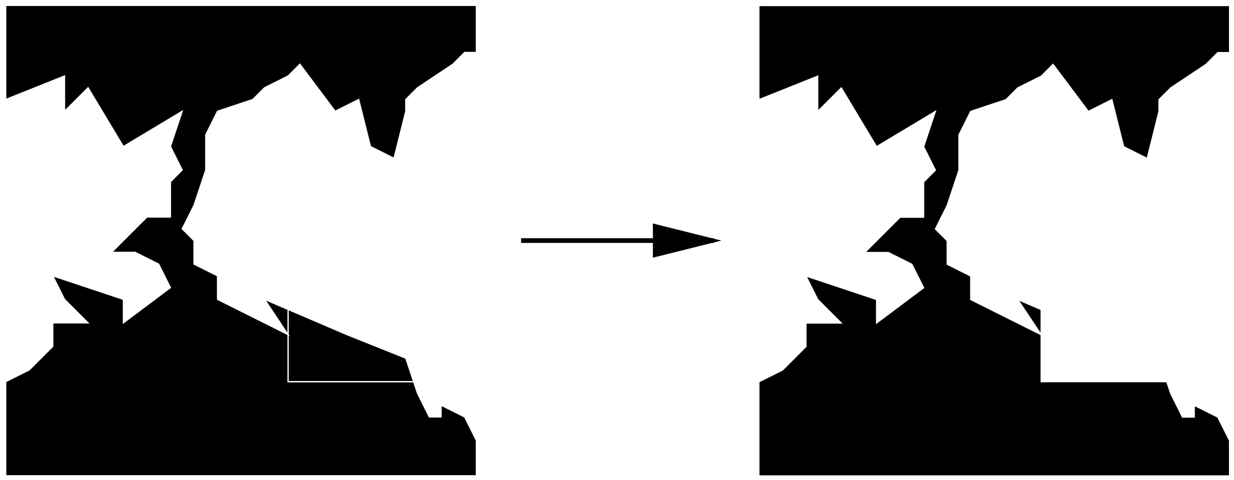
Figure 11: Final Annulus Cross-Sections From Improved GA

Figure 12: Results without and with Smoothing Triangles

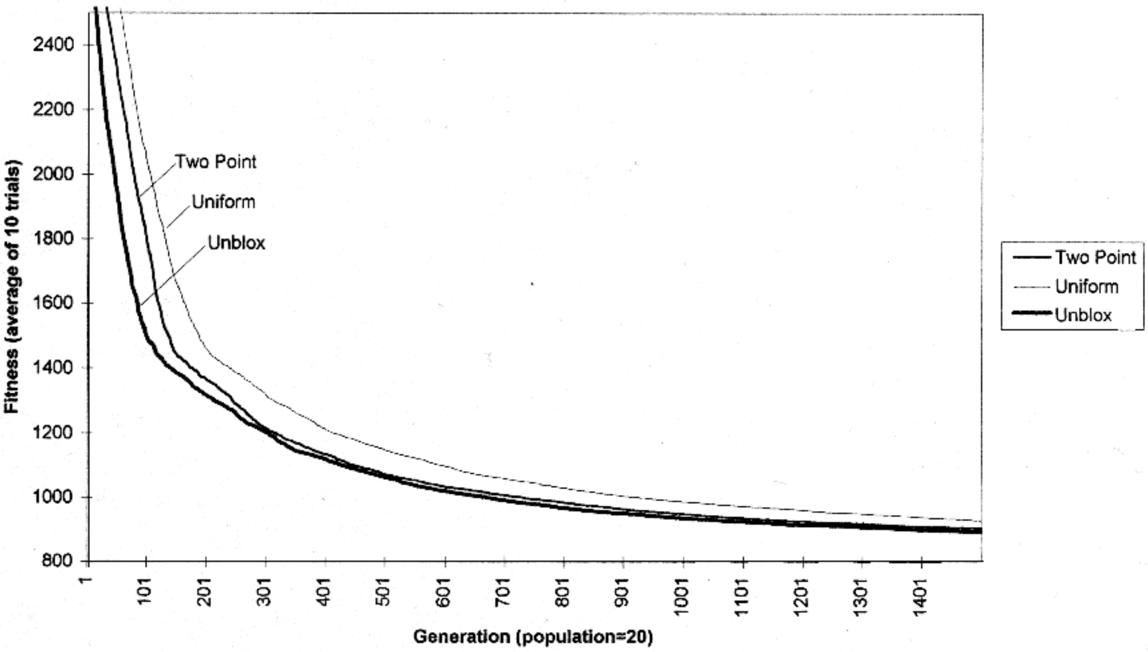
Figure 13: The Best Annulus Design From the Final Set of Experiments

Electronic copies of the original diagrams are available from the corresponding author.

