Determining the Optimal Cross Section of Beams

D.R. Griffiths & J.C. Miles

Cardiff School of Engineering, Cardiff University, P.O.Box 925, Cardiff CF24 OYF, UK

MilesJC@cf.ac.uk

FAX 00442920 874597, Tel 00442920 875694

Abstract

Constrained shape discovery and optimisation are difficult engineering problems. Shape discovery deals with evolving randomly generated solutions into useful intermediate solutions. These intermediate solutions are then optimised to suite their environment. In this paper a Genetic Algorithm (GA) is applied to the problem of finding the optimum cross section of a beam, subject to various loading conditions. Previous work using GAs for this problem has relied heavily on heuristics and domain knowledge that operates directly on the genotype to guide the search. It is sound engineering practice to utilise all available information to reduce design and evaluation times, and encourage the formulation of useful solutions. However, domain design knowledge is not always available. This research attempts to explore the efficiency and effectiveness of a GA, when applied to a difficult design task, without being unnecessarily constrained by preconceptions of how to solve the task. Heavy guidance of a GA potentially stifles innovation, can only be applied to situations where the correct answer is known and limits the generic abilities of the search system. This research demonstrates the ability of the GA to evolve good, near optimal solutions without direct guidance. Performing an unbiased search, using only the evolutionary process to search for good solutions, allows a GA to be applied with a high degree of confidence to situations where a priori knowledge of the optimum solution is unavailable. Advanced 2-dimensional genetic operators, in conjunction with a suitably designed fitness function, allow a productive evolutionary search. The initial test case is the evolution of an optimal I-beam cross-section, subject to several load cases, starting with an initial random population. It is shown that the methods developed lead to consistently good solutions, despite the complexity of the process.

Key Words: Genetic Algorithm, heuristics, beam cross-sections, domain knowledge

Introduction

Although some common structural materials, such as steel cannot easily be formed into bespoke, specialist shapes, there are many materials such as aluminium and plastics which can, within reason, be easily and cheaply formed into any desired profile. To date structural requirements have tended to be met using standard shapes such as I beams and columns or tubular sections. In part, this is because the determination of the optimum cross section for a given loading pattern is difficult and so designers are not able to specify exactly what they require.

This paper describes the application of a Genetic Algorithm (GA) to a realistically constrained, structurally based, Shape Discovery and Optimisation problem. Structural optimisation methods typically utilise a base shape and a predefined set of parameters. An iterative process is applied to determine the optimal combination of parameters, resulting in evolved shapes which out-perform the base shape. Shape discovery is the process of evolving feasible shapes without the use of a base shape. Shape Discovery using a GA deals with an initial population of random shapes that are then evolved into feasible and, in cases where the method is well formed, optimal shapes. The system presented here encompasses the ability to generate useful intermediate shapes, through shape discovery and also the ability to optimise these generated shapes. Using computers to generate the form of designs rather than a collection of predefined high level concepts has the advantage of giving greater freedom to the computer based method (Bentley, [1]). A Voxel based representation has been utilised in this work as it can represent any two-dimensional shape, with curves being approximated by a series of steps.

An important feature of the work described in this paper is the evolutionary development of optimal shapes without using heuristics to directly modify the form of the solution. The omission of such heuristics requires the system to execute a thorough search, evaluating a large range of potential solutions. Hence the GA used is designed to be able to produce two-dimensional solutions for any loading condition, and not the solutions to specific problems where the optimum result is known in advance as in the work of Baron [2].

A difficult problem has been used as the test bed for determining the effectiveness of the GA and associated techniques that have been developed during this work. The system is required to evolve optimal beam cross-sections for given loading cases. To establish the technique, problems with known solutions have been utilised. Care is taken to ensure that all final solutions satisfy the applied constraints. In previous work, GAs for determining optimal beam cross sections have relied heavily on heuristics which operate on the genotype to achieve optimal solutions (Baron, [2]). This creates a situation where the search ability of the GA is limited to the shapes to which the heuristics apply. The aim of this research is to prove that a GA based system can find optimal solutions without the use of heuristics to directly modify inappropriate solutions and therefore can potentially be applied with confidence to problems where the optimal solution is unknown.

Shape Optimisation

Shape optimisation is defined as the process of reducing a base shape to a series of parameters and altering these parameters until a new shape is created that performs better than the base shape (Rozvany, [3]). Explicitly defined parameters are selected, thus defining what elements of an existing solution are to be optimised. This is usually an iterative process where changes are continuously made to explore combinations of variables to discover the optimal combination.

Non-generic systems, tailored towards solving a very narrow range of optimisation problems, have been applied to shape optimisation tasks with varying degrees of success. Numerous techniques are available for the optimisation of existing designs. However, the difficulty level of the optimisation task and the size of the search space limit options. This research utilises a search space of 2^{2048} ! (based on the representation method described later) and the method is required to find the optimal solution without the use of a base solution. Except for very basic problems, an exhaustive search is not possible and a random search is also unlikely to find the global optimum. Even more advanced search methods such as stochastic hill-climbing (SH), which searches through local optima looking for the global optimum, are likely to find the global optimum evasive in a search space with many local optima. Juels [4], compared GA and SH methods on a series of optimisation problems, with the GA being able to consistently outperform SH, although in making such comparisons one has to be aware of the work of Wolpert and Macready [5].

Simulated Annealing (SA) is a combinatorial optimisation technique that is based on the annealing of solids (Ruthenbar, [6]). This technique is capable of escaping local optima in the search space, and can therefore frequently find good solutions to many optimisation problems. Shea and Cagan [7] present an interesting application of SA for determining the structural layout of geodesic domes. Although their results do not appear to be optimal, their work shows that the technique can be used to generate feasible, alternative configurations.

Recently, the use of GAs and other evolutionary/adaptive based optimisation systems has been favoured. GAs with seeded initial populations (initial populations already containing good solutions) have been proven to work efficiently and effectively on a wide variety of shape optimisation problems. For example, Azid & Kwan [8] use a GA to optimise the shape of trusses. With its ability to rapidly converge on areas of high fitness within the search space, the GA can provide valuable information to the human designer.

Rasheed [9] used a GA (with an initial population seeded with basic aircraft shapes) to optimise a supersonic transport aircraft design by varying twelve conceptual design parameters. To allow the evolution of optimised results, a *Diversity Maintenance Module* was required to ensure genetic diversity could be maintained, and avoid converging on local optima. Additionally, a *Screening Module* was necessary to reduce computational times to within acceptable limits. The *screening module* performed a preliminary evaluation of solutions, eliminating those likely to perform poorly. The computational expense of the full evaluation module allowed only a limited number of full evaluations to be performed. Robinson [10] applied a GA to the optimisation of satellite boom shapes with the aim of reducing vibrations. To accurately assess the designs considered by the GA, very significant computations were required, even when using a highly tuned and customised code to carry out these calculations.

These examples demonstrate both the tendency of researchers to restrict and/or heavily guide the GA's search and the effect of computational limitations. Although the application of heuristics and domain knowledge can help reduce computational time, the search will tend to be restricted to a predefined portion of the overall search space. Restricting the random nature of the GA can potentially stifle the innovative process by limiting the range of solutions that are evaluated.

Cvetkovi´c and Parmee [24], explore established methods for optimising multi-objective functions whilst addressing the problem of preliminary design. Eight input (e.g. cross-wing plan area, wing aspect ratio, wing taper ratio) and 9 output (e.g. landing speed, ferry range, take off distance) parameters are supplied to the system. Input parameters are manipulated for optimal output. Their system, although capable of optimising the multi-objective problem, does not produce an actual design. The end user is supplied with parameter values that satisfy multiple objectives but not a complete buildable design shape.

Sisk [11] developed a decision support system (DSS), based on a GA to assist members of a design team at the conceptual stage of the creation of multi-storey office building. The system aims to augment the decision-making process at the conceptual stage by introducing the designer to a range of different ideas and solutions, hence expanding the search of the problem space. Extensive use is made of rules-of-thumb that reflects the designer's experience, and heuristics based on fundamental theoretical and experimental knowledge. Sisk's system works on predefined parameters and constraints and then translates these into a usable design for the end user. This bridges the gap between the evolutionary algorithm's manipulation of a parameter list and the requirement to provide the user with an actual buildable design.

Although many GA systems are tailored directly towards a very narrow range of applications, the potential generic qualities of GAs have also been demonstrated. Gen and Kim [12] applied seven GA-based approaches to reliability design problems. In addition to outperforming alternate search methods, the GA was easily used to solve various reliability design problems, without any direct interference with the search process.

