

**Evolutionary Shape Optimisation
Using a Voxel Based Representation.**

Peter James Baron

**MSc Information Technology: Knowledge Based Systems
Department of Artificial Intelligence
University of Edinburgh
1997**

THE UNIVERSITY OF CHICAGO
LIBRARY

1965

THE UNIVERSITY OF CHICAGO
LIBRARY
1965

Abstract

Shape optimisation within constraint limits 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 observing some constraints. Much work has been done on various problems within this domain [Bentley 96][Chapman *et al.* 94] [Smith 95a][Watabe & Okino 93], however the majority to-date has focused on parametric and other highly restricted representations of the shapes being optimised. This thesis describes experiments performed using a voxel based representation and an evolutionary optimisation algorithm. As any attempt to perform shape optimisation using an unrestricted object representation must use a highly effective evaluation function in order to be able to deal with the varieties of design that an evolutionary optimisation algorithm may produce, the work detailed here was intended to first design and test suitable operators to deal with the extremely large chromosomes required by a voxel representation, and by then extending the work to permit evaluation from a commercial Finite Element analysis package, it was expected that the program would be able to perform real-world shape optimisation tasks with very few restrictions on the shapes being tested. The results obtained indicate that it is possible to circumvent the difficulties presented by a voxel representation [Watabe & Okino 93], and that with suitable operators, useful and unexpected results can be gained from a real-world optimisation task.

The first part of the document discusses the importance of maintaining accurate records of all transactions and activities. It emphasizes the need for transparency and accountability in financial reporting. The second part of the document provides a detailed overview of the company's financial performance over the past year, including a breakdown of revenue, expenses, and profit. The third part of the document discusses the company's strategic goals and objectives for the upcoming year, and outlines the key initiatives and projects that will be undertaken to achieve these goals. The fourth part of the document provides a summary of the company's financial position and outlook, and concludes with a statement of confidence in the company's future success.

Acknowledgements

I would like to thank my primary supervisor Dr. R. Fisher for his guidance, advice and assistance throughout all of the stages of this project. I would also like to thank Dr. F. Mills of the Department of Mechanical Engineering for his assistance with the Mechanical Engineering side of things, and Andy Sherlock at the Department of Mechanical Engineering for his invaluable advice and his willingness to help with any and all aspects of the project.

During the past three years Andrew Tucson has been an invaluable source of useful knowledge, good advice and friendship, and for these gifts I wish to express my sincere appreciation.

Finally I want to thank Julie Humphrey who has constantly reassured me that the effort is worthwhile and without whom it might not have been.

QUESTION 1

The following table shows the results of a survey of 100 people about their favourite sport. The table is partially completed. Complete the table by calculating the missing values. Write your answers in the boxes provided.

Sport	Male	Female	Total
Football	45	30	75
Tennis	20	15	35
Swimming	10	10	20
Other	5	5	10
Total	80	60	140

Contents

1	Introduction	1
1.1	Introduction to the problem	1
1.2	Approach taken	2
1.3	Contributions made by this work	5
1.4	Outline of this thesis	5
2	Background and previous work	8
2.1	The problems to be addressed	8
2.2	The genetic algorithm	11
2.2.1	Other search techniques	12
2.2.2	Advantages of the genetic algorithm	15
2.2.3	The standard genetic algorithm	16
2.3	Introduction to shape optimisation	19
2.3.1	The stress analysis model mathematics	20
2.3.2	Possible approaches to shape optimisation	21
2.3.3	The voxel based approach to shape optimisation	22
2.4	Miscellaneous relevant work on GA based shape optimisation	24
2.5	Conclusions	25
3	Design and testing of new operators	27
3.1	The one-dimensional experiments	27
3.2	A two-dimensional representation of a beam cross-section	31
3.3	The GA experiments	32
3.3.1	Experiments using the naïve GA	33
3.3.2	Results obtained from the naïve GA	33
3.3.3	Experiments with the rank selection pressure	36
3.3.4	Experiments with the constraint penalty multiplier	40
3.3.5	Experiments with a dynamically changing constraint penalty multiplier.	46
3.3.6	A new operator: smoothing	53
3.3.7	An n-dimensional crossover operator	56
3.3.8	An n-dimensional mutation operator	60
3.3.9	Dynamic mutation operator probabilities	63
3.3.10	Comparison of modified GA with naïve GA	66
3.4	Conclusions about the simplified physics model experiments	68
4	Transferring the solutions to real problems	71
4.1	Motivation for this approach	71
4.2	The annulus design problem	72
4.3	Implementation details	73
4.4	Results from the basic system	77
4.5	Improvements made to the system	82
4.6	Conclusions about the real-world system	90

The following text is a scan of a document page. It contains several paragraphs of text, which are mostly illegible due to extreme blurring. The text appears to be organized into sections, possibly numbered or bulleted, but the specific content cannot be discerned. The page number '1' is visible in the top left corner.

5 Conclusion	92
5.1 Summary of conclusions	92
5.2 Further work	94
Appendices	95
A Rolls Royce annulus problem specification	95
B Ansys macro listings	99
B.1 Full optimisation macro mask listing	99
B.2 Single optimisation macro listing	109
C Results log files	110
C.1 Log of basic GA results on annulus problem.	110
C.2 Log of improved GA results on annulus problem	116
Bibliography	123

10
11
12
13
14
15

16
17
18
19

20
21
22
23

24
25
26
27

28
29
30
31

32

Chapter One.

Evolutionary Shape Optimisation Using a Voxel Based Representation.

1.1 Introduction to the problem.

Shape optimisation is the process of attempting to discover the optimum shape to perform some given task, frequently within some pre-determined constraints. In some simple cases the solution may be readily apparent, even trivial, for example; to maximise volume and minimise surface area the shape required is a sphere. However in most cases the solution is not so easily found. Engineering knowledge and a sound understanding of the underlying physics of a problem may enable a well-trained person to make a good guess as to a suitable form, but this leaves open the possibility of major improvements being totally overlooked and may not be sufficient if the problem being addressed is very highly constrained.

This thesis examines two different shape optimisation problems and will attempt to show that the combination of genetic algorithms with suitable operators and a voxel based representation, forms an effective tool for solving this type of problem.

The first optimisation addressed in this thesis is that of minimising the amount of mass in load-bearing beams, whilst ensuring that the maximum stress they are capable of withstanding remains above a given minimum level. This is tackled first by means of a mathematical model which contains some simplifying assumptions about the physics of the situation. Although this is not very realistic in terms of a real-world analysis, it has the advantage of being very quick to run which makes it a suitable test-bed for experimentation with the operators and the parameters of the GA being used.



The second optimisation is that of designing a minimum mass annulus for a jet-turbine engine which must withstand several different forces without exceeding the maximum stress limits in any one of several key areas. This problem does not have any simplifying assumptions, and so is analysed in full by a commercial Finite Element analysis package; Ansys 5.3^{*}. As the evaluation function is very time-consuming it was necessary to design several new GA operators which increase the rate of improvement considerably.

Previous work in this field includes:

- [Bentley 96] which has successfully evolved a wide range of shapes and structures using geometric primitives as building blocks.
- [Chapman *et al.* 94] which addresses issues in structural topology design using pixels as design primitives with a standard GA.
- [R Smith 95a] which uses a GA with a parametric representation to optimise the same annulus problem covered in Chapter 4 of this thesis.

A more detailed description of shape optimisation and stress analysis will be presented in Chapter Two.

1.2 Approach taken.

The representation of the problem space chosen for this work is that of voxels⁺, which are a method of partitioning the search area or volume into rectangular regions or boxes which are then assigned a binary full/empty value.

^{*} Ansys is a registered trade mark of SAS IP, Inc.

⁺ *N-dimensional* pixels

1. The first part of the document discusses the importance of maintaining accurate records of all transactions. This is essential for ensuring the integrity of the financial statements and for providing a clear audit trail. The records should be kept up-to-date and should be easily accessible to all relevant parties.

2. The second part of the document outlines the various methods used to collect and analyze data. This includes the use of surveys, interviews, and focus groups. Each method has its own strengths and weaknesses, and it is important to choose the most appropriate method for the specific research objectives. The data collected should be analyzed carefully to identify any trends or patterns.

3. The third part of the document discusses the importance of maintaining accurate records of all transactions. This is essential for ensuring the integrity of the financial statements and for providing a clear audit trail. The records should be kept up-to-date and should be easily accessible to all relevant parties.

4. The fourth part of the document outlines the various methods used to collect and analyze data. This includes the use of surveys, interviews, and focus groups. Each method has its own strengths and weaknesses, and it is important to choose the most appropriate method for the specific research objectives. The data collected should be analyzed carefully to identify any trends or patterns.

5. The fifth part of the document discusses the importance of maintaining accurate records of all transactions. This is essential for ensuring the integrity of the financial statements and for providing a clear audit trail. The records should be kept up-to-date and should be easily accessible to all relevant parties.

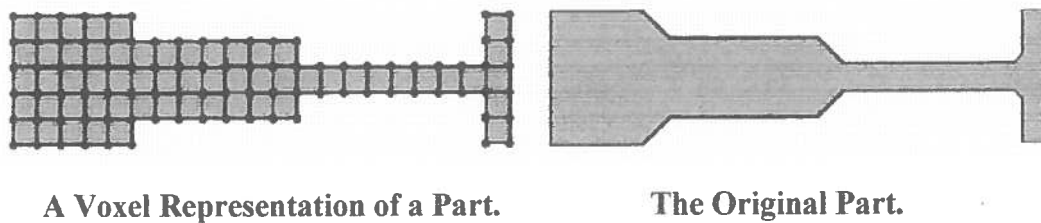


Figure 1.1

This approach has many advantages over the alternatives including:

- The ability to represent any shape without restriction.
- The ease with which an existing engineering solution can be converted into voxels.
- A natural mapping to the traditional binary chromosomal representation frequently used by GAs.
- The simplicity with which domain knowledge can be integrated into the solution.

However in [Watabe & Okino 93] the authors object that voxels suffer from several apparently fatal weaknesses when used to represent this type of problem including:

- The occurrence of small holes in the final shape.
- The long length of the chromosomes.
- The expectation that cross-over would be ineffective.
- The lack of smoothness in the final shape's outline.

