A Voxel Based Approach To Evolutionary Shape Optimisation

Peter Baron[1], Robert Fisher[1], Andrew Sherlock[2], Frank Mill[2], Andrew Tuson[1]

[1]Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN. Email: {peterba,rbf,andrewt}@dai.ed.ac.uk

[2]Manufacturing Planning Group, Department of Mechanical Engineering, University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh, EH9 3JL. Email: A.Sherlock@ed.ac.uk, F.Mill@ed.ac.uk

Abstract. Shape optimisation is a hard problem from the field of Mechanical Engineering with the potential for significant cost savings if successfully performed. In the past, evolutionary optimisation approaches have proved successful. In those studies, a form of the shape was assumed, and its parameters optimised. An alternative is a voxel (N-dimensional pixel) based representation, which makes no such assumptions about the form of the solution, and allows the user to add domain knowledge if desired. This paper outlines a preliminary investigation into this approach and shows that the objections to this approach in the literature can be overcome if care is taken over the design of the operators.

1 Introduction

Shape optimisation within constraints is a hard problem from the field of Mechanical Engineering. The objective is to design a shape which best satisfies some predetermined goal whilst at the same time maintaining some property of the shape within a constraint, or perhaps even a set of constraints.

Previous work has applied genetic algorithms [5] to shape optimisation problems with encouraging results. Using engineering software packages to evaluate the suitability of a given design, successful shapes have been evolved — examples include [8,6]. However, previous work on evolutionary shape optimisation has primarily been concentrated around parametric representations of structural design and shape optimisation problems.

The investigation detailed here evaluates the use of a voxel (N-dimensional pixel) based representation instead, within which the shapes being optimised are represented as a series of binary 0's and 1's. This approach has the advantage that it can describe any topology, but it also has its drawbacks. The question is: are the drawbacks sufficient to prevent this approach being useful?

This paper will outline the case for such an approach, and describes an implementation of a basic genetic algorithm, and an extended version. A summary of the results obtained for a simple shape optimisation problem is then given, from which we conclude that the potential problems of the voxel representation that have been suggested in the literature can be circumvented if suitable operators are designed and used.

2 The Case For A Voxel-Based Approach

Traditionally, the representational approach used in structural design has been parametric — a form of the solution, for example as a set of splines, is derived or assumed and the parameters associated with this form are optimised. However, parametric approaches do make strong assumptions about the form of the solution, which may not necessarily correspond to the optimal solution. In addition, it is not always straightforward to devise a good parametric form for some problems.

Taking beam design as an example, holes in the shape of the beam are entirely possible and even probable if some of the mass of the beam is occupying a low-stress area — a parametric representation can only create holes where the user is expecting them to be required and has defined the appropriate parameters.

A report by [7] has discussed the possible use of voxels as a representational approach. As this encoding can describe any shape, it can deal with the situation above, and makes no assumptions about the form of the final solution. Furthermore, a voxel based representation, by virtue of its directness of representation allows domain knowledge to be easily added, and to the level felt appropriate by the designer. Two examples will help illustrate this.

First, areas of the voxel representation can be fixed to be permanently on or off; this allows the designer to prohibit material from being placed in locations where it is not desired. A second example lies with the ease in which existing designs can be utilised by the system — all that is required is for the initial design to be digitised and the bitmap used to initialise the population of the genetic algorithm.

However, [8] argues that using a binary voxel representation leads to the following problems:

- The long length of the chromosomes ($>$ 1000 bits).
- The formation of small holes in the shape.
- No guarantee that the final shape produced will be smooth.
- Even if the parents represent a valid shape, the children will not necessarily be valid.

This paper will evaluate whether these objections constitute a problem in practice.