Shape Discovery

Shape discovery is the process of finding an optimal shape without the use of a base shape as a starting point. In such a situation, shape discovery is utilised to determine some rough initial shapes and sometimes (although not always) to optimise them. Shape discovery is essentially the process of evolving random shapes into something feasible and then optimising those shapes to suit their given environment. Typically, such systems are free to evolve any form capable of being represented and the evolution of such forms may well result in the emergence of implicit design concepts (Harvey et al, [13]). Using computers to generate the form of design directly often involves just the preliminary stages of design, with an emphasis on the generation of novelty and not the production of globally optimal solutions (Bentley, [1]). This research expands upon this notion, using the computer to generate useful solutions, and subsequently to optimise them.

To evolve random shapes into something useful, the shape discovery system must have control of or be able to influence the values of all parameters and variables. As the form of the optimal shape may be entirely unknown, it is not possible to devise heuristics, which operate directly on the genotype, to ensure the production of good solutions. However, the chosen system still requires complete control over the evolved shapes. This can be achieved through a robust system.

Due to its potential industrial importance, shape discovery has been an area of much interest with greatly varying approaches and results. As search tools, GAs are able to rapidly locate areas of high fitness (feasible solutions) within relatively large search spaces (Parmee & Bonham, [14]). When dealing with shape discovery, the search spaces involved are typically large with many local optima.

Bentley & Wakefield [15] utilised a GA with a primitive building block representation to form a system of 'clipped and stretched cuboids'. The system formed a spatial partitioning representation that evolves designs through modifying combinations of cuboid shapes. Although the system is capable of evolving complex 3-dimensional shapes, the shapes are at best adequate for their purpose and are not optimal. Solutions that in the real world would perform poorly, and contain significant fragmentation cannot be accepted in structural engineering where poor performance costs money or poses the risk of collapse. It is a primary aim of this research that all final solutions are of a high quality.

Even when the initial population is randomly generated, it is possible to bias the search by introducing properties by means of the shape representation used. Bentley's [1] approach to the 3-dimensional shape discovery of tables starts with initial shapes consisting of various sized cuboids. These initial cuboids already contain flat surfaces and are stable when free-standing, both essential attributes for a feasible table.

Baron [2] devised a GA system for the discovery of optimal beam cross-sections but found that the evolved cross-sections invariably contained jagged edges and fragmentation (i.e. holes), leading to very poor solutions. A smoothing heuristic was developed to remove jagged edges from the evolved solutions. This operator selects a randomly sized rectangular section of an evolved cross-section solution (up to ¼ of the size of the shape), and then fills in or entirely removes material from this section. Over sufficient generations this leads to smooth surfaces. The shapes are directly modified, based on domain knowledge of known good solutions. Such drastic modifications of the evolved designs create good solutions but heavily influences the search. The method only works because the solution to the problem is known and hence the heuristic has been devised to steer the result towards the desired answer.

To be an innovative designer it is necessary to divorce the mind from the practicalities of manufacturing techniques, the life of the product, and particular types of material. The sole interest is in meeting the functional requirements of the design (Hawkes et al, [16]). Utilising domain knowledge to direct a search towards known good solutions is an inadequate substitution for a suitable fitness function and genetic operators. Devising a GA system that works effectively without heuristics which operate directly on the genotype increases its generic capabilities, allowing an individual system to be applied to a greater variety of applications and especially those applications where the answer is not known in advance.

Chapman et al [17] applied a GA to the topological optimisation of highly-discretized, cantilevered plates to determine if a genetic algorithm could find complex, non-symmetric topologies. Utilising a representation system similar to the voxel system used in this research, solutions are represented through a binary string. After a chromosome is mapped into the design domain the resulting "material" is analysed for connectivity. All material in the design domain which is not connected (whether directly or indirectly via other elements) to a "seed" element is considered to be to void. This "connectivity analysis" guarantees that all topologies are stable. When a disconnected element is switched to void or a seed element is switched to material, the corresponding chromosome allele is not modified—only the design domain element is changed. Feasibility checks, based on domain knowledge, directly modify the form of a solution, in an attempt to ensure useful solutions are evolved.

Chapman et al [17] also highlight the computational cost associated with the use of a GA. In their experiments, thousands of function evaluations (*i.e.*, finite element runs) were required to obtain a solution. Because each finite element run is computationally expensive, structures subject to GA based topological optimisation using this approach must currently be of limited size and complexity. The system presented in this research utilises custom built evaluation modules, designed specifically for use with a voxel representation. Each evaluation module strikes a balance between the need to accurately evaluate given solutions, and the computational expense of doing so.

Pablo, et al. [18], evolved buildable designs using miniature plastic bricks as modular components. Instead of incorporating an expert system of engineering knowledge into the program, which would result in more familiar structures, they combined an evolutionary algorithm with a model of the physical reality and a purely utilitarian fitness function, providing measures of feasibility and functionality. The evolutionary process runs in an environment that was not unnecessarily constrained by preconceptions of how to solve the problem. Although the system does not generate optimal solutions, it demonstrates an ability to generate novel and 'fit for purpose' solutions, without the need for guidance.

Liang et al, [25] present a methodology for developing topologies for the minimum-weight design of continuum structures within stress constraints. The system applies Evolutionary Structural Optimisation (ESO) (Xie and Steven, [20]), which optimises a topology by gradually removing inefficient material. Optimum solutions can be achieved by allocating a Performance Index (PI) to a solution, and recording the PI history. A PI is a dimensionless number that measures the efficiency of structural topologies. The design domain, which is large enough to cover the final design, is divided into a fine mesh of finite elements. The nature of the material layout in an optimised structure depends on the type of constraint. A different type of constraint leads to different optimal topology. However, it has been found that the magnitude of stress constraints might have significant effects on the weight of a final design but not on the optimal topology. Therefore, the structural optimisation process can be divided into two steps. The first step is to obtain the optimal topology of the continuum structure using the PI and any structural optimisation method. The second step is to size the obtained optimal topology in order to satisfy the stress constraints, a procedure not attempted by Liang et al, [25]. The research presented in this paper evolves not only optimal topologies but also optimally sized topologies.

Louis et al, [19] use a GA, augmented with engineering heuristics, to design the topology, geometry and member properties of trusses, while minimising the weight and maintaining feasibility. Although the initial population is generated randomly, throughout the evolutionary process members are created to connect nodes such that the profile is, with high probability, triangularised thus giving a stable truss. Heuristics are constantly applied to the search process, with the aim of ensuring that recognisable truss structures are formed, resulting in useful solutions that benefit from optimised weight. Upon relaxing the use of the heuristics (a less constrained GA run), the results achieved by Louis et al [19] became very unorthodox and impractical. When comparing results achieved through the heuristically driven search, and the less guided search, it can be concluded that the best results were the product of the knowledge-based heuristics and not the exploration of the search space.

Du et al [21] attempted to predict both the optimal unrestricted material design as well as for design with a specified material, assuming that optimisation of the material may lead to considerable improvements in structural performance. Design variables representing material properties are decomposed into global and local measures. The method iterates between these to acquire the final results having both optimal global material distribution and optimal local properties. Optimisation is achieved by solving for optimal global material distribution for local properties, and then solving for optimal local properties in each point for global material distribution. The work of Du et al [21] evolves only one solution (as opposed to a population), and is therefore likely to converge prematurely on a sub-optimal solution. Additionally, evolving local properties on more complex structures may be computationally prohibitive. Like many attempts at complex optimisation problems, their research does not confirm the system's ability to produce optimum shapes.

For structural engineering, the most successful applications of evolutionary techniques to shape discovery have been in the area of truss design. For example, Azid & Kwan [8] describe a GA based method using genotypes as the representation which is shown to give good results.

Test Problem Outline

This research combines both the generative discovery of useful solutions, with the ability to optimise these solutions. This is achieved in an environment that is not constrained by preconceptions of how to solve the problem. The excessive use of heuristics for the direct modification of a solution based on domain knowledge is avoided.

The determination of the optimum cross sectional shape for a beam has proven to be a difficult problem to solve. To achieve satisfactory results, it has been necessary to break the problem down into a number of sub-problems and gradually build up the technique. The following describes this process.

The initial optimisation problem to be addressed is the design of a beam cross-section under a given loading condition and only considering the impact of bending. A cross-sectional shape is required that keeps normal stress levels, those caused by the application of a bending moment, within the constraint boundaries while utilising the minimal amount of material. Minimising the required material arrives from a desire to reduce self-weight. For this initial solution only bending moments are considered with the load being applied uniformly and symmetrically to the upper surface of the beam and the support being provided in the same manner at its lower surface. The solution to the problem is well known and requires the formation of two symmetrical flanges to carry the normal stresses. Further on in the paper, this initial optimisation problem is developed to include the requirement to simultaneously carry bending moment, shear and additional forces. The test problem is heavily constrained, reflecting real-world conditions, and can therefore be considered to be a difficult problem.