It will be shown during the early experiments described in Chapter 3 that these objections do not constitute a problem when addressed with appropriate operators and GA optimisations.

The choice of voxels as a representational form was motivated by several key points. The previous work which has made use of voxels has tended to either avoid



using domain knowledge in the operator design, or has concentrated on very low resolution voxel grids in an attempt to avoid huge chromosomes. Voxels also offer some important advantages over other representations, including the ability to generate unexpected designs, the ease with which the user can initialise the population to the previous best solution and quick incorporation of domain knowledge in the form of areas of voxels with fixed values.

Genetic Algorithms (GAs) have proven themselves to be a highly flexible search technique, able to find good solutions to a diverse range of problems in many different fields [Goldberg 89]. They use evolutionary computation techniques to search large and difficult fitness landscapes in order to find near-optimal solution vectors.

[Michalewicz 92] describes “evolution programs” as a sub-class of the range of algorithms which rely on analogies to natural processes. In [Michalewicz 92] he writes:

“... a population of individuals (potential solutions) ... undergoes a sequence of unary ... and higher order ... transformations. These individuals strive for survival: a selection scheme biased towards fitter individuals selects the next generation. After some number of generations, the program converges - the best individual hopefully represents the optimum solution.”

In this thesis the term GA will be substituted for Michalewicz’s “evolution programs” to reflect the fact, widely recognised in the field of evolutionary computation, that the original definition of GA (as using only binary chromosomes, simple crossover and bitwise mutation) is far too narrow and explicit to encompass this rapidly growing field.

As the nature of the voxel representation leads inevitably to extremely large search areas the use of GAs is an obvious choice. A further introduction to the theory of GAs can be found along with a more detailed description of shape optimisation, stress analysis and the voxel based representation in Chapter 2.



1.3 Contributions made by this work.

During the first set of experiments a number of new and modified operators were designed and tested, and some of them proved to be extremely effective at reducing the amount of time taken for the GA to achieve a good solution. Several other operators were found which directly address the issues raised by [Watabe & Okino 93] and solve the problems described by them.

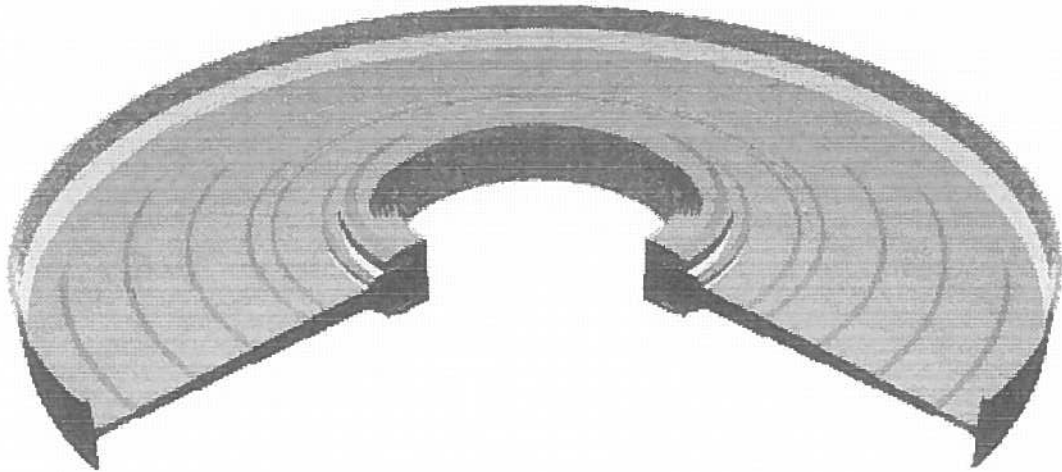
The second set of experiments extends the range of previous work in this domain to use a commercial FE package to perform the optimisation. Two sets of experiments were performed, firstly a simple adaptation of the GA to the new problem was tested. Although these experiments did discover some difficulties when using Finite Element Analysis with a voxel based representation, further work resulted in the design of a set of solutions to those difficulties and some very interesting results were obtained from the analysis.

The work in this thesis will show that a genetic algorithm using a voxel based representation, whilst suffering from several difficulties, is by no means an implausible approach to the problem of shape optimisation. The benefits obtained by using this representation may in many cases out-weigh the disadvantages, thus making it an appropriate technique for optimisations where the optimum is unknown or where engineering insights are to be used to assist the process.

1.4 Outline of this Thesis.

The task was addressed as a series of experiments, each of which built on the results of the previous steps; the format of this thesis reflects this approach, with the majority of the work being presented in the order of experimentation and with appropriate comments where one set of experiments suggested the approach to be taken next.





Final Annulus Design from the Results Set.

Figure 1.2

The majority of Chapter 2 deals with the background explanations of GAs, shape optimisation and stress analysis necessary to understand the rest of the thesis. An investigation was then made into the current state of the art in shape optimisation, with particular reference to the use of GAs to perform the search. The results of this study are presented in Section 2.3.

Initially, a two dimensional shape optimisation problem was undertaken using simplifying assumptions in the stress analysis model. This allowed rapid evaluations of the shapes produced by the GA, thus permitting in-depth investigation into the effectiveness of various modifications to the basic optimisation technique. A description of these experiments, their results, and the optimisations made can be found in Chapter 3.

Once the GA code was running effectively on this simplified problem, the level of realism in the stress analysis model was increased by using a commercial Finite Element package to perform the shape evaluation. At this point the experiments were using a realistic representation of all stresses involved in the



shapes generated by the GA. Chapter 4 gives details of the approach taken, the results and the difficulties encountered during this final stage of the experimentation.

Chapter 5 contains the conclusions arrived at during this study and the recommendations for future work.

The Appendices include details of the annulus design problem specification, the macros used to make Ansys perform the optimisation and some results data files.



Chapter Two.

Background and Previous Work:

2.1 The problems to be addressed.

The first optimisation addressed in this thesis is the design of a beam cross-section which minimises the amount of material in the beam, whilst leaving it capable of supporting a given minimum mass. This problem will be approached using a simplified model of the physics involved in order that the speed of evaluation is as fast as possible. The second optimisation is the design of an annulus for the centre of a jet-engine's turbine propeller, which must be as light as possible but still be able to withstand stresses in two different directions at various key positions. This will be evaluated using a full commercial Finite Element Analysis (FEA) package in order that all of the forces and stresses experienced by the part are simulated as accurately as possible.

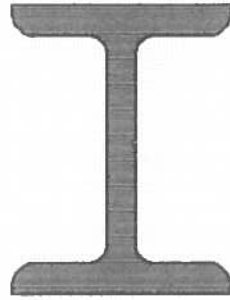
The beam optimisation was chosen as an example of a sufficiently hard problem that, even when using a simplified model of the physics involved, the solution surface would be extremely complex. An additional reason for the choice of this problem is that a good solution is well-known to engineers. The 'I' beam (shown in Figure 2.1), which is commonly used in sky-scraper construction, is generally accepted as being a good form for a beam cross-section to gain maximum strength for minimal mass.

Issues to be addressed in this optimisation were;

- How well does a standard GA perform on a complicated problem using a voxel representation?
- Can operators be designed that overcome any perceived flaws in the standard GA?



- What alternative operators can be designed to improve the GA performance in regards to voxel based optimisations?
- What adjustments should be made to the balance of operators to further increase the rate of optimisation?



An 'I' Beam Cross-Section.

Figure 2.1

The annulus design problem is specified in detail in a document supplied by Rolls Royce to Richard Smith during his time with the Mechanical Engineering department of Edinburgh University, which is included as Appendix A. As specified by Rolls Royce, the annulus design is very tightly constrained, ie: the figures given for maximum stress represent a reasonably good solution. This was expected to cause some difficulties as a random population initialisation would probably create an initial population which contains only invalid solutions.

The choice of a voxel based representation was expected to raise some difficulties when the solutions produced were analysed by the FE package. This is because FEA does not cope well with sharp corners and will normally not use uniformly sized meshes. Voxels are by definition uniform in size, and though it would in theory be possible to subdivide edge voxels into a finer grid, the FE package used for this analysis (Ansys v5.3) could only recognise elements which



connect at their major vertices. Thus in order to subdivide the voxels into a finer grained mesh would involve at least one intermediate step and a lot of extra processing, as seen in Figure 2.2.

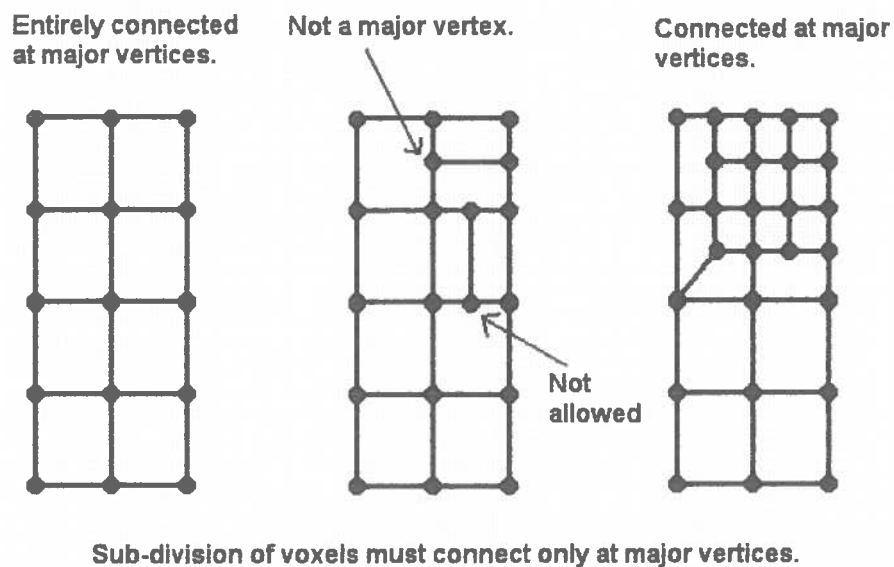


Figure 2.2

One further issue was of concern at the beginning of the experimental work described here; what would be the best way of getting Ansys 5.3, a commercial, stand-alone, GUI driven FE package, to perform the analysis of the GA produced voxel shape representations? This was a particularly worrying problem as all of the work was performed on PC compatible machines, which are not noted for the flexibility of their inter-process communications systems.