3 The Optimisation of the Cross Section of a Beam

One of the problems encountered in shape optimisation, no matter what approach is taken, is interfacing the optimiser with a suitable evaluation package. For example, in wing optimisation, the wing shape has to be smooth, else the

CFD (Computational Fluid Dynamics) package will act strangely: either returning negative drag, or resulting in the program crashing. Problems involving Finite Element Analysis (FEA) evaluation packages are somewhat better behaved, but interfacing is still not trivial, and the calculations do take some time.

The shape optimisation of a beam cross-section is the problem considered in this study. Evaluation of the candidate cross-sections was made using bending theory for symmetrical beams, considering only normal stresses [4]. This is a greatly oversimplified model, but sufficient to test whether the potential problems with a voxel representation outlined above do pose a problem in practice.

Each candidate solution can thus be represented as a 2-D grid of voxels, with the optimisation objective being to minimise the mass m of the beam whilst ensuring that at all points (ie. at all voxels) in the beam the normal stress does not exceed a maximum stress (σ_{max}), which is a constant that is determined by the material to be used. This maximum stress constraint is given by Equation (1):

$$\sigma_{max} \geq \frac{-My}{I} \; for \; all \; voxels \tag{1}$$

where M is the bending moment, y the distance of the voxel from the neutral axis, and I is the second moment of area of the cross-section. The neutral axis of a shape is defined as a line which passes through the centroid of mass of the shape. As the voxels are of uniform size and density, the centroid can be found by taking the average of the positions of all the occupied voxels; also, for a symmetric beam the neutral axis would be horizontal.

The bending moment M is a constant determined by the loading on the beam. The mass m of the beam is proportional to the area of the cross-section and hence the number of voxels turned on. The second moment of area is given by Equation (2):

$$I = \sum_{i=0}^{all \; \text{``}on\text{''} \; voxels} y_i^2 \; dA \tag{2}$$

where y_i is the distance of the ith voxel from the neutral axis and dA is the area of a voxel (as we are dealing with a cross-section).

The optimum cross-section for a beam using this evaluation can be deduced to being a flange at top and bottom of the design domain. For the experiments described here, the following values of the constants were used: $M = 2 \times 10^6$ Nm, $\sigma_{max} = 100$ MPa, and a beam of dimensions 320×640mm was considered.

In practice, this would correspond to an I-beam, but that also requires a web to connect the two plates of the beam together. In a full calculation with shear stresses, the web would arise 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 (by a 4 connect rule) to a seed pixel in the centre top edge of the beam. In addition, a straight web was enforced 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.

4 A Basic Genetic Algorithm Implementation

This study builds upon the following basic implementation of a genetic algorithm. The encoding digitises the shape as a 32×64 grid and represents this as a 1-dimensional string of 2048 bits. Standard two-point crossover and bit-flip mutation were used to operate on this encoding. After the application of each operator, pixels that were not connected to the seed pixel were set to zero. An unstructured, generational population model of size 20 was used, with rank-based selection with selection pressure 1.7.

Strings are initialised randomly to a set 'density' percentage, the higher this value, the more voxels are initially turned 'on'. For this study, this was set to 70%. All initial population strings must pass the validity checks used by the evaluation function: the number of active voxels must be non-zero; the second moment of area must be greater than 1×10^{-12} (goes to 0 if insufficient active voxels); and the fitness must be greater than 0.0001.

4.1 The Fitness Function

The fitness function was designed to minimise the area of the beam (number of active voxels) within the maximum allowed stress, with a penalty value being applied to any solution which broke that constraint. A small additional factor, $\frac{1}{S}$, was included in the fitness calculation based upon the maximum stress point in the beam. This had the effect of causing any valid solutions to continue to evolve towards better solutions (in this case a beam which minimises area and minimises the maximum amount of stress present). The fitness, F, is given by Equation (3) below:

$$F = \frac{1}{V - \frac{1}{S} + k \times max\{(S - \sigma_{max}), 0\}} \tag{3}$$

where V is the number of active voxels (ie. beam mass), S the highest stress (calculated using Equations (1) and (2)) felt by any of the pixels, σ_{max} the maximum allowed stress, and k a constant which can be adjusted to vary the weight of the penalty associated with the maximum stress constraint (in this study it was set to $k = 5.0 \times 10^{-5}$).