Although the application of only a bending moment makes the task appear relatively simple, the search for an optimal solution is a non-trivial problem. The search space includes local optima, solutions that meet applied constraints but do not optimise the use of material.

The GA must be capable of creating solutions from an initial random population which is not seeded with any base solution. Any learning achieved by the GA will then be the product of the evolutionary process. The system only needs to know what material constraints must be satisfied and general criteria such as minimum weight. It must also be able to locate the globally optimal solution. Test cases where the optimal solution is known permit an assessment of the effectiveness and efficiency of the GA and demonstrate its potential for application to optimisation problems where the optimal solution is unknown.

Voxel Representation

A voxel-based representation has been used. The choice of voxels also defines the *design space* (fig 1a) and the maximum possible dimensions of any beam cross section, which in the following is fixed throughout the evolutionary cycle. Further into this paper, the size constraint imposed by the maximum dimension of the design space will be removed.

The voxel representation decomposes the design space into a grid of identically sized squares. Each voxel is a division of the design space capable of being void or containing material of uniform density throughout the solution. Each square or subdivision is then represented in the genotype by a binary value which indicates whether the box is active (a binary value of one contains material) or inactive (a binary value of zero is void of material). Fig. 1b shows a voxel grid containing a simple shape, and how it would be represented on the genotype string.

The nature of the representation unfortunately leads to very long genotype strings. To provide adequate resolution across the design space a suitable number of voxels is required. As each voxel is represented by one gene, the higher the definition of the grid, the longer string. A balance is required between control over the design space and the resultant computational expense. For this work a grid resolution of 32*64, resulting in a genotype string of 2048 genes, has been used.

A potential limitation of the chosen form of representation is the 1-dimensional representation of a 2-dimensional design space. Adjacent voxels on the design map are not necessarily adjacent on the genotype string. The genetic algorithm has no knowledge of this, leading to standard genetic operators [22] causing more disruption than evolution.

The development of small holes, isolated voxels and jagged edges are also potential deficiencies in the solutions. Eliminating these deficiencies without having to apply strong guidance using heuristics poses a significant challenge.

The initialisation of individuals within the initial population randomly selects 70% of voxels to be active. This value supplies each design domain with sufficient material for the crossover operator to be effective. As dictated by fabrication and design requirements, user defined voxels can be permanently switched on or off, allowing for the provision of building services etc.

DESIGN, TESTING AND DEVELOPMENT OF THE GENETIC ALGORITHM

Stress Analysis Model

As explained above, the initial test case for the GA is the evolution of a beam cross-section subject to bending but without shear. Such a load case produces *normal stresses* over the entire cross-section of the beam which vary in intensity depending upon the shape of the cross section and the location of material within it. Each solution is evaluated utilising two standard stress analysis equations. The stress value σ for an individual active voxel can be calculated using the *bending stress* (Gere & Timoshenko, [23]):

$$\sigma = \frac{(My(i))}{\mathrm{I}} \qquad (1)$$

where *M* is the applied bending moment (Nmm), y(i) is the distance of the ith active voxel from the neutral axis in millimetres, and I(mm^4) is the *second moment of area* with respect to the horizontal neutral axis.

The second moment of area for the shape (with respect to the neutral axis) is calculated as (Gere & Timoshenko, [23]) shown below:

$$\mathrm{I} = \sum_{i=1}^{n} (y(i)^2)(A) \qquad (2)$$

where y(*i*) is distance of the *i*th voxel from the neutral axis of the shape in millimetres, *A* (mm^2)is the area of a voxel and n is the number of active voxels.

The neutral axis is defined as a horizontal line that passes through the centroid of mass of the shape. The voxel representation system applied in this research reduces the design space to a series of squares of uniform size and density. Therefore it is acceptable to calculate the neutral axis as being the average position of active voxels (average position of material).

$$NeutralAxis = \frac{\sum y_{base}}{Active} \qquad (3)$$

Where *Ybase* is the distance of the material from the bottom of the design space in millimetres and *Active* is the total number of active voxels.

It is evident from a simple analysis of these equations that the distance of an active voxel from the neutral axis is most significant in determining the optimal shape for the cross-section.

Increasing the average value of *y*, will increase the second moment of the shape and decrease the maximum value of voxel stress. Normal stresses require material to migrate towards the vertical extremities of the design space where the normal stress is greatest (forming two symmetrical flanges). A valid solution to this simplified mathematical model would be any shape that does not exceed the stress constraint, with the best solutions being those that satisfy this constraint with the minimum amount of material. As only normal stresses are currently being considered, there is no requirement for the GA to develop a web (primary shear stress carrier) or any other means of connecting the two elements.

Fitness Function

For evolutionary computing, explicitly defining an objective function (raw fitness value) is not always necessary, requiring only a fitness function. A GA's fitness function is utilised to allocate a score to each individual within a population, giving relative measures of performance. Higher fitness values are given to solutions that perform better in the chosen environment and are used to give them a greater opportunity for mating. The fitness function has been designed to minimise the area of the beam (active voxels) within the maximum allowed stress. Table 1 details the constraints (must be satisfied) and criteria (desirable qualities) to be satisfied by the evolved solutions. For all test cases the required criteria remain constant. Constraints are given priority over criteria through the use of penalties relative to constraint violation. However, no constraint is given priority over another.

The fitness function used when applying only a bending moment is shown below.

$$Fitness = 1000 \Big/ \left(Active + \left(1 \big/ SVoxelMax \right) + A \times (P1) + B \times (P2) + C \times (P3) \right) \quad (4)$$

where *Active* is the number of active voxels and *SVoxelMax* is the maximum stress value.

$$P1 = (SVoxelMax - \sigma_{\max})$$

$$P2 = (SVoxelMax - SVoxelMin)$$

P3 = (Number Of Exposed Voxel Sides)

A,B,C = Scaling factors (A=100, B = 30, C = 10)

$\sigma_{\max}$ is the maximum stress value permitted in N/mm^2 within the material and *SVoxelMin* is the minimum voxel stress.

The fitness of an individual is inversely proportional to the number of active voxels (required material) and hence favours minimum self-weight. To promote the efficient use of the material and minimise redundant stress carrying capacity, *(1/SvoxelMax)* is applied so that solutions operate at the most efficient stress values for the material utilised for the cross-section. *P1* is applied to solutions that violate the maximum stress constraint. *P1* is the only hard constraint at this point and therefore violations are penalised heavily. *P2* again encourages the GA to evolve solutions that operate at high stress levels to promote efficiency. *P3* is applied to minimise the development of small holes and isolated voxels by penalising solutions relative to the number of exposed voxel sides. *P3* also aids the development of solutions with minimal material as it requires voxels to form tightly packed shapes for optimality. Each isolated voxel or small hole will result in a *P3* value of four (four exposed sides), thus encouraging the GA to group together sections of material.

Scaling factors *A, B* and *C* are used to increase the weight of each variable, a standard practice when applying evolutionary searches. Scaling has been applied to prioritise the satisfaction of constraints over criteria. During initial generations the scaling factors for constraint violations are set relatively low to allow the GA to explore the greatest range of potential solutions. During final generation the scaling factors are increased to ensure the final solution meets all constraints. Additionally, the scaling factors are applied to ensure that no single variable dominates the search. This is necessary as the range of values for each variable varies significantly. For example, during final generations, *Active* has an average value of 750 per solution, where as *P1* tends towards +/- 3. These penalties can ensure all constraints are satisfied simultaneously, without any bias. Scaling factors are also applied to the criteria elements of the fitness function to ensure no one criteria dominates another.

The fitness function scaling factors are a form of heuristic. However, unlike Baron [2] and Chapman [17] these heuristics are not applied directly to the design space modifying solutions.

More appropriately, the heuristics merely weight the importance of certain constraints and criteria and hence are used to guide the search without making arbitrary adjustments to the solutions.

Advanced Two-dimensional Genetic Operator Application

To protect schema (sections of good material) within a solution, genetic operators are required that can allow offspring to contain the better elements of their contributing mates. Azid and Kwan [8] applied evolutionary computing to the layout optimisation of simple structures. Mismatches in genotype length, and large structural differences between mates made it difficult to devise a crossover operator robust enough to produce offspring structures that bore an architectural resemblance to its two parents.

Standard genetic operators, simple one-point crossover and bit-flip mutation [22], were applied in this work during initial testing. The random splitting of strings, caused by the crossover operator, proved too disruptive to the evolutionary cycle to produce good results. Simple bit-flip mutation (randomly altering one gene in a string of 2048) also proved inefficient at maintaining genetic diversity, even when applied at very high rates.