These problems and their solutions are described in detail in Chapter Three for the beam and Chapter Four for the annulus optimisation.



2.2 The genetic algorithm.

'Genetic Algorithm' is the term which has come to cover the use of evolutionary directed algorithms for searching complex parameter spaces to find a near-optimal solution. This technique requires the formulation of a suitable 'fitness' function which can evaluate any solution possible within the representation chosen and return a valid fitness score. The search is performed by maintaining a population of potential solutions and allowing them to interbreed (crossover), mutate and die. The selection of survivors from one generation to the next is based on each individual's fitness score relative to the population as a whole, thus those individuals with better fitness scores have a higher probability of surviving and breeding in the next generation.

The sequence of data which represents an individual member of the population of solutions is called a chromosome, taking its name from the DNA sequence in humans as this whole technique is based on an analogy to nature's evolutionary processes. Each item of data in the chromosome is called a gene following the same naming convention. Genes in a traditional GA are single bits, however floating point numbers and more complicated data types have also been used [Michalewicz 92] (pages 2-9) and [Davis 91].

Crossover is the process of taking some genes from one 'parent' and swapping them with the genes in the same position on the chromosome of the other parent. This has the effect of propagating short gene sequences throughout the population and as the chromosomes are more likely to reproduce if they are fit, only sequences which make chromosomes fitter will spread through the population. Mutation is the method whereby new genetic variations can enter the population and generally consists of randomly changing the value of a small number of randomly chosen genes.

A more complete description of the theory and implementation details of conventional genetic algorithms can be found in [Holland 75], and [Michalewicz 92] (Chapter Five) includes details of some of the possible modifications to the basic approach which extend the power and capabilities of this optimisation technique.



2.2.1 Other search techniques.

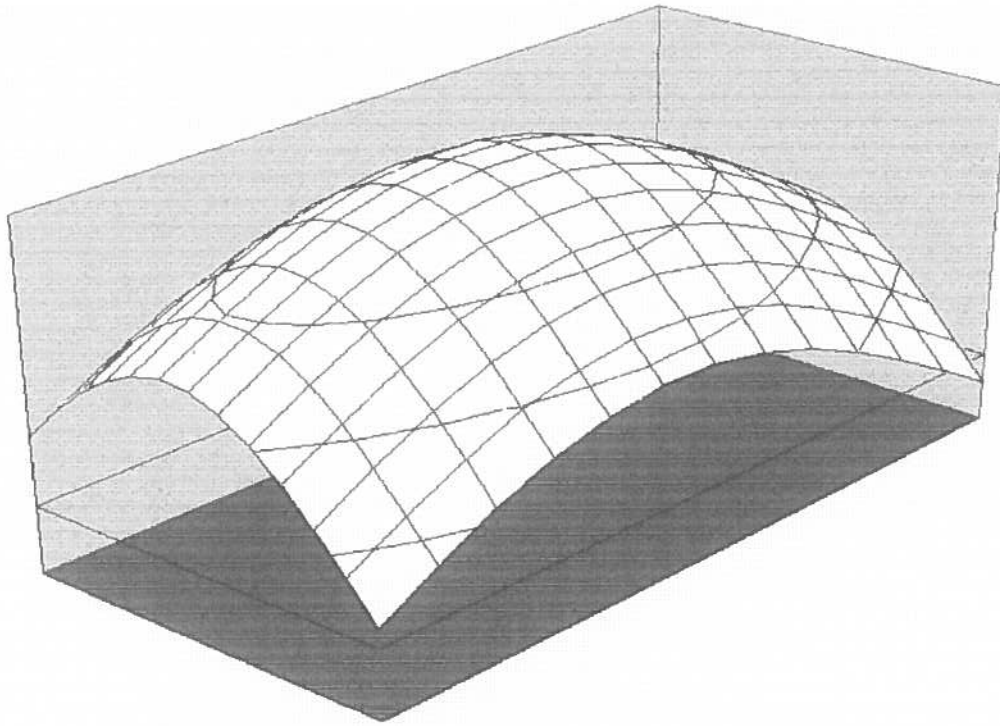
Exhaustive search is the procedure of iteratively supplying an evaluation function with every single parameter setting within the scope of the problem being studied. It is impossible to do this in a reasonable amount of time for most real-world problems due to the range and the number of parameters which define the problem. In a voxel based representation of a beam using a two-dimensional grid divided into 32x64 voxels, the total problem space is 2^{2048} !

Random search involves iteratively trying different randomly selected parameter settings whilst always remembering the best solution found. For large problem spaces this is ineffective due to the small probability of randomly choosing good parameter settings, this is especially true when a large portion of the solution space constitutes invalid solutions due to violations of the problem constraints.

Stochastic hill-climbing (SHC) is an optimisation technique based on the concept of hill-climbing. In a well-behaved function optimisation problem, the solution space may often be visualised as a smooth surface in multiple dimensions as shown in Figure 2.3. Given a randomly selected initial parameter setting, a steepest-ascent hill-climbing algorithm will evaluate the solution function and find a single point on the surface. By varying the problem parameter settings and re-evaluating the function, other points can be found which will either be worse, the same or better than the previous solution. Repeatedly varying the parameters and keeping the best solution from a set as the starting point for the next iteration will result in continual progress up the hillside, when no solution is better than the current starting point the algorithm has achieved an optimal solution.

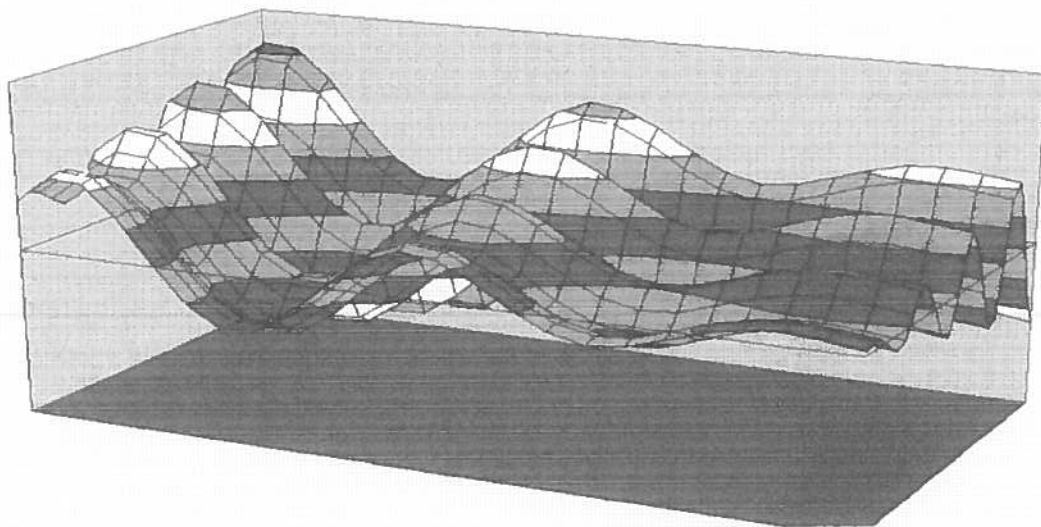
The type of problem which can be solved by hill-climbing in this simplistic formulation is highly limited. Most real-world problems involve far more complex solution surfaces with many bumps and pits which represent local optima as seen in Figure 2.4.





A Smooth Two-Dimensional Solution Surface.

Figure 2.3



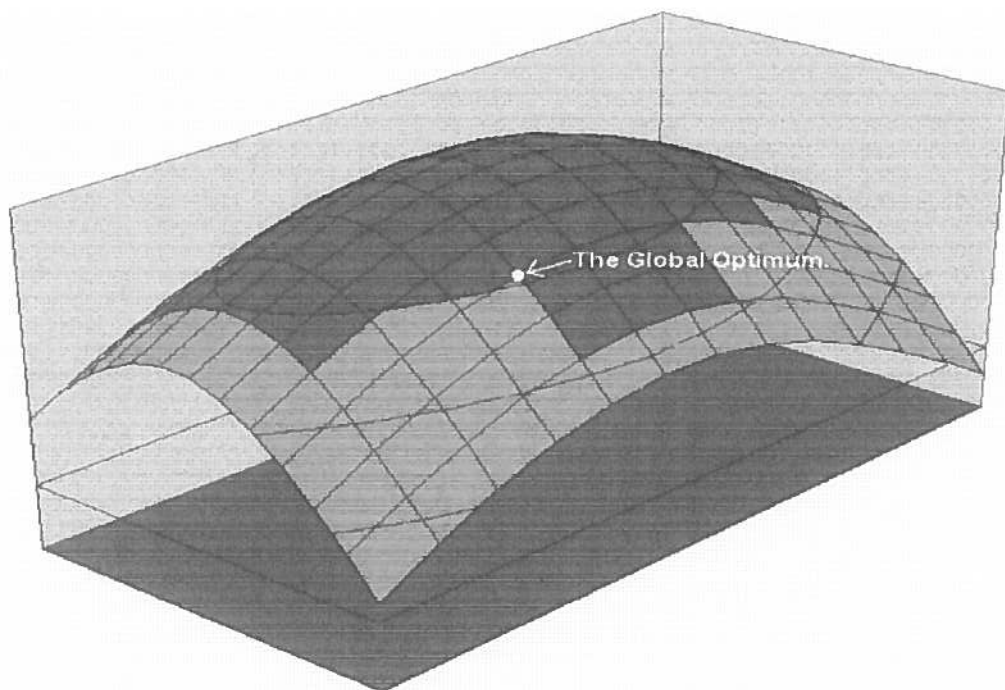
Uneven Solution Surfaces May Have Many Local Optima.

Figure 2.4



SHC attempts to address these more complicated problems by adding the capability to randomly reinitialise the current starting point when no more improvement is possible from the position found. The best solution found is continually recorded, so that after a given number of iterations the SHC can report the parameter settings which produced that solution.

A major problem with SHC is that when a solution space with a very large number of local optima is faced, the chance of a random re-initialisation of the parameters producing a solution which is on the same hill as the global optima is very small. Another problem is encountered when constraints are applied to the solution space. Figure 2.5 shows a very simple optimisation problem which has been complicated by the addition of a constraint limitation. SHC can only find the optimum for this problem if it starts somewhere on the line going straight up the hill through it. From this it is apparent that the probability of SHC finding a good solution is reduced drastically by the addition of problem constraints.