4.2 Results Obtained

The basic genetic algorithm was found to give disappointing results. When allowed to run to convergence (over 2000 generations), the shapes produced, though recognisable as approaching an I-beam shape, were highly irregular and possessed small holes. Figure 1 illustrates this by showing the 4 best solutions from ten genetic algorithm runs. Therefore it appears, at least for a simple genetic algorithm implementation, that the potential problems described in Section 2 do manifest themselves.

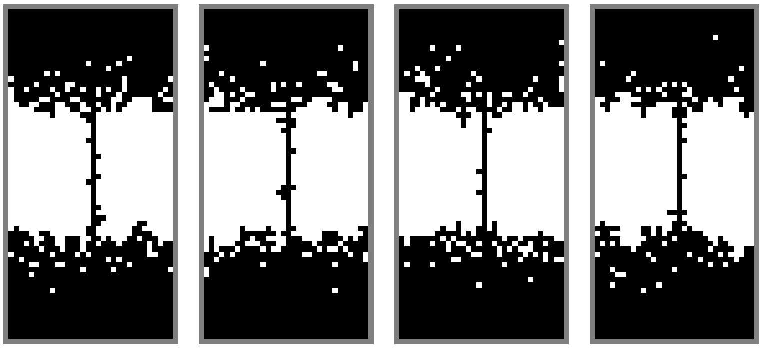

Fig. 1. Typical End-of-Run Results Obtained by the Basic GA

5 An Improved Genetic Algorithm Implementation

The performance of the basic genetic algorithm was disappointing, however, many successful applications of the genetic algorithm use domain specific operators [3]. Knowing the nature of the problems that arise with this approach, would it be possible to design operators to overcome them? With this end in mind, the basic operators were replaced in order to improve the following:

- Crossover effectiveness.
- Removal of holes and isolated pixels.
- Removal of rough edges.

All of the operators also address one weakness in the basic genetic algorithm implementation: the encoding of a 2-dimensional problem as a one-dimensional string. The modifications made will now be described in turn, and results of experiments to evaluate their effectiveness summarised.

5.1 A Mutation Operator for Smoothing

A mutation operator for smoothing was then devised. An area of x/y size ranging from 2 pixels to 1/4 of the dimensions of grid was randomly selected. The most common value for the pixels in the area selected was then found, and then written to all of the pixels in that area (Figure 2).

Examination of the results obtained confirmed that this can remove isolated pixels with great success. With this as the only mutation operator, the shape was optimised to two near perfect horizontal bars at the vertical extremities (an I-beam) after 1000 generations.

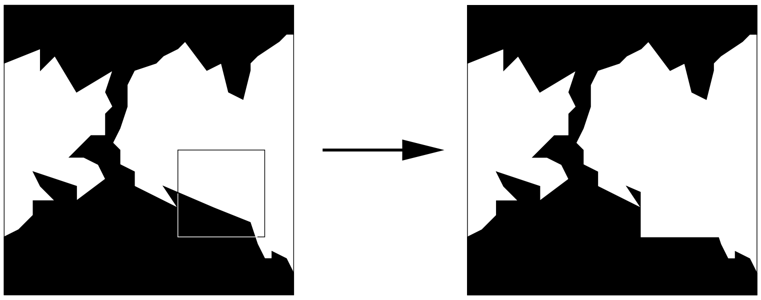

Fig. 2. The Smoothing Mutation Operator

5.2 Two-Dimensional Crossover

Another problem encountered in representing what is a 2-dimensional optimisation problem, as a one-dimensional chromosome, arises with *linkage*. In a 1-D encoding, voxels that correspond to spatially close points on the actual shape can be far apart on the string. Therefore, use of crossover can more easily disrupt building blocks such as one part of the shape being of high fitness, because the bits that correspond to it are spread across the string.