The disruptive nature of one-point crossover can be explained through fig 1b by noting that adjacent voxels on the genotype, are not necessarily adjacent in the design space. As the design space is two-dimensional, to protect the schema, the genetic operators must also be two-dimensional to reflect the two-dimensional nature of the problem. Fig. 2a shows in concept the principle of the crossover mechanism and mutation operators that have been developed. These operators are applied in relation to the design space and not the genotype string, thus, bridging the gap between the 1-dimensional string and the 2-dimensional design space. The operators are more reflective of the nature of the design space.

The mutation operator has also been developed to reflect the two dimensional nature of the search space. With the 2D operator, mutation is applied to a 2*2 voxel area, effectively increasing its effect from one voxel to four at each application, fig 2b. This results in the mutation operator supplying the population with significantly more new material at each generation.

Results

The following results were achieved through the application of two-dimensional crossover and mutation and the normal stress (i.e. bending only) evaluation module.

Fig. 3 shows typical solutions evolved by the GA after the 2000th generation. Note that the problem is exactly the same as that employed by Baron [2] but that the results have been achieved without the use of heuristics which arbitrarily remove material from or add it to the design space. Additionally, the quality of the evolved solutions exceeds the quality achieved by Chapman [17], whose solutions contain internal voids and a lack of geometric smoothness. Table 2 highlights the details of the best evolved solutions, at the end of a 2000 generation run (60000 evaluations). In addition to near optimal shapes being achieved (+/- 5 voxels from known optimal), the solutions are within the stress constraint limits. The GA has a success rate of approximately 90% (locating near optimal solutions).

Both Roulette and Tournament selection operators have been used (Goldberg, [22]), with Tournament selection performing significantly better. In addition elitism was applied, ensuring that the best solution within a current population is automatically selected for reproduction, before tournament selection is applied.

INCLUDING SHEAR FORCES

The second part of the this optimisation problem is to make the beam able to carry both a bending moment and a shear load, hence the cross-section is required to simultaneously carry the induced normal and shear stress. The optimal solution to the dual load cases is shown in fig. 4 and highlights how the inclusion of shear loading requires a web connecting both flanges to complete the shape.

Shear loading results in complex stresses, acting in different directions depending on the location within the cross-section under consideration. A suitable model is required for the analysis of the shear stresses and also a suitable evaluation system capable of judging the chosen model. As the central theme of the work is that there can be no assumptions regarding the form of evolved shapes, the application of simple mathematical formulae is not possible. Instead, a model is required that is not only applicable to the organised shapes which are to occur later in the evolutionary process but also the non-regular shapes likely to be present in the initial generations (including fragmentation and holes etc).

The bending moment, with the additional shear load, leads to a multi-constraint optimisation problem with the combination of bending and shear requiring material to migrate in multiple directions, to satisfy the constraints. To satisfy normal stresses, material must move to the vertical extremities of the design space (Fig.4). To satisfy shear stresses, material must move to the horizontal neutral axis where shear stress is greatest. This provides a basis for constraint conflicts as if either constraint is unfulfilled, the solutions become unfeasible. The ability to safely carry bending and shear force in terms of stresses is the only hard constraint applied. There is no constraint or validity check applied in relation to the actual evolved shapes.

Stress Analysis Model

The Shear Force Evaluation Module (SFEM) is required to determine how shear stress varies throughout the cross-section and be applicable to any shape evolved by the GA. The primary variable required to form good webs is the maximum shear stress. To calculate this, the model is required to determine which voxel is the most heavily loaded, and calculate the shear stress induced in this voxel.

To determine the maximum shear stress induced in complex shapes, a Finite Element Analysis (FEA) is typically applied. Although many FEA packages are available, they tend to be computationally expensive. To avoid a system, which requires excessive run times, a SFEM designed specifically for dealing with shear stresses in a voxel representation has been developed.

S*hear flow analysis* [23] was determined to be the most robust model for the given problem. *Shear flow* follows stress paths from the point of load application through the entire shape. Fig.

5a, shows the principle of shear flow applied to an I-beam. Essentially the shear flow model analyses each voxel, determines the load it receives from adjacent voxels and, as the shear stress accumulates on voxels, a value for maximum loading can be deduced.

Voxel to Voxel Shear flow

Fig. 5a shows a section through an I-beam and the transfer of shear flow from the point of load application down through the shape. The shape is divided into small square sections, with each one representing an active voxel in the design space. The diagram highlights how shear flows down through the upper flange, then across the bottom of the flange to the web/flange intersection. At this point, shear flows from the left and right hand side of the flange combine then travel down the web to the bottom flange.

In-order for the GA to receive all necessary information, the SFEM must be able to reproduce a model of this shear flow, which includes every active voxel within the design space and for any evolved shape. It must therefore perform a voxel to voxel analysis of the shear stress for the entire shape.

The voxel to voxel shear analysis studies each voxel sequentially through the length of the genotype string. For the current voxel, the SFEM determines which adjacent voxels are supplying it with load and to which voxels it passes this load. Fig. 5b, highlights the procedure for the current voxel within the design space. The system develops a shear stress map for the entire shape, highlighting sections of the shape where high accumulations of shear stress occur. Having performed the stress analysis, the value of the most heavily loaded voxel can be determined.

Direction of Shear Flow

As discussed, the direction of shear flow is dependent on the location within the design space. When flow reaches a point where it can no longer pass downwards, it must flow towards the centre of the design space, until it next has the ability to flow downwards. This assumes that the loading is symmetrical and uniform. At some point flow travelling in one direction will meet with flow from the other direction. This intersection of flows is typically the web/flange intersection. The location of the web in the centre of the cross-section is to ensure that the build up of shear along the bottom of the top flange is minimised (Fig. 6). If shear in the flange is allowed to build beyond a given limit, then the flange will buckle.

The optimal method for carrying shear in an I-beam is to have a thin, vertical, and centrally located web that connects both flanges. The web provides a means for shear to continue its flow and to avoid the accumulation of excessive stress in the top flange. Additionally, the web connects both flanges to create a complete shape.

Although a single central web is the optimal solution for this problem, there are possible alternatives; local optima that provide good or at least valid solutions. Closed thin walled sections, multiple webs, solid rectangular sections, or a non-centralised thick web can all accommodate the load case. Each solution allows shear to flow downwards but not at the optimal location. Fig. 7, highlights some of these feasible solutions.

For any thin walled section, there must be some point on the solution where shear flow from the left and right and sides meets and travels downward (this can potentially happen at multiple points resulting in multiple webs). In-order to allow the GA explore any shape this intersection point must be allowed to be located anywhere. This potentially allows the creation of I-beams, C-channels, or even closed tubular beams.

The SFEM must reflect that the direction of shear flow is dependent on the location within the cross-section. Additionally, the load a voxel receives and the voxels to which this load may be passed is dependent on the direction of flow. For vertical flow, the model represented in fig 5b is suitable for representing all shear flow variables. However, for the parts of the cross-section where flow is restricted to horizontal flow only (vertical flow not possible due to the arrangement of voxels), fig 5b can be simplified. Upon determining that vertical flow is no longer possible, the model is simplified to fig 8a, and 8b, to reduce computational times.

Figs 8a, and 8b, highlight how, in regions of horizontal flow only, the SFEM analyses the voxels contributing to the current direction of flow. Fig 8a, and 8b, transmits loads to a voxel only in the relevant direction of shear flow.

Fitness Function

The fitness function for this problem has to encompass both the bending moment and shear force cases. Only the value of maximum shear stress is supplied to the fitness function regarding the applied shear force and this has to be scaled sufficiently to ensure the GA tries to minimise its value (Table 3). Some readjustment of all variables within the fitness function is required to ensure that the additional factors are compatible with the remainder of the solution requirements.

$$Fitness = 1000 \Big/ \left(Active + \left(1 / SVoxelMax \right) + A \times (P1) + B \times (P2) + C \times (P3) + D \times (P4) \right)$$

(5)

P4 = Maximum value of shear stress

A,B,C,D = Scaling factor (A=100, B = 30, C = 10, D = 200)

Initial Tests and Problems with the SFEM Solutions

The addition of shear force results in a more complex fitness function. As it becomes more difficult for the GA to locate good solutions, any weakness in the evaluation system is highlighted. It is useful to look at some of the aspects of the evolution of the SFEM to show what problems had to be overcome.

The first prototype for the SFEM traced the path of shear flow from the point of load application through the evolved shapes. However, the system was flawed because it did not allow an accurate representation of shear stress. The initial development allowed shear to flow into a voxel, then searched for a path (active voxels) below or horizontally adjacent to the current voxel. Tracing shear flow to a voxel, such as the one illustrated in fig 9a, leads to a situation where there is no further path available for shear to flow. As no provision was made for shear to flow upwards out of the voxel, high shear stress concentrations would occur at such voxels. The GA responded to this by having nodules form along the base of the top flanges, fig. 9b. These nodules disrupted the flow of shear as it moved to wards the web and were in effect a means of limiting the accumulation of shear stress. Instead of two stress concentrations at the web/flange intersection, each accounting for half the applied load, four smaller stress concentration often developed through the development of these nodules

Allowing shear to flow upwards, essentially allowing shear to flow into and then out of such nodules, as illustrated in fig 9c, provides a more realistic model of shear stress. This results in the shear flow being distributed around such nodules and minimises their effect on maximum stress values achieved in the cross-section.