Constraint Limitations Cause Difficulties For Hill Climbers.

Figure 2.5



Simulated annealing is an approach to optimisation which comes from an analogy with the annealing process used to toughen materials such as metals or glass. The process uses heating, followed by a long, slow cooling period to alter the molecular structure of the materials being treated. Similarly the optimisation technique permits acceptance of lower quality solutions with a probability dependent on a 'temperature' variable which gradually decreases over the length of the run. This technique is capable of escaping local minima in the search space and will therefore find a good solution to many optimisation problems. Although it does not suffer from any of the major drawbacks in the other techniques discussed previously, simulated annealing does require great care in the establishment of the initial and final temperatures and the rate of cooling.

2.2.2 Advantages of the genetic algorithm.

Genetic Algorithms provide a good balance between exploiting the best solutions and exploring the search space [Michalewicz 92] (page 15) which gives them a huge advantage over random search (all exploration) and hill climbing (all exploitation). The cross-communication of information (via cross-over operators) within a GA [Michalewicz 92] (page 3) implies the possibility of continual improvement over the entire search time period. This constitutes an important advantage over the random reseeding used in SHC. GAs are also ideally suited to parallel computation because each member of the population can be processed simultaneously [Michalewicz 92] (page 9). This implies the ability to gain considerable processing speed improvements through a parallel implementation of the algorithm, which is not possible with simulated annealing as it is an iterative algorithm with cumulative result variables.

A major advantage of the GA is the relative ease with which new operators can be formulated and added to the existing selection. This is primarily due to the total separation of operators from one another. To add a new operator which directly



addresses a perceived difficulty with the optimisation is simply a matter of designing an algorithm which will fix the problem if it is applied in a suitable area, and will do nothing (or act to prevent the problem occurring) if applied elsewhere. After a short period of experimentation it is usually possible to find an appropriate probability setting for the new operators - usually by trying settings 10 or 20 percent apart.

This ease of design for new operators permits incorporation of domain knowledge which can greatly increase the search efficiency, in the same way that heuristics can improve a standard depth-first or breadth-first search.

As the voxel representation selected for this thesis codes directly into a very long sequence of binary digits, the solution space is highly uneven and the problem is very tightly constrained; Genetic Algorithms are therefore the natural choice of search algorithm for this problem.

2.2.3 The standard genetic algorithm.

The basic GA used as a foundation for all of the experiments outlined in this thesis varies somewhat from the GA described by [Holland 75]. Many of the variations were introduced to maximise the speed of operation as it was realised that the optimisations being performed would be very time intensive and that even small improvements in the run-time would be useful.

The GA overall logic is shown below in Figure 2.6 and at this level of detail the algorithm is very similar to that included in [Michalewicz 92] as Figure 0.1.



```

GA control logic
  begin
    initialise population
     $pass \leftarrow 0$ 
    while  $pass < number\_of\_passes$  do
      Evaluate population
      Display population
      Select survivors for next generation
      Apply operators to new population
       $pass \leftarrow pass + 1$ 
    end while
  end

```

Control logic for the standard Genetic Algorithm.

Figure 2.6

Initialisation is performed entirely by random numbers, with every gene in each chromosome being set to either a one or a zero. These chromosomes map directly into the voxel grid, where a one represents the voxel is occupied and a zero indicates an empty space. After the chromosome is initialised in this way, a connectivity check is performed and all voxels which are not connected by a chain of active voxels to a given ‘seed point’ are set to empty space. This eliminates all of the loose or floating voxels and can lead to an invalid chromosome if the two ‘seed points’ at the top and bottom centre of the grid are not connected to each other - if this is the case then the initialisation is performed again.

Evaluation uses the stress analysis values returned from the function described in Section 2.3.1 to calculate a ‘fitness’ value for each chromosome according to the following formula (Equation 2.1).



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

where:

F	= Fitness
V	= Count of active voxels
S	= Maximum stress of any voxel
k	= Constraint penalty multiplier
σ_{max}	= The stress constraint value

The Fitness Formula.

Equation 2.1

This formula must be minimised and can be interpreted as meaning that a greater number of active voxels implies a higher fitness value. A larger value of maximum stress also causes a higher fitness value (though this value will never exceed the contribution provided by even a single extra voxel). The rest of the formula is zero unless the stress exceeds the constraint, at which point it will increase the fitness value by an amount proportional to the amount by which the constraint has been broken.

The display routine is included so that the progress of the GA can be visually monitored as the optimisation progresses, it would also be a useful feature if user-guided analysis was added to the capabilities of the program at a later stage of development.

Survivor selection is achieved through use of the *Genitor* rank selection algorithm described in Section 3.3.3. This algorithm has several advantages over fitness proportionate selection methods including the removal of the need to scale the fitness values returned from the evaluation algorithm and the ease with which the amount of selective pressure can be altered through variation of a single parameter. A full discussion of this approach can be found in [Whitley 89]. The rank selection procedure gives 'fit' individuals a better chance of surviving into the next generation than unfit individuals by using the fitness values to create an ordered list of the population and then returning randomly selected indices to that list with a



progressively higher probability towards the 'fitter' end of the list. It is used to generate a new population of selected chromosomes which replaces the previous population after the selection is complete.

The operators used in the standard GA were two-point crossover and bitwise (or *uniform*) mutation [Michalewicz 92] (page 21) however the algorithm was designed to allow the addition of new operators and the replacement of the old ones with a minimum of code alterations. Two point crossover involves randomly choosing two 'parent' chromosomes, selecting any two genes at random along the length of the chromosome and then taking the sequence of genes between those two points and swapping it with the same gene sequence from the other 'parent'. Bitwise mutation simply alters the value of randomly selected genes in randomly selected chromosomes.

Unless mentioned otherwise the GA used the standard settings described below in the experiments that follow.

Mutation Rate	= 0.001 per bit
Crossover Probability	= 0.3 per chromosome
Population Size	= 20
Ranking Pressure	= 1.5
Chromosome Length	= 2048 bits

2.3 Introduction to shape optimisation.

A worthwhile problem from the area of Mechanical Engineering is that of optimising a beam to support various loads with a minimal amount of material. Currently these optimisations are frequently done by hand, by a person with long experience, who will then often use a Finite Element package for stress analysis to ensure that the resulting beam meets the various requirements.



This section presents an introduction to the theory and mathematics required for stress analysis, lists some previous work in the area of shape optimisation and describes the differences between that work and the approach taken here.

2.3.1 The stress analysis model mathematics.

A simplified stress analysis model requires very little mathematics, is quite easy to implement and can be used to generate some interesting results. The mathematics for the model used in the early experiments is shown as Equation 2.2.

$$\sigma = - \frac{My}{I}$$

Where:

- σ = total stress within a given area
- M = bending moment (amount of force being applied to the beam)
- y = distance of the area from the neutral axis of the shape
- I = second moment of area
= $\int y^2 dA$
- A = size of the area being analysed

Equation 2.2.

The *neutral axis* of a shape is defined as a horizontal line which passes through the centroid of mass of the shape. As a voxel representation uses areas which are all of uniform size and density, the centroid of mass can be found by taking the average of the positions of all occupied voxels.

The second moment of area is approximated by calculating Equation 2.3.

$$I \approx \Sigma (y^2A)$$

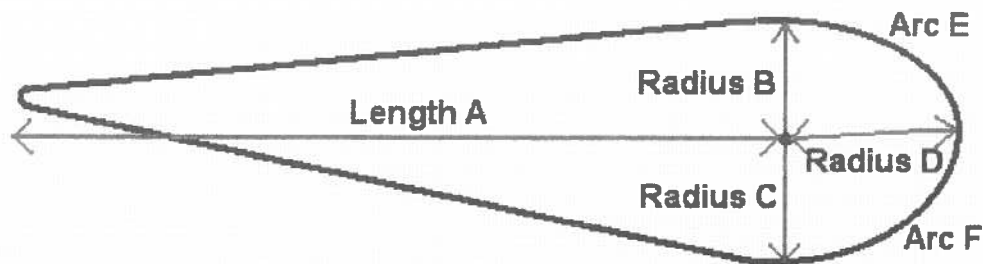
Equation 2.3.



2.3.2 Possible approaches to shape optimisation.

There are many possible approaches to shape optimisation problems, in this section a few of the more common techniques will be briefly described.

The parametric approach to shape optimisation involves creating a parameterised model of an initial shape. Some of the parameters can be constants, which allows the designer to apply domain knowledge to reduce the size of the search space. Other parameters will be allowed to vary due to the optimisation process. These are normally restricted to a range, defined as the maximum and minimum values that are permitted. Figure 2.7 shows a parameterised representation of an aircraft wing cross-section.



A Parameterised Wing Cross-Section.

Figure 2.7

The cell-growth method of shape optimisation uses an analogy with biological cellular growth to evolve suitable shapes. The cells are given 'growth rules' when they are created, and then when they develop according to those rules the resultant shapes are evaluated for their fitness values. The growth rules are treated as chromosomes and the search is performed using a GA to optimise the rules. Some dynamic feed-back during the growth process is possible using a variation on this approach which involves continually evaluating the developing



shapes. If an area is suffering from very high stress for example, then more cells can be encouraged to grow in that area by adjusting the local growth parameters.

Primitive building blocks have been used by [Bentley 96] to evolve complex shapes using his Generic Evolutionary Design system. This system uses ‘clipped, stretched cuboids’ as the spatial partitioning representation and a GA performs the search by modifying genotypes which represent each individual design. This technique has enjoyed some great successes and has been used to evolve designs for tables, steps, heat sinks and many other components.

2.3.3 The voxel based approach to shape optimisation.

Previous work on shape optimisation has primarily concentrated around parametric representations of simple machineable shapes and has produced some very useful results [Smith 95a]. The parametric representation, which uses a series of parameters which are varied to alter the shape being represented, is a common choice for shape optimisation problems due to the simplicity with which restrictions on the shapes can be incorporated. As noted previously, unrestricted shape optimisation representations will frequently lead to ‘breaking’ the evaluation function and so parametric representations are often used to prevent this from happening.