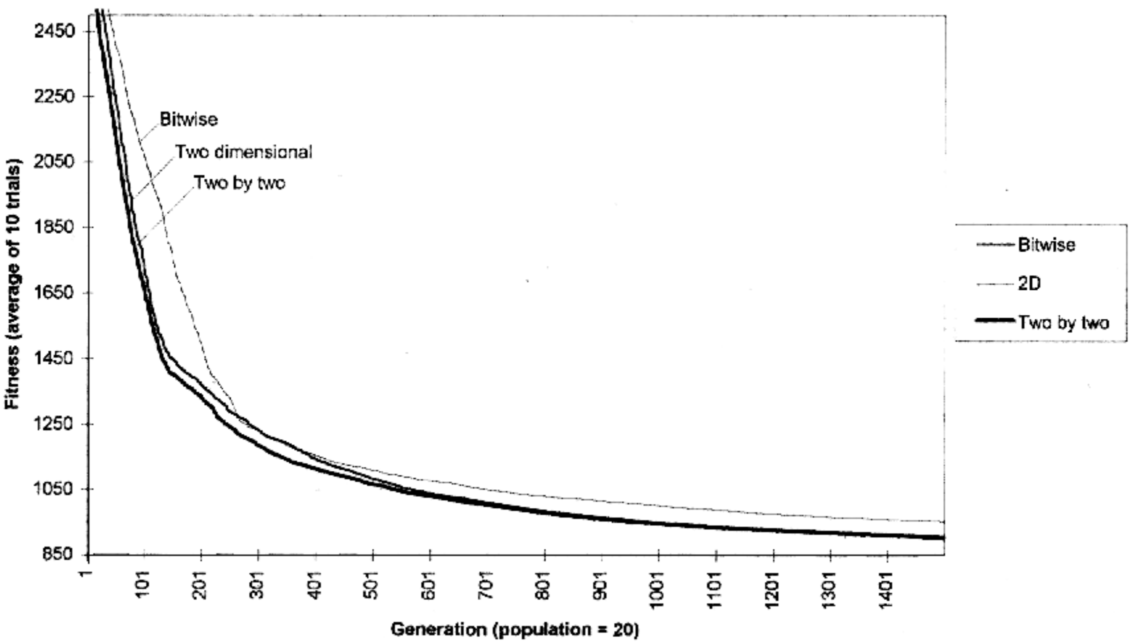

911 941 959 953 968 950 960 947 1007 919
1018 950 957 956 959 943 963 979 922 929



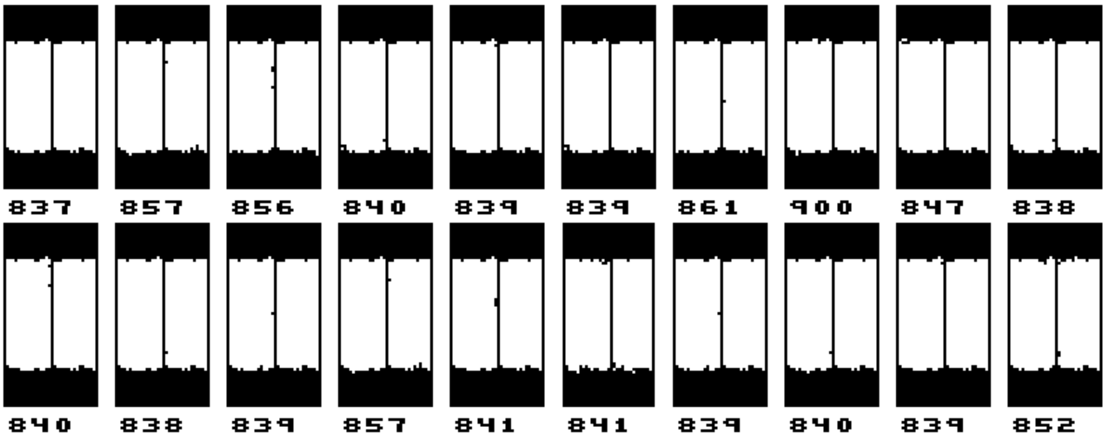

855 921 864 860 861 881 858 856 919 857
934 857 855 856 2361 855 857 3447 1321 991