The increased complexity of the problem also leads to optimisation difficulties. Firstly, the GA is required to develop three beam elements simultaneously (a web and two flanges), with the elements being connected. During initial stages the flanges are very thick and hence carry far more shear loading than is required in the final solutions. This leads to the formation of a very thin web during initial generations, as the flanges limit the load it is receiving. As the generations progress and the flanges move closer to their final, much reduced, size the web starts to become insufficiently thick to carry the load it now receives. The efficiency of the GA tends to remove unnecessary material in early generations leaving a shortage of active voxels to thicken the web. Fig. 10 demonstrates the typical evolution of an I-beam over 2000 generations.

This problem was overcome by dynamically varying the penalty functions. During initial generations, the penalties are relatively small, allowing the GA to explore a large range of potential solutions. Constraint violation penalties are gradually increased, without preference towards any one constraint, to ensure all final solutions are at-least within constrain boundaries, if not optimal. Ensuring all constraints are equally important further emphasises the learning ability of the GA by not forcing the evolution of predefined a shape element through heavily penalising the omission of certain shape elements.

Shear & Normal Stress Results

The results highlighted, (fig 11) have been achieved through a combination of the GA, two-dimensional operators, the bending moment evaluation module and the shear stress evaluation module. As the optimal results are well documented, the results evolved by the GA can be directly compared and an evaluation of the GA's ability to create and optimise has been undertaken.

Care has been taken to ensure the shear force evaluation module does not unduly influence the form of the final solutions. The evaluation module is not instructed to replicate any predetermined shear flow paths, other than to direct flow down through the shape. As this research deals with realistically constrained problems, realistically large loads are applied to the shapes. This necessitates a suitably thick web in addition to suitable flanges.

The results were achieved through only supplying the GA with information regarding material constraints and the need to minimise self-weight. Only maximum shear and normal stress is required to allow the GA to find these optimal shapes. Based on the stress value of the voxel with the maximum load, the GA arranged the other active voxels to reduce this value. The GA has a success rate of approximately 80% (locating near optimal solutions) after 2500 generations (75000 evaluations).

The evolved results contain the three desired elements (web and two flanges), with each being optimally located and shaped for the given constraints. Table 4 contains key information about evolved solutions and demonstrates that in-addition to the solutions taking the form of optimal shapes, all constraints are also met.

The GA's ability to evolve optimal solutions was heavily dependent on the amount of learning achieved during the initial shape discovery period of typically 10% of the total of 2500 generations. Material migrates from the middle third of the design space to both vertical extremities, leaving only the web connecting both flanges. If the integrity of the web is maintained beyond the first 200 generations then optimal results were usually achieved. If the integrity of the web is lost at any point in the run the shape soon disintegrates, with the GA concentrating on satisfying only normal stress constraints.

The complexity of this optimisation problem leads to a large search space with only a very small percentage of it containing feasible solutions and with an even smaller percentage of it containing optimal solutions. On a large percentage of runs, the GA was capable of locating a region of high fitness but with the search space containing many local optima (due to constraints and criteria), if for any reason there was even a slight deviation from the optimal region, the GA would locate the nearest local optima and converge on that.

The problem of local optima is compounded by the fact that during early generations, shear stress values are significantly lower than those that occur in the optimal shape. As discussed, it is not until good flanges are formed that the web starts to receive more realistic stress values. Therefore, during initial generations, the difference between local optima and the true optimal solution appears less pronounced to the GA. The application of dynamically varying penalty functions minimised this problem by balancing current maximum loading with possible maximum loading at later generations.

Inclined Bending Moment Applied at the Neutral Axis

To further increase the complexity of the search and investigate the robustness of the evolutionary design system, an additional load case was applied to the design space. An inclined bending moment is applied through the horizontal neutral axis. The load has been applied, in addition to shear and bending moments, through the centroid (C) of the cross-section, at a user defined alignment. Fig.12 shows the horizontal loading applied to an I-beam.

The inclusion of this horizontal loading requires further evaluation of the evolved shapes to determine their ability to safely carry all loads. An Inclined Bending Moment evaluation module was attached to the core GA. Several variations of evolved shapes can be considered as suitable for this combination of loading. Increasing the thickness of the web, or alternatively, splitting the web into two, thus creating a closed box section are suitable options. Both solutions have the affect of increasing the second moment of area about the vertical axis, and hence reducing the maximum value of stress.

Evaluation system

An additional evaluation module was attached to the GA for the purpose of evaluating solutions with an inclined loading. Solutions are evaluated by applying the following equation [23]):

$$\sigma_{\max} = \frac{M_y \times x}{I_y} - \frac{M_x \times y}{Ix} \quad (6)$$

Where $\sigma_{\max}$ is the maximum stress (Nmm^{-2}), Iy and Ix are the second moments of area relative to the vertical and horizontal principle centroidal axis respectively (mm^4). My and Mx are the applied bending moment resolved in the x and y direction, x and y are the co-ordinates of material located furthers from the point of load application. This information is then incorporated into the fitness function by means of an additional penalty:

$$Fitness = 1000 \Big/ \left(Active + \left(1 / SVoxelMax \right) + A \times (P1) + B \times (P2) + C \times (P3) + D \times (P4) + E \times (P5) \right)$$

(7)

P5 = Maximum Stress induced by the inclined loading

A,B,C,D,E = Scaling factor (A=100, B = 30, C = 10, D = 200, E = 10)

Results

As anticipated, the inclusion of this additional loading did not make significant changes to the evolved shapes. To minimise the stresses induced by this loading, the GA consistently thickened the web, thus maintaining optimal shapes within the new stress constraints. Evolved results were very similar to those in fig 12, with a slightly thickened web to carry the inclined load. Similarly to previous test cases, 2500 generations and a total of 75000 evaluations were required to locate near optimal shapes.

Removal of Size Constraint

The size of the design space (the available space in which a solution evolves) is set before the start of the evolutionary process. This sets a limit on the maximum dimensions available for the beam cross-section designs and is essentially a shape constraint. Removing this constraint, allowing shapes to evolve cross-sections that are not restricted in size, is the final step in this research.

Due to the nature of a voxel representation, allowing the design space to vary in size during the evolutionary process will create varying length genotype strings if a constant voxel resolution is to be maintained. Variance in genotype lengths may require alteration to the cross-over operators to maintain a robust search (Azid and Kwan, [8]). To maintain the theme of evaluation modules being easily combined without the need to alter he core GA, it was determined that a fixed sized design space is a necessity.

To minimise normal stresses within the cross-sections, material is pushed to the extremities of this design space. However, there is a practical limit on the overall height of a cross-section. If the height to width ratio becomes too large, beams are likely to fail due to lateral buckling. Setting the maximum dimensions of the design space to an infeasible size would require the GA to evolve shapes within these maximum boundaries in-order to create optimal shapes (in particular minimising the material required in the web).

All past tests involved setting the design space with a height of 300mm and a width of 175 mm (maximum available dimensions). To satisfy the stress constraint imposed by the material, the GA required this maximum height. For the next series of tests the overall height is increased to 350mm. The required optimal dimensions for the cross-section not have changed, but the space available to evolve the cross-sections has increased. Optimal dimensions are now less than those available, requiring the GA to evolve designs that do not utilise the full design space extremities, hence limiting the effects of the final shape constraint.

Several new problems arise when removing this final shape constraint. Firstly, and most importantly, difficulties arise in applying shear loading to the cross-sections. For the initial tests it could be assumed that the shape would be connected to the top of the design space, and hence, the loading was applied to the top row of voxels. The GA now has freedom to adjust, vertically and horizontally, the position of the solution within the design space. Finding an appropriate means of applying load to the shape under such conditions is a difficult task. Fig. 13 highlights the anticipated problems associated with applying the load.

Solution

A suitable platform for applying shear loading to the evolved shapes has been achieved as follows. A search is conducted up through the centre of the web(s), starting from the horizontal neutral axis and stopping when the search encountered the first inactive voxel. This is then determined to be the top end of the web and the horizontal level on which the shear loading is applied. Figure 14 highlights this principle. Such a system avoids trying to apply loads to isolated voxels.

The shear load is only applied to active voxels directly connected to the web, thus ensuring that the load is applied to the main body of the shape and again avoids applying loads to isolated voxels. Ensuring shear loading is applied to the main body of the shape, and not isolated sections, reduces computational effort on material unlikely to be present in the final solutions.