This project uses a voxel based space representation which divides the shapes being optimised into a uniform grid of rectangles or boxes. Each subdivision is represented by a binary value which indicates whether the voxel is empty or full. Similar work has been done by [Chapman *et al.* 94] and [Shoenuer 95] however [Chapman *et al.* 94] used only standard GA operators and so suffered from the problems raised by [Watabe & Okino 93], and [Schoenuer 95] became concerned about the size of the chromosomes required to represent detailed or real-world problems and moved into alternative forms of problem representation.



Voxels may have any dimensionality, however representations of more than three dimensions are not required for the approach to shape optimisation taken here. Early experiments were performed in both one and two dimensions in order to explore the possibilities and gain insights into the problem area and the operation of the Genetic Algorithm.

The binary voxel representation leads naturally to very long chromosomes and the problems concomitant with the huge search spaces that they impose. However it also allows far more complex and varied shapes than a parametric representation - which is limited by the user's choice of what aspects of the representation will be controlled through the parameters. In particular, holes in the material are entirely possible and even probable if there is a small low-stress area in a high-stress region - a parametric approach can only create holes where the user is expecting them to be required and has defined some appropriate 'hole-control' parameters to represent them.

Voxels are limited in that they can only form an approximation to a curve, however as the number of subdivisions used is increased, the approximation becomes more accurate and this can easily be controlled to suit the user's accuracy requirements (Figure 2.8). There is a trade-off to be made in this situation however; the higher the number of subdivisions, the longer each individual evaluation will take. Thus the user must attempt to set the level of accuracy to be just sufficient if the amount of time taken by the analysis is also a consideration.



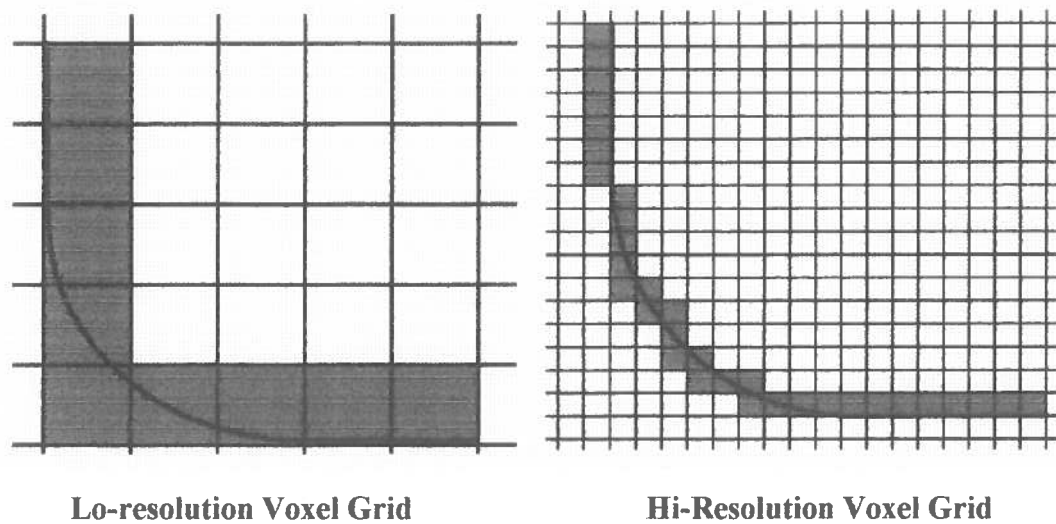


Figure 2.8

2.4 Miscellaneous relevant work on GA based shape optimisation.

[Watabe & Okino 93] describe shape design and optimisation by use of Free-Form Deformations (FFD) which use a matrix of control points to deform a default shape. This work includes three criticisms of a voxel based representation method for shape optimisation problems:

- Long chromosomes.
- Creation of small holes.
- Valid parents don't often make valid children.

The crossover technique used in this work is to take a sub-set matrix of the control points and swap with the other parent. Three sets of experiments are described and the results are very good. The third set of experiments uses a Finite Element Method package in an attempt to minimise mass for a given maximum stress.



The paper: [Chapman *et al.* 94] covers genetic algorithms for shape optimisation under stress constraints. They use a one-point crossover claiming better results than from uniform crossover, a result which will be confirmed by the experiments detailed in Section 3.3.7. The work includes a simple rule for avoiding isolated pixels in the final representation by requiring a level of connectivity (4 directional) to a given 'seed' pixel, a concept which was adapted to the experiments performed here and extended to require two 'seed' voxels, one at each end of the voxel grid. They also assert that balancing a penalty parameter (for beams that fail the stress constraint) is critical for obtaining good results, as too high a penalty will produce beams that fall too far within the constraint (with the corresponding extra material this entails). This result is again confirmed by the experiments detailed in Section 3.3.4.

In [Bentley 96] a description of a very powerful *generic* shape optimisation tool is given. This system uses clipped, stretched cuboids as a primitive which is used to partition space into occupied/void areas. The results are very impressive and the system has developed designs for a large number of different parts and components. The emphasis is very much on the generality of the method, and it includes a technique which allows the user to easily specify the evaluation function from a library of already developed evaluation programs.

2.5 Conclusions.

This chapter has introduced the basic theories behind genetic algorithms, shape optimisation and stress analysis, and has described the problems to be addressed in later chapters. Previous work in this field has tended to concentrate on parametric representations and this has evolved good designs for aircraft wings and other components. The voxel based approaches to this problem have also been successful with work by both [Chapman *et al.* 94] and [Schoenauer 95] being done on a side-view beam topology optimisation problem with interesting results.



The rest of this thesis will describe the investigations undertaken on two separate problems from the field of shape optimisation using a voxel based representation and a modified GA to perform the search.

The first investigation will deal with an artificially simple example in order to experiment with GA operators with the intention of designing a selection of good, new, general purpose operators that perform well in a voxel based design space. The second investigation is intended to take those operators and the knowledge gained about the GA, and try to apply it to a real-world optimisation problem.

The operators designed and described in Chapter Three were also intended to be easily adaptable to a three-dimensional voxel representation, however this intention has yet to be proved.

This work will necessarily be restricted to just these two optimisations due to time limitations, however as the investigations have been successful, future work could extend the discoveries to virtually any two-dimensional shape optimisation problem which can be evaluated by a FEA package. This includes stress, loading, heat, liquid flow and electro-magnetic field optimisations.



Chapter Three.

Design and Testing of New Operators:

The techniques for the design of effective Genetic Algorithms are many and varied [Holland 75][Davis 91][Michalewicz 92] and a great deal of time can be spent adjusting parameters and trying different operators to obtain the most effective balance between convergence on a good solution and exploration of the problem domain. The stress analysis GA was initially used to evolve results for simplified versions of the problem area in order that the optimisations suggested by early results could be investigated rapidly. The simplified physics model of the beam's stresses uses the mathematical approximations described in Section 2.3.1 in order to permit rapid calculation of the fitness of each individual in the population.

The work described in this chapter details early experiments with a one-dimensional chromosome followed by the extension of the experiments to two-dimensions and the subsequent GA optimisations. A comparison is then made between the results gained from the naïve GA and the results made possible following the addition of new operators.

3.1 The one dimensional experiments.

The first experiments were performed on a one-dimensional chromosomal representation of a beam cross-section. This very simple case was chosen as the starting point for the experiments as the expected results are well understood [Gere & Timoshenko 84] (pages 251-261) so it permits cross-checking of actual results gained against expected results. The programming involved in the Genetic Algorithm was not very complicated, and neither was the implementation of the

The Role of Ethics in Business

The role of ethics in business is a topic that has gained significant attention in recent years. As businesses grow and expand, the ethical implications of their actions become increasingly important. This paper explores the various ways in which ethics can influence business success and the responsibilities of business leaders in this regard.

One of the primary ways in which ethics influences business is through the reputation of the company. A company known for its ethical practices is more likely to attract customers, investors, and top talent. Conversely, a company with a poor ethical reputation may face financial difficulties and legal challenges. Therefore, maintaining high ethical standards is essential for long-term business success.

Business leaders also have a responsibility to ensure that their actions align with ethical principles. This involves making decisions that are fair, transparent, and respectful to all stakeholders. By doing so, leaders can build trust and loyalty among their employees and customers, which is crucial for the company's growth and sustainability.

In addition, ethics can play a role in shaping the corporate culture. A strong ethical culture can lead to higher productivity, better teamwork, and a more positive work environment. This, in turn, can result in a more competitive and resilient organization. Therefore, it is important for business leaders to prioritize ethics in their management practices.

Conclusion

In conclusion, the role of ethics in business is a complex and multifaceted one. It is not just a matter of following rules, but of cultivating a strong ethical foundation that guides all business decisions. By prioritizing ethics, business leaders can ensure that their companies are not only successful in the marketplace but also contribute positively to society.

simplified stress analysis physics model being used at this time, however it seemed reasonable to take this opportunity to verify their operation together.

The material and force parameters used for this initial set of runs were:

Area Represented	= 0.75m ²
Maximum Stress Allowed	= 100 000 000 Pa (Steel)
Bending Moment	= 200 000 Nm (Force applied)

The GA used settings of:

Mutation Rate	= 0.005 (per bit)
Crossover Probability	= 0.25 (per chromosome)
Population Size	= 60
Ranking Pressure	= 3
Chromosome Length	= 64 bits

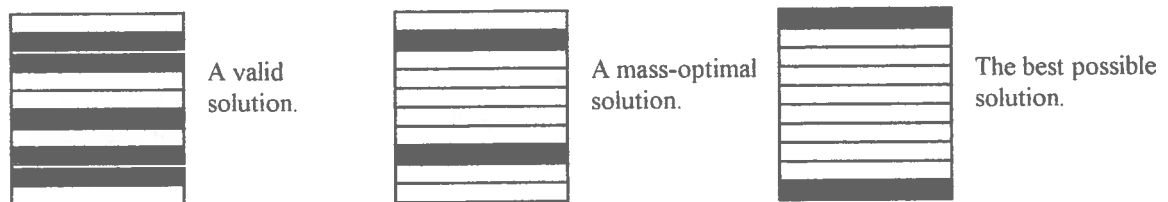
The mutation operator was the standard *uniform mutation* [Michalewicz 94] which flips bits in the chromosomes with each bit having an equal probability of being altered. The crossover operator used was also standard; the *uniform crossover* [Michalewicz 94] gives every bit in one parent's chromosome a set probability of being swapped with the corresponding bit from the other parent. The GA used a *roulette wheel* selection mechanism with *standard ranking* (refer to [Whitley 89] for a good description of the most common selection mechanisms and their influence on diversity and convergence). An *elitist mechanism* was also implemented whereby the best solution found in any generation was guaranteed to survive unaltered into the next generation.