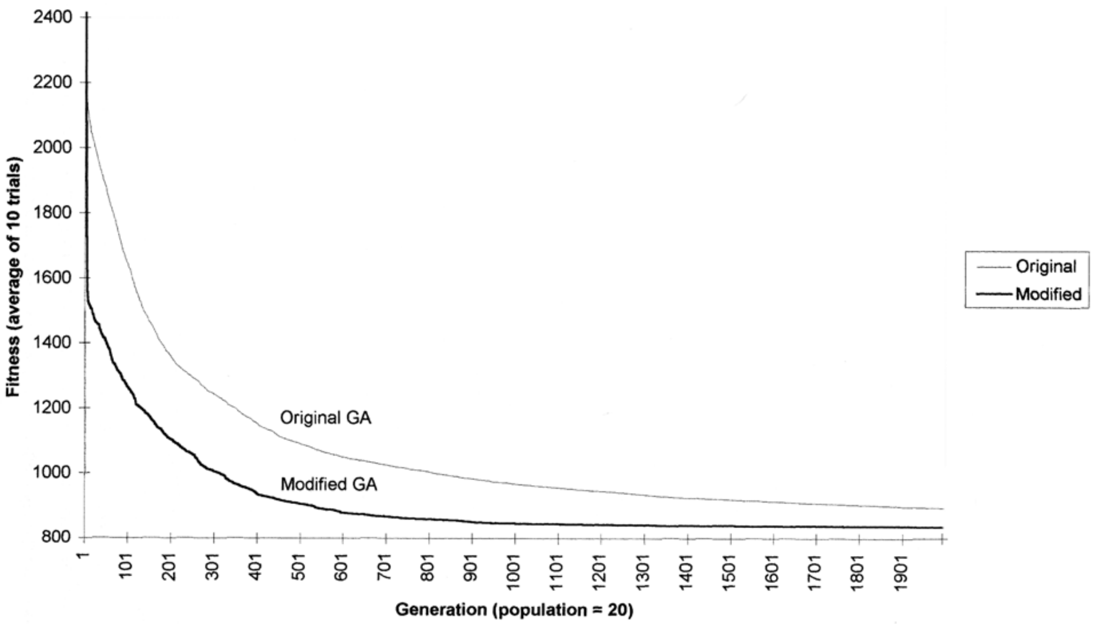


2400
2200
2000
1800
1600
1400
1200
1000
800
Fitness (average of 10 trials)
Two Point
Uniform
Unblox
1
101
201
301
401
501
601
701
801
901
1001
1101
1201
1301
1401
Generation (population=20)
Two Point
Uniform
Unblox

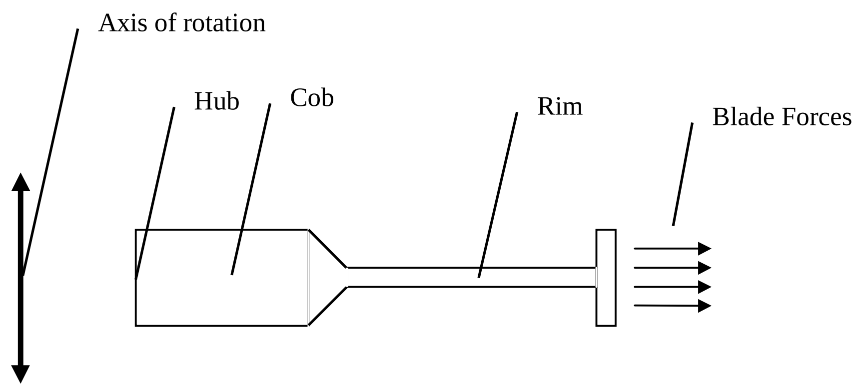


Fitness (average of 10 trials)
2450
2250
2050
1850
1650
1450
1250
1050
850
Bitwise
Two dimensional
Two by two
1
101
201
301
401
501
601
701
801
901
1001
1101
1201
1301
1401
Generation (population = 20)
Bitwise
2D
Two by two


837
857
856
840
839
839
861
900
847
838
840
838
839
857
841
841
839
840
839
852

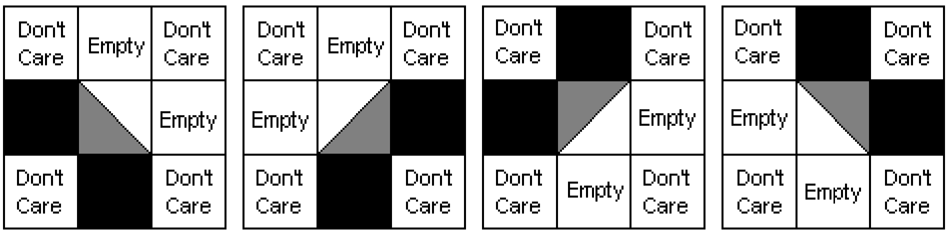


2400
2200
2000
1800
1600
1400
1200
1000
800
Fitness (average of 10 trials)
Original GA
Modified GA
1
101
201
301
401
501
601
701
801
901
1001
1101
1201
1301
1401
1501
1601
1701
1801
1901
Generation (population = 20)
Original
Modified


Axis of rotation
Hub
Cob
Rim
Blade Forces







Don't Care
Empty
Don't Care
Empty
Don't Care
Don't Care
Don't Care
Empty
Don't Care
Empty
Don't Care
Don't Care
Don't Care
Don't Care
Empty
Don't Care
Empty
Don't Care
Don't Care
Don't Care
Empty
Don't Care
Empty
Don't Care