Such a situation has been encountered before in the use of a genetic algorithm to solve the source apportionment problem [1,2]. That investigation found that representing the problem as a 2-D matrix, and devising a crossover operator (UNBLOX) that swapped a 2-dimensional section between solutions lead to improved performance (Figure 3).

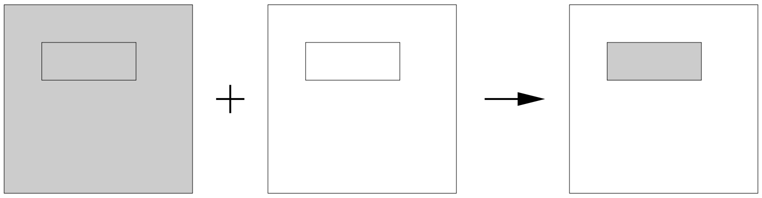

Fig. 3. The UNBLOX Crossover Operator

When UNBLOX was implemented for this problem it was able to produce a recognisable I-beam after 500 generations — a large improvement in performance over the one-dimensional two-point crossover operator.

5.3 2×2 Area Mutation Operator

This operator acts on a 2×2 area of the chromosome array, and only modifies the contents if at least one voxel in the chosen area is turned on and one other voxel is turned off. This so so that this mutation operator would work well on surfaces. If so, standard bit-flip mutation is applied to each voxel in the area. This operator was devised for three reasons: first, the belief that as we are tackling a 2D problem, the operators should reflect this; second, a 2×2 area was thought sufficient to eliminate many of the loose pixel/wobbly line problems found with the basic genetic algorithm; third, applying the operator only where at least one pixel is present gives a good probability of a worthwhile modification being found. This operator can also be thought of as having Lamarkian characteristics.

Use of this operator greatly increased the rate at which excess material is chopped away, whilst the ability of the genetic algorithm to continue to find improvements after convergence was also improved. In addition, the final form of the I-beam was found to be much cleaner than that obtained by the basic genetic algorithm.

5.4 The Final System

After the formative evaluations of each of the new operators above, it was necessary to see if these operators would work effectively in combination. Therefore, the genetic algorithm was run with the following settings: $p(UNBLOX) = 0.3$, $p(2 \times 2\, mutation) = 0.125$, $p(Smoothing\, mutation) = 0.125$ (all probabilities are per-string); where each operator is applied sequentially to the string with these probabilities. The probability of the 2×2 mutation operator was increased by 0.0005 per generation to a maximum of 0.4 as this was found to improve the quality of the results obtained slightly. All of the other genetic algorithm settings were left unchanged from the basic genetic algorithm.

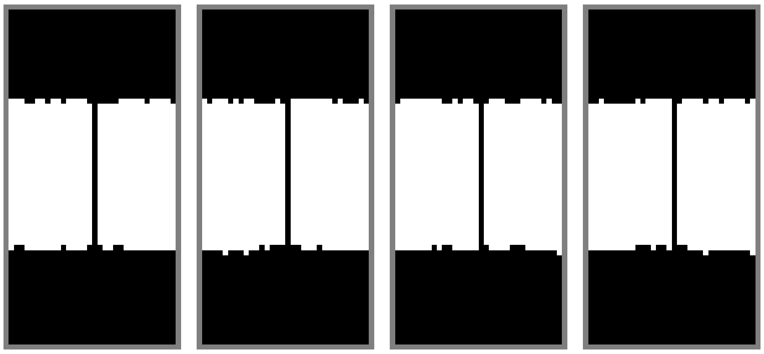

Fig. 4. Typical End-of-Run Results Obtained by the Improved GA

Figure 4 shows some typical end-of-run results, which are near-perfect I-beams, with no holes — a noticeable improvement over the basic genetic algorithm (Figure 1).