The fitness function described in Chapter Two as Equation 2.1 was designed to minimise the area of the beam cross-section (number of active voxels) whilst remaining capable of withstanding the maximum allowed stress, with a penalty value



being applied to any solution which broke that strength constraint. A small additional factor was included in the fitness calculation based upon the maximum stress point in the beam. This has the effect of causing the valid solutions to continue to evolve towards the best possible solution (in this case a beam which minimises area *and* minimises the maximum amount of stress present).

A valid solution for this simplified model of the physics is any shape which does not exceed the maximum stress constraint, and the best solution would be the shape with minimal mass and the lowest value for its' maximum stress point. For the purpose of this one-dimensional test virtually any shape at all will be valid. The row labelled 'solution' in Table 3.1 represents the number of generations taken to find a valid solution with the minimum required mass (*mass-optimal*).



Categories of solutions.

Figure 3.1

The best possible solution for this simple problem has all of the mass evenly divided with as much separation between the two areas as possible. By looking at the stress analysis equations in Section 2.3.1 it is clear that the further apart the two stress bearing components are, the lower the maximum stress encountered in the beam will be.

Ten trials of this program produced the results shown in Table 3.1. In each case a mass-optimal solution was found rapidly and then evolution continued towards the expected best solution for this problem (though it rarely found that



solution during the period of the trial). Each trial was allowed to continue for 1000 generations (1000 Generations x 60 Population = 60000 Evaluations).

Trial	1	2	3	4	5	6	7	8	9	10
Solution	134	140	163	114	121	153	108	231	121	119
Best	-	-	-	464	-	-	-	905	-	-

Number of generations required to find solutions.

Table 3.1

These results show a mass-optimal solution being found after an average of 8424 evaluations, followed by a highly variable amount of time spent searching for the best possible solution. This is the type of behaviour expected from a Genetic Algorithm with a strong selection pressure for mass reduction and a much lower emphasis on optimality.

To confirm that the stress analysis mathematics was correctly implemented, the result of this particular problem was calculated independently, assuming that an 'I' beam is the best solution. For a steel beam 0.5m high the expected value for the second moment of area is approximately 0.001m^4 . Inspection of the variables generated by the stress analysis routine revealed a value of 0.00145m^4 for the fittest member of the population when the chromosome represented a beam 0.516m high. This result indicates that the second moment of area is being calculated correctly and from this and Equation 2.2 it is clear that the stress calculations must also be correct.

Having determined that the GA and the mathematics involved in the stress analysis fitness function both work as desired, the experiments progressed to a more complex treatment of the problem; the representation of the beam cross-section was expanded to include two dimensions, vertical and horizontal.



3.2 A two-dimensional representation of a beam cross-section.

The one-dimensional representation was incapable of representing a complete 'I' beam and so made do with representing the top and bottom surfaces, omitting the vertical *web* from the calculations entirely. The two-dimensional representation is theoretically capable of representing the web, however the additional mathematics required to simulate the vertical and horizontal forces and calculate the stresses which necessitate the existence of a web are sufficiently complex that it would require the use of a finite-element method package. In order to maintain the advantages to optimisation available from extremely fast evaluations, it was decided to continue to use the same simplistic model of the physical forces applied to the one-dimensional problem and to treat the consequent absence of a properly designed web as an unavoidable side-effect.

The code modifications required to extend the beam cross-sectional representation to two dimensions were minor because the original representation was already dealing with the width of the beam, although as a single binary parameter which represented a whole line. The new representation used was a two-dimensional array of binary cells, a change that required the addition of an x component to the calculations finding the centroid of the shape. This simple change increased the size of the chromosomes from 64 bits to $32 \times 64 = 2048$ bits, creating a search space of 2^{2048} which is extremely large even for a GA.

Large search spaces can prove problematical for all forms of search unless they have an extremely simple solution space, for example: A problem with only a single well-defined optimum can be searched easily by even a simple technique such as hill-climbing. However most real-world problems are far more complicated than this and consequently increasing the size of the search area decreases the probability of the global optimum being found.



3.3 The GA experiments.

The following experiments were intended to improve the performance of the GA on a reasonably simple two-dimensional beam analysis problem in order that the optimisations discovered would be available for the more complex real-world analysis detailed in Chapter Four of this thesis. To try to ensure that the alterations and improvements made to the GA here will also prove beneficial to the real-world problem, it was decided not to concentrate on fine-tuning any of the various parameters available but rather to focus on the design and operation of various new operators. Parametric variations were restricted to an absolute minimum and were used only to determine the approximate values required to gain reasonable advantages from the new operators.

In the following experiments the following parameter settings remain constant unless mentioned otherwise:

Beam Dimensions	= 0.05 x 0.10 metres
Bending Moment	= 13000 Newton
Young's Modulus	= 2.0e8 Pascal
Voxel Grid	= 32 x 64 voxels

3.3.1 Experiments using the naïve GA.

The first set of experiments with a 2D representation treated the chromosome as a long one-dimensional binary string which wrapped around at the vertical edges onto new lines to form the two-dimensional cross-section. The GA used a standard *two-point crossover* operator and the *uniform mutation* operator described in Section 2.2.3.

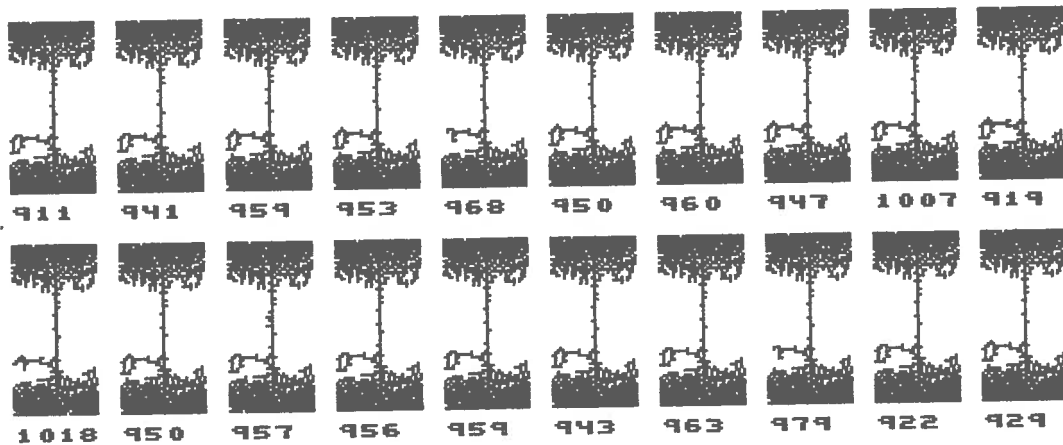


It used settings of:

Mutation Rate	= 0.001 (per bit)
Crossover Probability	= 0.35 (per chromosome)
Population Size	= 20
Ranking Pressure	= 3
Chromosome Length	= 2048 bits

3.3.2 Results obtained from the naïve GA.

With this particular optimisation problem, the difficulty lay not in getting a valid solution, as even a solid beam will satisfy that requirement, but in getting a near optimal-mass solution. The first experiments were relatively unsuccessful in this regard, despite showing a tendency to evolve towards the same known best case found in the 1D trials, the results after 2000 generations were full of small holes and had extremely uneven inner edges. This can be seen in the typical end-of-run results shown in Figure 3.2 (the numbers represent the fitness value for each individual).



End of Run Results From Naïve GA

Figure 3.2

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and activities. It emphasizes the need for transparency and accountability in financial reporting.

2. The second part of the document outlines the various methods and techniques used to collect and analyze data. It highlights the importance of using reliable sources and ensuring the integrity of the data collection process.

3. The third part of the document provides a detailed overview of the results of the study. It includes a comprehensive analysis of the data and a discussion of the findings and their implications.

4. The final part of the document concludes the study and offers recommendations for future research and practice. It emphasizes the need for continued monitoring and evaluation of the findings.

The stresses were concentrated at the vertical extremes of the beam, so the material in the middle contributes less towards the total maximum stress that the beam can withstand than the material at the top and bottom. The GA even in this simple standard form rapidly removed material from the middle of the cross-section, and in the later stages of the experiments was observed to be moving material from low stress areas into high stress areas where holes were left near the extremities. However this first naïve GA approach took an extremely large number of evaluations in order to make significant progress, and this is not acceptable as later experiments will have a greatly increased evaluation time when the FE package is integrated. The rate of improvement was also seen to decrease as the run continued, levelling off to almost none at all by the end of the run. This means that the GA was not finding any further improvements to the chromosome, and as the results are visibly poor it indicates a general weakness in the operators being applied.

Figure 3.3 shows a graph of fitness values versus generation number for a typical selection of three of the ten trials performed with this program. The horizontal axis represents the generation number at which the results were found and the vertical axis represents the fitness values. Fitness values should be minimised for this optimisation, so the lower the fitness value, the better the quality of that individual solution.

The steepness of the curve during the first two hundred generations indicates that improvements were being made rapidly during this period, but as the optimisation approached the five-hundredth generation the rate of progress had diminished considerably. This effect was expected as the GA will rapidly find most of the obvious optimisations available and then the search will have to work harder to find any further worthwhile improvements. The problem with these results was that the naïve GA levelled off too early; the quality of solution at the end of the trial indicates that there were still many improvements to be made but the rate of improvement visible on the graph is almost zero.