Results

Applying the fitness function (eqn. (5)) resulted in the GA maintaining the use of the vertical extremities, hence producing inefficient solutions with very low normal stress values. A small alteration to the fitness function was applied to encourage the GA to evolve shapes with high normal stress values and therefore not utilise the extremities of the design space.

$$Fitness = 1000 \Big/ \left(Active + \left(F \times (EfficientStress - SVoxelMax)\right) + A \times \left(P1\right) + B \times \left(P2\right) + C \times (P3) + D \times (P4)\right) \tag{8}$$

Efficient Stress = Desired working stress for the material

A,B,C,D,E, F = Scaling factor (A=100, B = 30, C = 10, D = 200, E = 10, F = 10)

This alteration to the fitness function is applied to emphasise the requirement for efficiently high stress values and hence indirectly penalise solution that utilise the height extremities of the design space. Fig. 15, demonstrates typical evolved solutions. Near optimal solutions have been achieved with a success rate of 75% (Table 5).

Alternative Cross-section Shapes

Thus far, the applied test cases have a known optimum solution consisting of an I-beam shape. However, the shape discovery and optimisation system has the ability to evolve various cross-sectional shapes as dictated by the applied loading. This section supplies example of the range of shapes evolved by the system under various loading conditions.

Applying either a large shearing load, or alternatively, a relatively low shear stress constraint leads to the development of a closed section. This occurs as the GA minimises the accumulation of shear stress at the top web / flange interface by creating two webs, thus limiting the build-up of shear stress on the top flange. Closed sections also occur under inclined bending, as a result of the GA reducing Iy (and hence internal stress) by moving material away from the vertical neutral axis. C-channels also tend to evolve under inclined loading, especially if there is only a small vertical shear loading. Varying the combination of applied loads, and the size of the loads, results in the evolution of a variety of cross-sectional shapes (fig.16).

Conclusions

This paper has demonstrated the shape discovery and optimisation abilities of a Genetic Algorithm when applied to realistically constrained design problem. Unlike many other applications, this GA system does not rely on heuristics which operate directly on the genotype, or any predefined notion of the solution to the problem. Having demonstrated the GA's ability to perform efficiently and effectively under such condition, it can hopefully be applied with confidence to situations where the optimal solution is not known in advance.

When combined with suitable evaluation modules, the GA, consisting of only crossover, mutation and tournament selection is all that is required to evolve optimal results. No additional genetic operators to repair poor solutions are necessary.

Several evaluation modules have been applied to the GA, requiring the GA to simultaneously satisfy multiple constraints. The difficulty inherent with the stress analysis, and the ease with which the GA locates optimal solutions highlights the possibility of broadening the evaluation modules.

The true potential for this GA application lies with problems where there is no currently known optimal solution. Having demonstrated the effectiveness of GA shape discovery and optimisation for a problem with a known optimum, it may potentially be applied where there is no known optimum. The GA system described here has not relied on application specific details. Only the evaluation modules contain such information but are not an intrinsic part of the GA. They can be combined and interchanged to evaluate any stress or shape requirements. Any number of evaluation modules can potentially be attached to the GA as they are only directly connected to penalties within the fitness function. New evaluation modules can be designed and tested without having to alter the core GA.

Future Work

The application of further load cases to the design space is required to investigate the potential of the system in more realistic design scenarios. For example, the ability to resist torsion is often a necessary design factor, and can be incorporated into the evolutionary system through attaching a torsion evaluation module.

The true potential of this system can only be realised when applied to a combination of load cases where the optimal beam cross-section is not known. The rapid generate-and-test nature of the GA system has the potential to evolve good alternative solutions that may not be evolved through more computationally expensive traditional design methods.

References:

1. Bentley, P.,. Evolutionary Design by Computers. Morgan Kaufmann. 1999

2. Baron, P., and Tuson, A., and Fisher, R. A Voxel Based Representation for Evolutionary Shape Optimisation. Special Issue on Evolutionary Design, AIEDAM journal v13:3 (1999).

Bentley, P.J. (Ed).

3. Rozvany, G.I.N.,. Shape and layout optimization of structural systems and optimality criteria methods, Springer-Verlag Wien1992 , 496pp.

4. Juels, A. & Wattenberg, M.. Stochastic Hillclimbing as a baseline method for evaluating genetic algorithms ,Department of Computer Science, University of California. 1994

5. Wolpert, D.H. & Macready, W.G.,. No free lunch theorems for optimisation, IEEE transactions on evolutionary computation1997 , 1(1),pp. 67 – 82.

6. Ruthenbar, R.,. 'Simulated Annealing Algorithms: An Overview,' IEEE Circuits and Devices Magazine. 1989

7. Shea, K. & Cagan, J.,. Innovative dome design: Applying geodesic patterns with shape annealing, AIEDAM, 11, 1997 pp 379-394.

8. Azid, I.A. & Kwan A.S.K.. A layout optimisation technique with displacement constraint, in Topping & Kumar (eds), Optimization & Control in Civil & Structural Eng, Civil-Comp UK 1999, 71-77.

9. Rasheed, K.M.,. 'GADO: A Genetic Algorithm for continuous design optimisation', dissertation, Graduate School New Brunswick, Rutgers State University of New Jersey. 1998

10.Robinson, G., Keane, A.,. Optimisation in Mechanical Design. Evolutionary Design by Computers, 1999, 147-165.

11. Sisk, G. The use of a GA-Based DSS for Realistically Constrained Building Design. PhD, School of Engineering Cardiff University., 1999

12. Gen, M. & Kim, J.. A GA-based approach to reliability design. Evolutionary Design by Computers, 1999 pp 191-219.

13. Harvey, I, and husbands, P., and Cliff, D., Issues in Evolutionary Robotics. In J-A Meyer (ed), From Aminals to Animats 2: Proc. Of the Second Intl. Conf. On Simulation of Adaptive Behaviours, MIT press 1997, pp364-373 .

14. Parmee, I.C. & Bonham, C.R.,. Towards the support of innovative conceptual design through interactive designer/evolutionary computing strategies, AIEDAM,14 2000, pp. 3-16.

15. Bentley, P. & Wakefield, J.. The Evolution of Solid Object Design Using Genetic Algorithms. Modern Heuristic Search Methods 1996, 197-211

16. Hawkes, B. and Abinett, R. The Engineering Design Process. Longman Scientific and Technical. 1993

17. Chapman, C.D., and Saitou, K., and Jakiela, M.J., Genetic Algorithms as an Approach to Configuration and Topology Design. ASME Journal of Mechanical Design 1994 .

18. Pablo, F. and Pollack, J. Computer Evolution of Buildable Objects. Fourth European Conf. on Artificial Life, MIT Press 1997 , pp 358-367.

19. Louis, J. and Zhao, F. Domain Knowledge for Genetic Algorithms. International Journal of Expert Systems. Blackwell Publishing. 1999

20. Xie, Y. M. and Steven, G. P. *Evolutionary Structural Optimisation*, 1997 (Springer-Verlag, Berlin

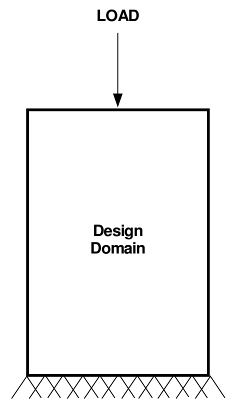
21.Du, J. and Taylor, J.E. . Application of an energy-based model for the optimal design of structural materials and topology. Structural Multidisciplinary Optimisation. Springer – Verlag 2002

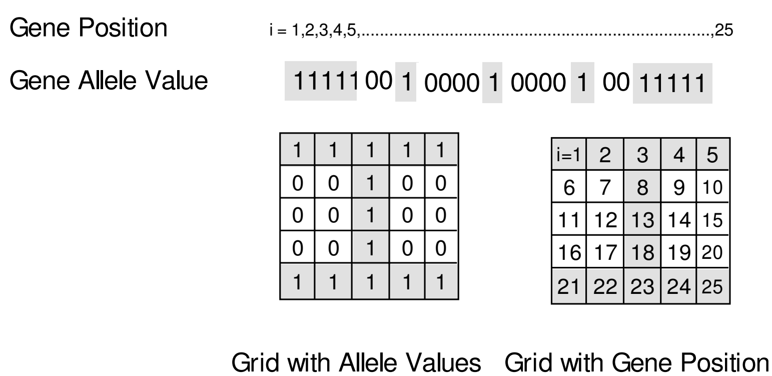
22. Goldberg, D.,. Genetic Algorithms in search, optimisation, and machine learning, Addison Wesley Longman, Inc. 1989

23. Gere, J. & Timoshenko, S.,. Mechanic of materials (Fourth Edition), PWS Publishing Company. 1997

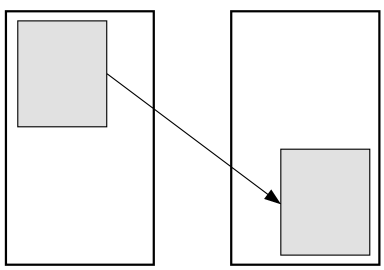
24. Cvetkovi´c, D. and Parmee, I.. Multi-objective Optimisation and Preliminary Airframe Design. In ACDM'98, 21–23 April, Plymouth, UK 1998, pp. 255–267.