Plots of the fitnesses against generation for the basic and improved genetic algorithms are given in Figure 5 (these figures are an average over 10 runs). As can readily be seen, the new operators have dramatically improved the performance of the genetic algorithm, finding a better quality solution more quickly than the basic genetic algorithm, thus vindicating our approach.

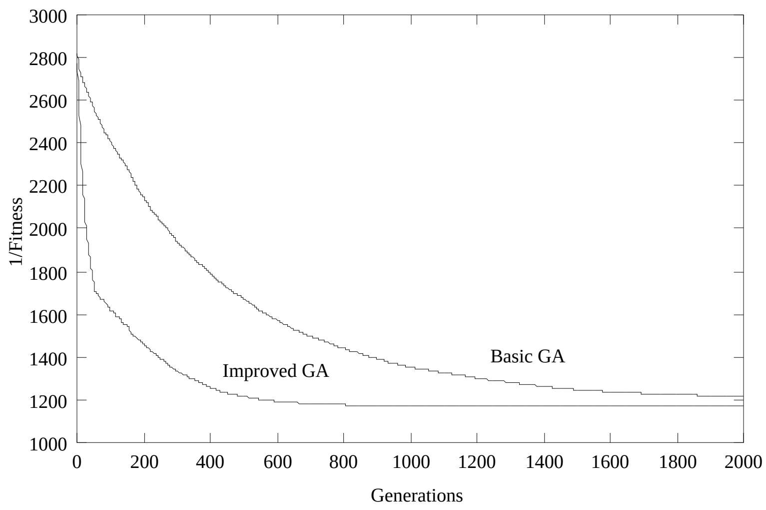

Fig. 5. Fitness Plots for the Basic and Improved GAs

6 Conclusion

This paper has highlighted the possible utility of a voxel-based representation for shape optimisation and has shown, contrary to previous arguments in the literature, that this approach is suitable for evolutionary shape optimisation when used in conjunction with suitably designed operators. The length of the strings, and the validity arguments were found not to be problems in practice — the augmented genetic algorithm was found to be able to discard quickly poor or invalid solutions in favour of good ones. It would appear that, on the basis of these results, a voxel-based approach to shape optimisation would be viable for real problems, especially when you consider that the population has been randomly initialised — which would not often be the case when tackling a real problem. Work applying this approach to real problems, with a view to a comparision with parametric methods, is currently underway.

7 Acknowledgements

Thanks to the Engineering and Physical Sciences Research Council (EPSRC) for their support of Andrew Sherlock and Andrew Tuson via studentships with references 95303677 and 95306458.

References

1. H. M. Cartwright and S. P. Harris. Analysis of the distribution of airborne pollution using genetic algorithms. *Atmospheric Environment*, 27:1783–1791, 1993.
2. H. M. Cartwright and S. P. Harris. The Application of the Genetic Algorithm to Two-Dimensional Strings: The Source Apportionment Problem. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Mateo: Morgan Kaufmann, 1993.
3. L. Davis, editor. *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.
4. J. M. Gere and S. P. Timoshenko. *Mechanics of Materials 2e*. Brooks/Cole Engineering, 1984.
5. John H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.
6. P. Husbands, G. Jeremy, M. Ilhagga, and R. Ives. Two Applications of Genetic Algorithms to Component Design. In Terry C. Fogarty, editor, *Selected Papers: AISB Workshop on Evolutionary Computing, Lecture Notes in Computer Science No 1143*, pages 50–61. Springer Verlag, 1996.
7. R. Smith. A First Investigation into a Voxel Based Shape Representation . Technical report, Manufacturing Planning Group, Department of Mechanical Engineering, University of Edinburgh, 1995.
8. H. Watabe and N. Okino. A Study on Genetic Shape Design. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 445–450. San Mateo: Morgan Kaufmann, 1993.