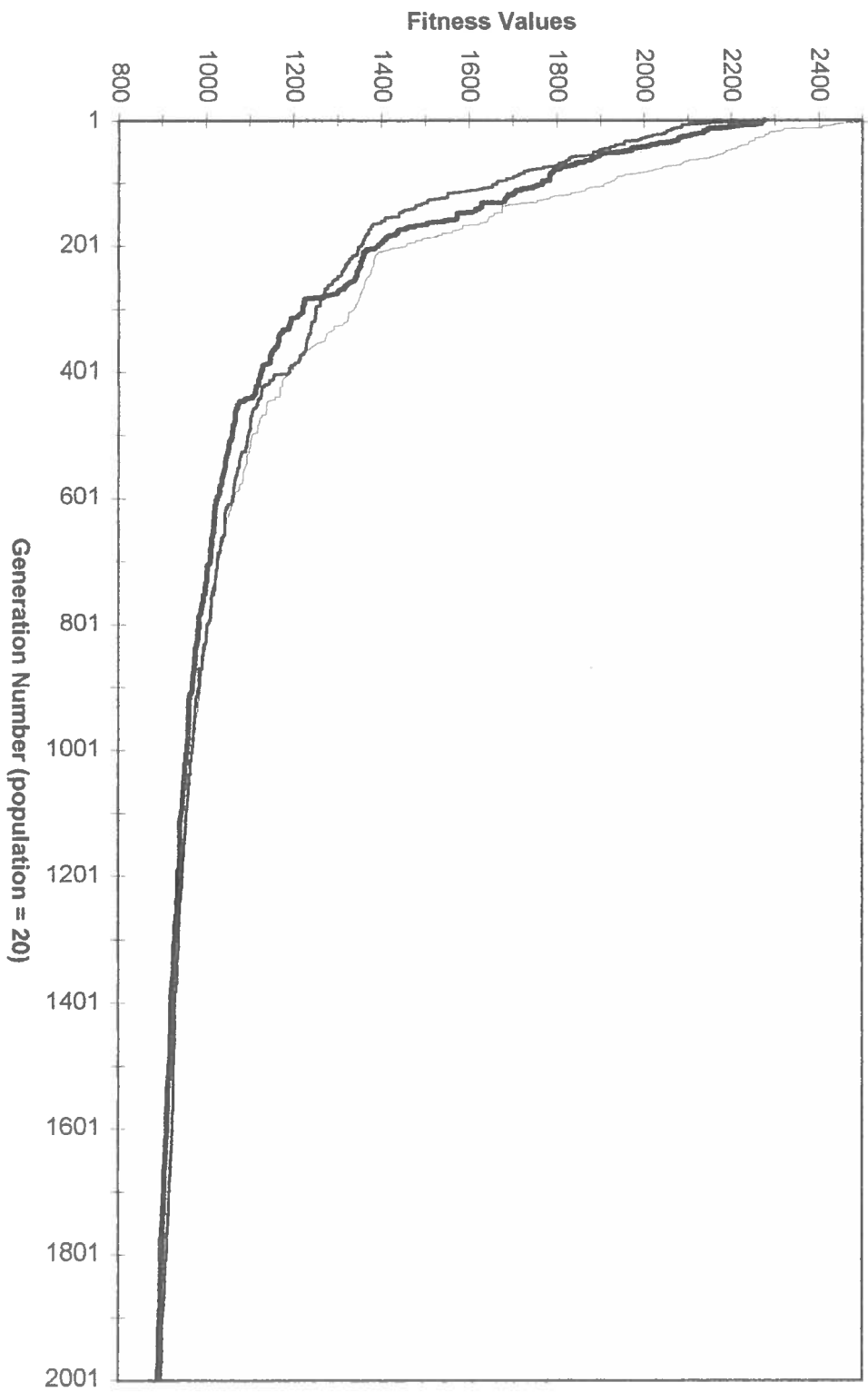


Figure 3.3
Three Fitness Curves from the Naive GA



Further work detailed in this chapter was therefore concentrated towards improving the operators and parameter settings for the GA, in order to achieve greater benefits during the early search period, and to produce better quality final results.

3.3.3 Experiments with the rank selection pressure.

This experiment was intended to discover the effect of changing the rank selection pressure variable used in the Genitor-type ranking algorithm [Whitley 89]. The pressure variable controls the proportional selection from the population based on the fitness ranked position within that population. A selection pressure of 1.0 results in a totally random selection (all members have equal probability of being selected), and higher selection pressures result in increasing probability that the fittest members of the population will be selected. The Genitor paper suggests that 1.5 is a suitable value to be used for many experiments, producing a good balance between population diversity and the rate of fitness improvements.

Implementation of the algorithm is exactly as given in the Genitor paper* :

```
linear_selection( pressure : float )  
    index = Population_size *  
           (pressure -  $\sqrt{(\text{pressure}^2 - 4 * (\text{pressure} - 1) * \text{random}(1.0))}$ ) / 2 / (pressure - 1)
```

The Genitor Linear_Selection() Function.

Figure 3.4

* incidentally, this algorithm will not operate with a selection pressure of 1.0 as this results in a divide by zero.



The population is sorted by its fitness values using a linked-list bubble-sort algorithm before the Genitor selection procedure is invoked, thus the result of the `linear_selection()` routine is an index to the linked-list, which in turn holds the position of the selected individual in the population.

The survivors selection procedure is simply to invoke this procedure once for every member of the population, each time copying the selected member of the old population into consecutive elements of the new population array.

The Genetic Algorithm used to perform the searches was the same as described in Section 2.2.3: ten trials were executed and the number of iterations per trial was set to 2000.

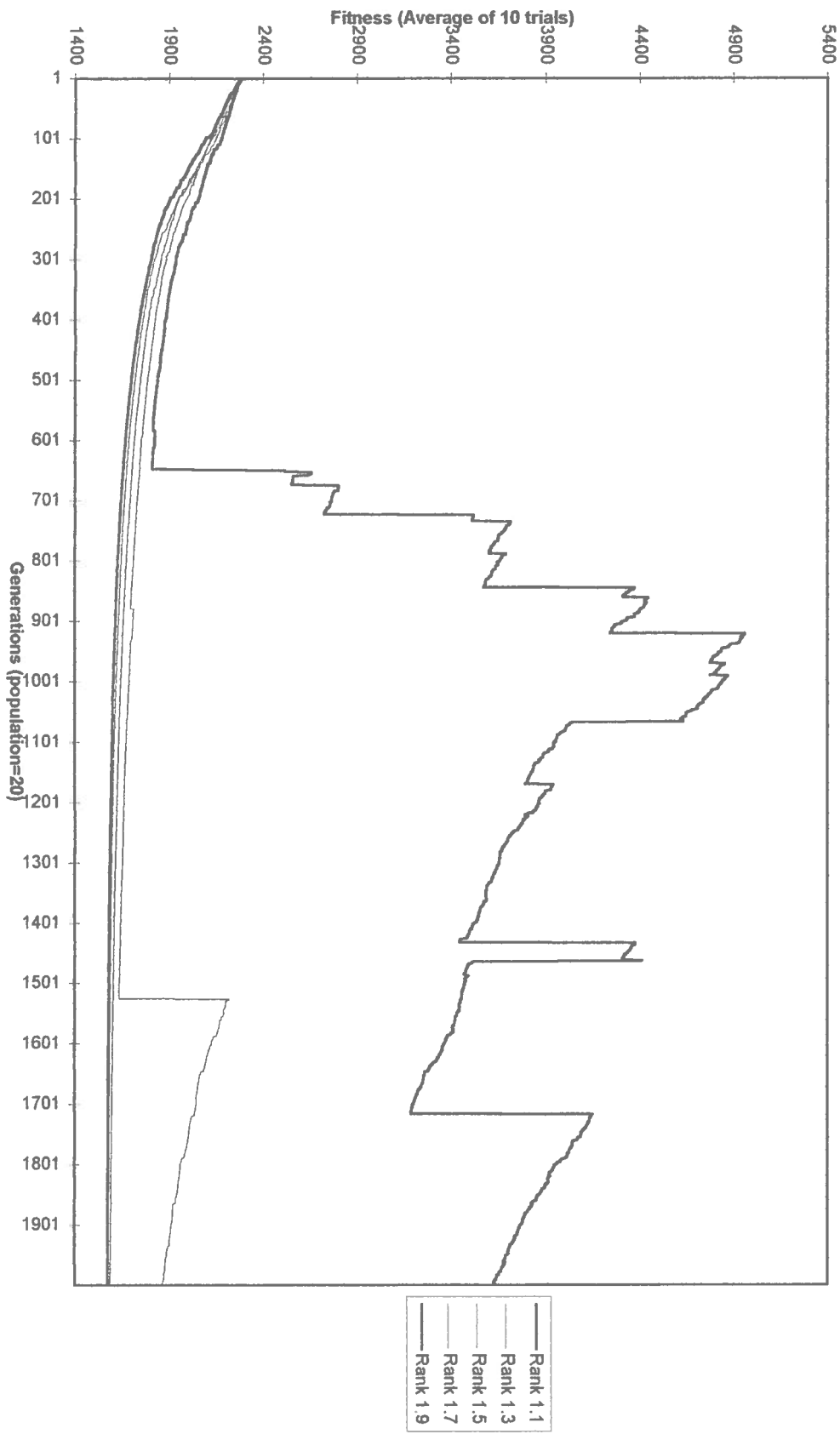
Figure 3.5 is a plot of the fitness of the best individual from the population averaged over all ten trial runs (on the vertical axis) against the generation number (on the horizontal axis). It shows the full set of results and clearly indicates that at selection pressures of 1.1 and even 1.3, there is insufficient differentiation between the poor members of the population and the fit ones. This results in chromosomes which violate the problem's constraints being selected frequently enough to significantly degrade the average results. The large steps visible in both of these lines are caused by selection of individuals who are extremely unfit due to large constraint penalties being applied.