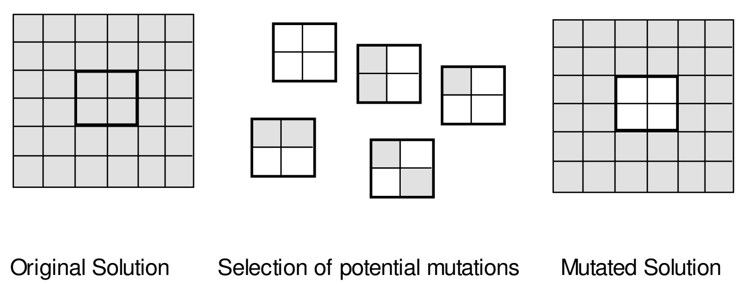
25. Liang, Q.Q, Xie, Y.M, Steven, G.P, "Optimal selection of topologies for the minimum-weight design of continuum structures with stress constraints," *Journal of Mechanical Engineering Science*, Proceedings of The Institution of Mechanical Engineers, UK, Part C, 1999, 213(C8), 755-762.

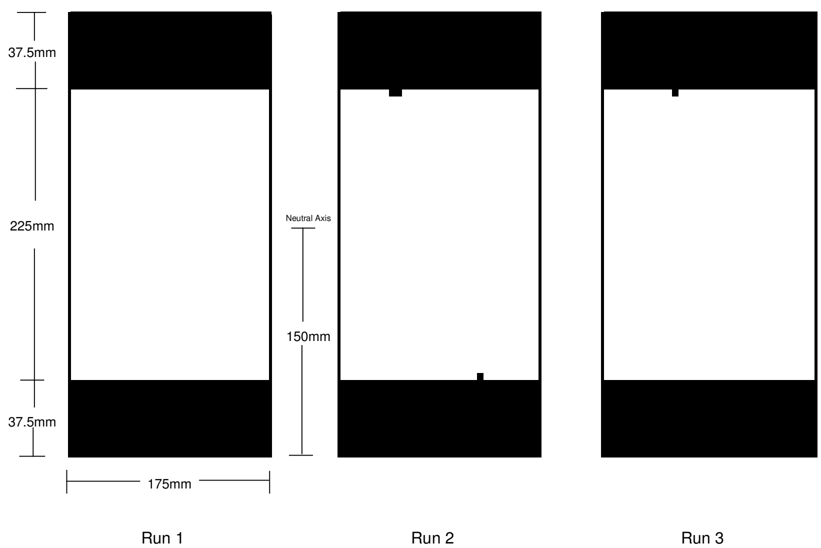

LOAD
Design
Domain



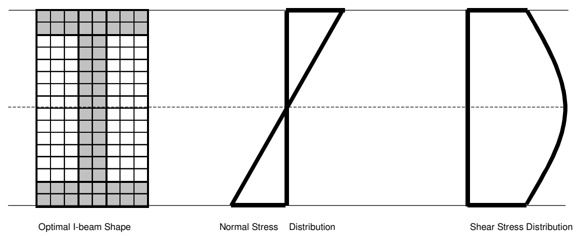
Gene Position
i = 1,2,3,4,5,..,25
Gene Allele Value
11111 00 1 0000 1 0000 1 00 11111
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
1 1 1 1 1
i=1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
Grid with Allele Values
Grid with Gene Position



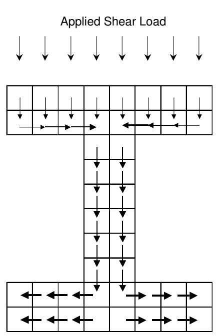

Original Solution
Selection of potential mutations
Mutated Solution

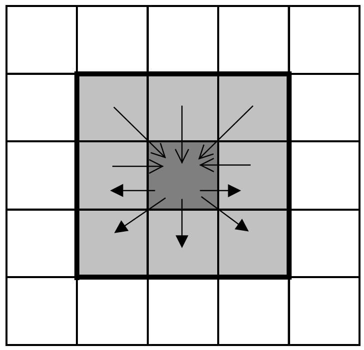


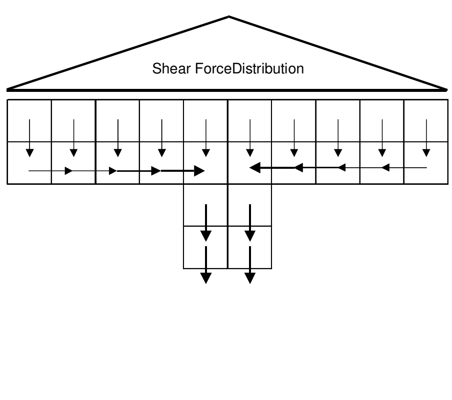
37.5mm
225mm
37.5mm
175mm
Neutral Axis
150mm
Run 1
Run 2
Run 3



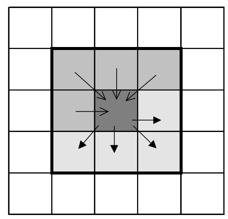
Optimal I-beam Shape
Normal Stress Distribution
Shear Stress Distribution

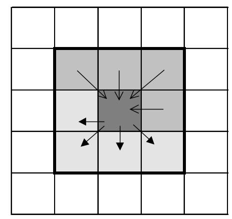

Applied Shear Load

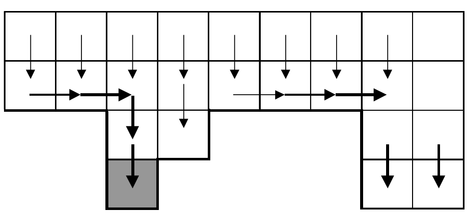



Shear ForceDistribution

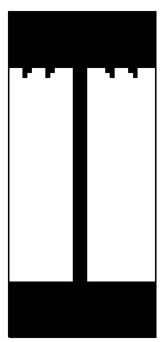
[illegible]

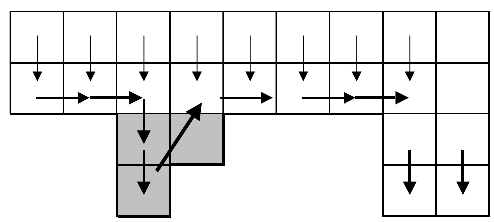




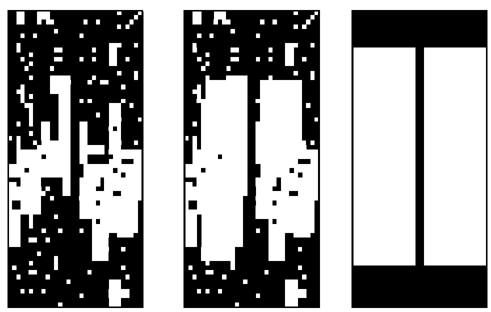


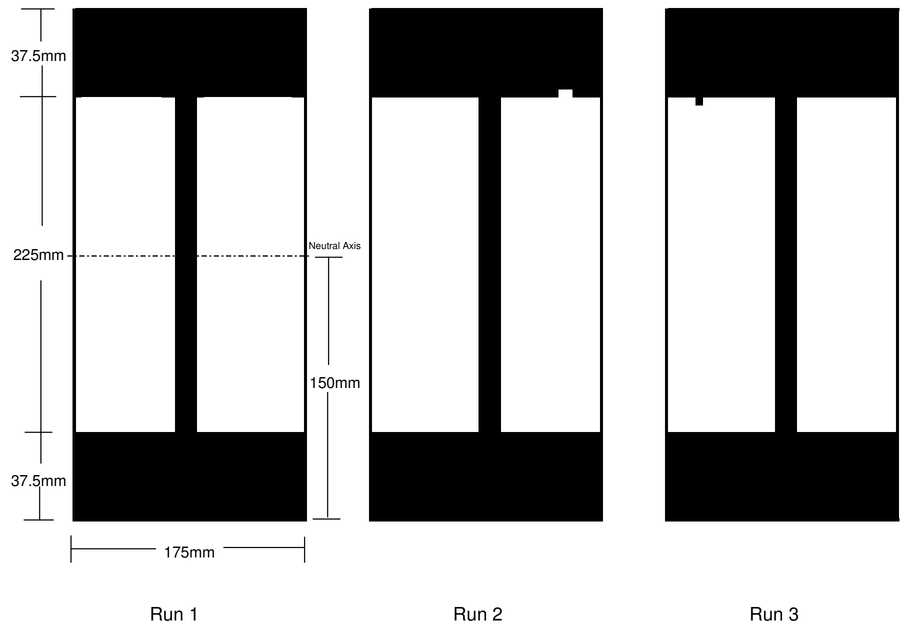
Flow cannot pass through or around this darkened voxel.

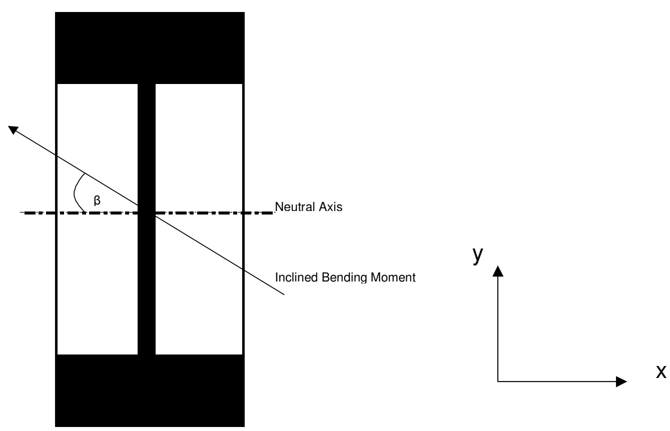


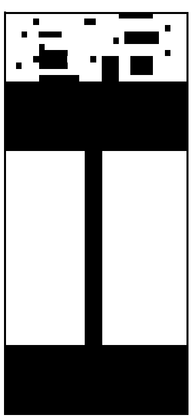


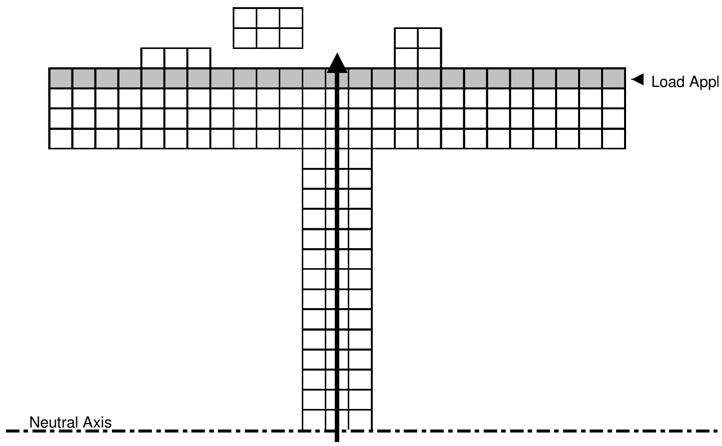
Shear flow can now flow out of any voxel, distributing shear flow around such nodules.



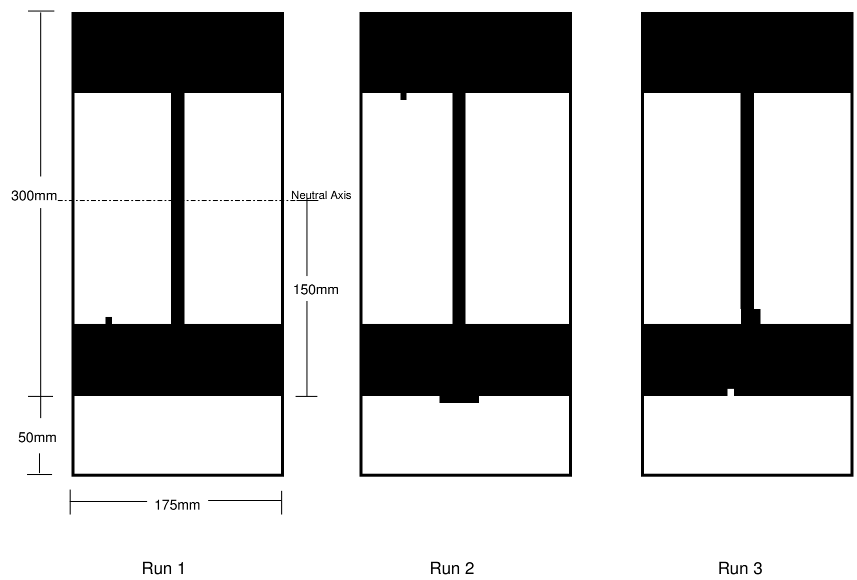

37.5mm
225mm
Neutral Axis
150mm
37.5mm
175mm
Run 1
Run 2
Run 3

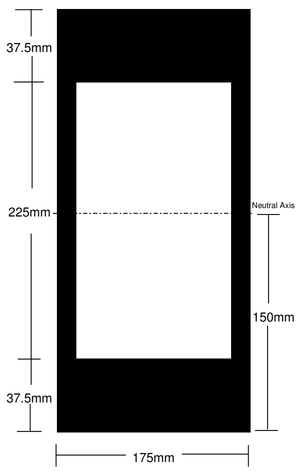

β
Neutral Axis
Inclined Bending Moment
y
x



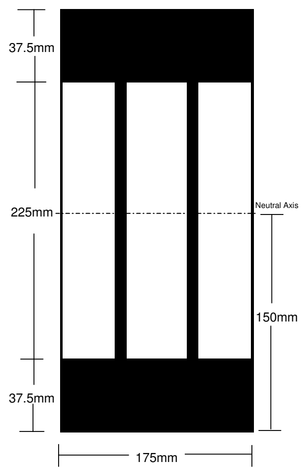


Load Appl
Neutral Axis

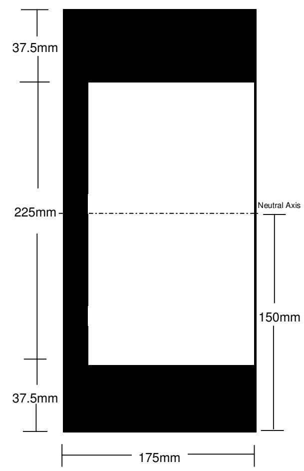

300mm
Neutral Axis
150mm
50mm
175mm
Run 1
Run 2
Run 3



Run 1



Run 2



Run 3

Constraint	Variable
SVoxelMax < = σ_{max}	P1 = (SVoxelMax - σ_{max})

Criteria	Variable
Minimise material used	Active
Minimise material used	P3 = Number of exposed sides
Efficient use of material	1 / SvoxelMax
Efficient use of material	P2 = (SVoxelMax – SVoxelMin)

Table 1 Constraints and Criteria

Run	Active Voxels (%)	Bending Stress/ N/mm^2	Surface Area / Exposed Voxel Sides	Optimal Shape Achieved
Initial Population	50 (random generations)	Average 155.00	Average 1925	N/A
1	34.38	100.01	172	Yes
2	34.52	99.09	179	Yes
3	34.42	99.85	175	Yes

Tests conducted with a design space of 300mm by 175mm, and an applied Bending Moment of $200 \times 10^6 Nm$, $\sigma_{\max} = 100MPa$.

Table 2 – Results for the Bending Only Problem

Additional Constraint	**Penalty**
Do not exceed max normal stress allowed (ShearVoxelMax < = τ_{max})	P4 = (ShearVoxelMax - τ_{max})

Table 3 - Constraints

Run	Active Voxels (%)	Bending Stress/ N/mm^2	Shear Stress/ N/mm^2	Surface Area / Exposed Voxel Sides	Optimal Shape Achieved
Initial Population	50 (random generations)	Average 155.00	Average 60	Average 1925	N/A
1	38.48	86.02	234.375	252	Yes
2	38.38	86.00	234.375	254	Yes
3	38.52	84.32	234.375	254	Yes

Tests conducted with a design space of 300mm by 175mm, an applied Bending Moment of $200 \times 10^6\ Nm$, , applied Shear Force of 500 N, $\tau_{\max} = 250 N/m^2$

Table 4 – Results for the combined Shear and Bending Problem

Run	Active Voxels (%)	Bending Stress/ N/mm^2	Shear Stress/ N/mm^2	Surface Area / Exposed Voxel Sides	Optimal Shape Achieved
Initial Population	50 (random generations)	Average 155.00	Average 60	Average 1925	N/A
1	38.53	86.02	234.375	254	Yes
2	38.77	86.00	234.375	256	Yes
3	38.53	84.30	234.375	255	Yes

Tests conducted with a design space of **350mm** by **175mm**, an applied Bending Moment of $200 \times 10^6 \, Nm$, , applied Shear Force of 500 N, $\tau_{\max} = 250 N/m^2$

Table 5, Results after 2500 generation(75000 evaluations), with a total runtime of approximately 100 minutes