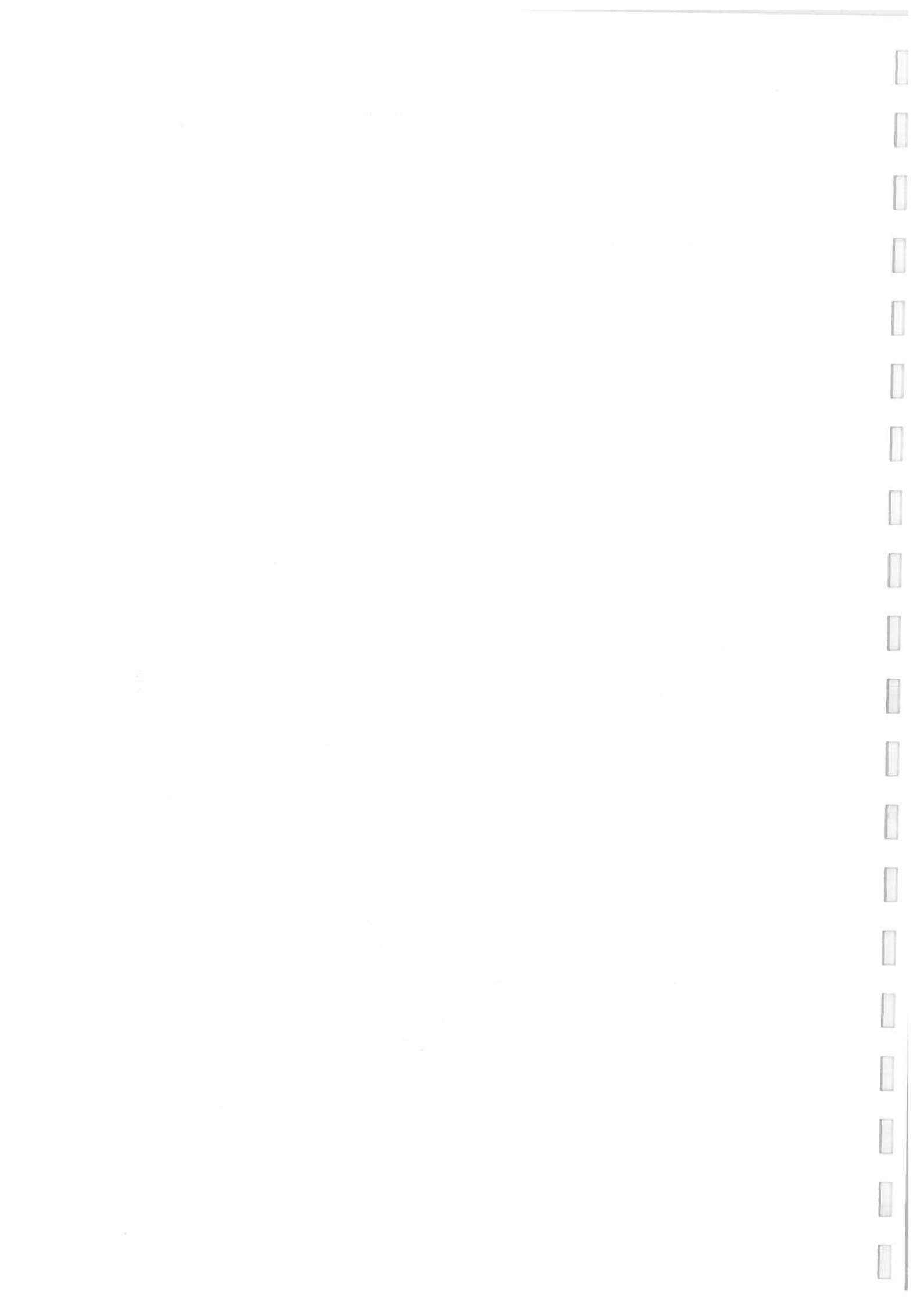
Figure 3.6 is a more detailed examination of the 1.3, 1.5, 1.7 and 1.9 selection pressures in which we see the expected results of higher pressures producing steeper fitness curves but ultimately resulting in similar overall fitness levels after 40000 evaluations. The 1.5 line is very slow to converge with the 1.7 and 1.9 lines, indicating that it takes significantly longer before it produces solutions of equivalent quality.

The very low selection values do not appear to cause enough selection pressure to avoid repeated constraint violations, although this may be affected by adjusting the Constraint Penalty Multiplier. The lower selection values also result in



Effect of Rank Pressure On Fitness.





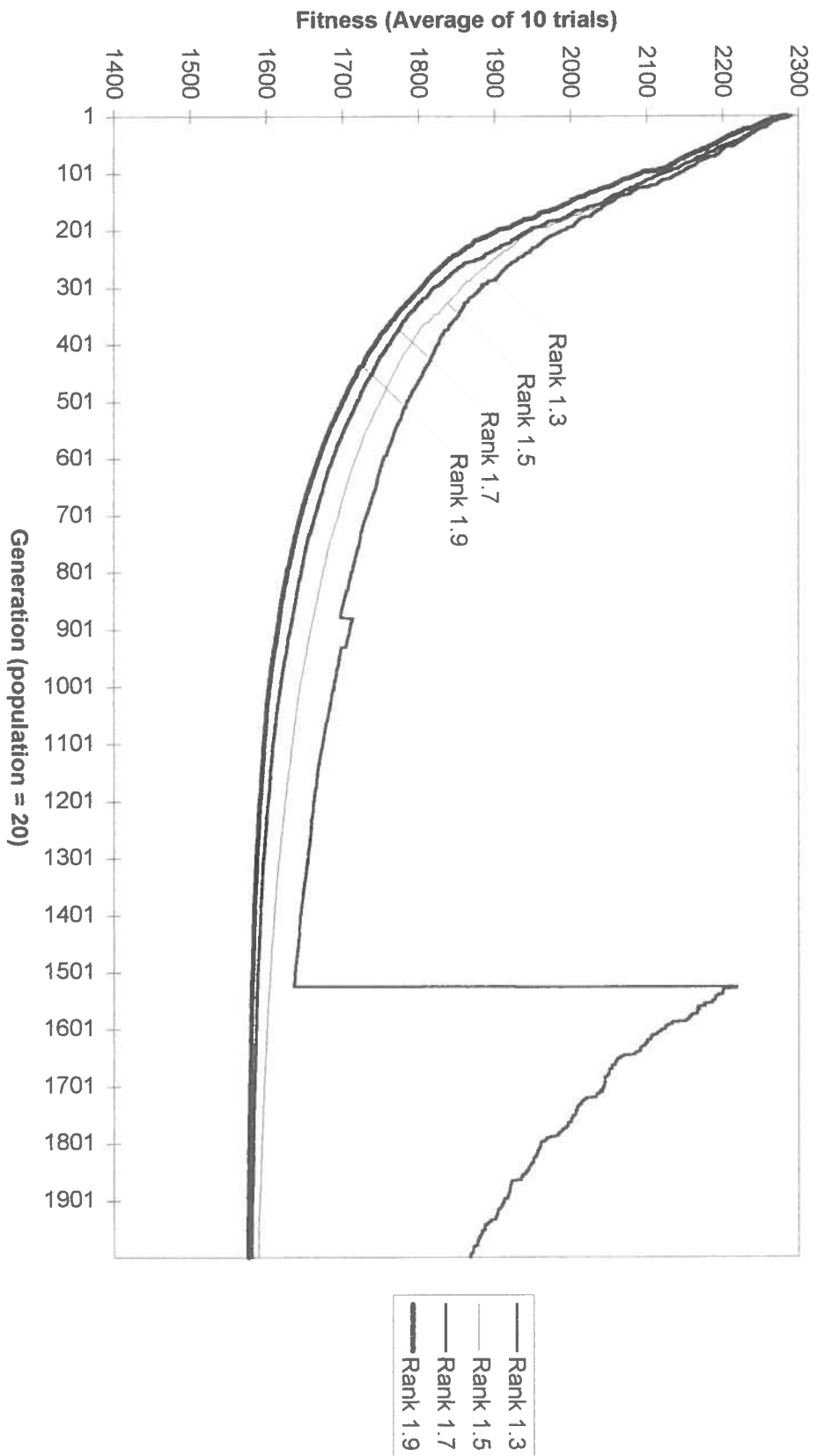


Figure 3.6
Effect of Rank Pressure on Fitness.

Figure 1: A graph showing the relationship between variables X and Y.



Y-axis label

X-axis label

Legend

a slightly worse result after 2000 generations. As a consequence of these trials further experiments will be performed using a rank selection pressure of 1.7, as the suggested 1.5 is too close to the border-line where the constraints are being violated regularly and it also does not converge rapidly enough to the better solutions found at higher rank pressure levels.

3.3.4 Experiments with the constraint penalty multiplier.

The purpose of these experiments was to investigate the effects of changing the constraint penalty multiplier k used in the fitness function (Equation 2.1), and to find an appropriate value for this variable which maximises the search capabilities whilst avoiding an excess of non-viable solutions. The constraint penalty multiplier (hereafter referred to as the penalty value) is applied to the fitness value of any chromosome which has a maximum stress value greater than the constraint level, this will lower the relative fitness of that chromosome reducing its probability of survival into the next generation.

The penalty value is multiplied by the amount of stress which exceeds the stress constraint, so the effect on a chromosome's fitness depends on the degree to which it is non-viable. This approach permits chromosomes which are only slightly over the constraint limit to survive for a while, thus increasing the population diversity at and around the stress constraint level.

The Genetic Algorithm used was as described in Section 2.2.3 except that the vertical *web* which runs through the middle of the 'I' beam was enforced after the operators were applied and before the evaluation function was called. Enforcing the web avoids the complicating issue of the amount of mass added by the web itself. A crooked web would contain more mass than a straight one, but the simple model of the physics being used for the evaluation stage does not contain any forces that would encourage a straight web. Rather than try to factor out the effect that the



different randomly evolving webs would have on the final fitness values, it was considered to be more sensible to avoid the issue entirely in this way.

A population size of 20 and an initial run length of 1500 generations was used. The optimisation was performed ten times with each of the penalty values 1E-3, 5E-4, 1E-4, 5E-5 and 1E-5. Other parameters were set as follows:

Beam dimensions	= 0.05m x 0.10m
Bending moment	= 13000 Newton
Voxel grid size	= 32 x 64 voxels
Crossover probability	= 0.3 (per chromosome)
Mutation probability	= 0.001 (per gene)
Rank selection pressure	= 1.7
Stress constraint	= 2.0e8 Pascal
The web was enforced.	

The first results graph (Figure 3.7) for these trials shows a plot of the average fitness of the best individual from the population over the ten trials against the generation number. This graph is somewhat misleading as it indicates that as the penalty value is reduced the average fitness of the population tends to improve, unfortunately this is at the cost of increasingly poor solutions. What is not clear from this graph alone is that most of the fittest individuals at the lower penalty values are non-viable in that they break the constraint limit - often by a considerable amount.

The second graph (Figure 3.8) shows the maximum stress level of the best individual from the population averaged over the ten trials against the generation number, the stress limit constraint is marked as a horizontal line for reference. This graph clearly shows the difficulty with the lower penalty values - the data for 1E-5 is almost entirely above the constraint limit meaning that the majority of the solutions from the population were non-viable.



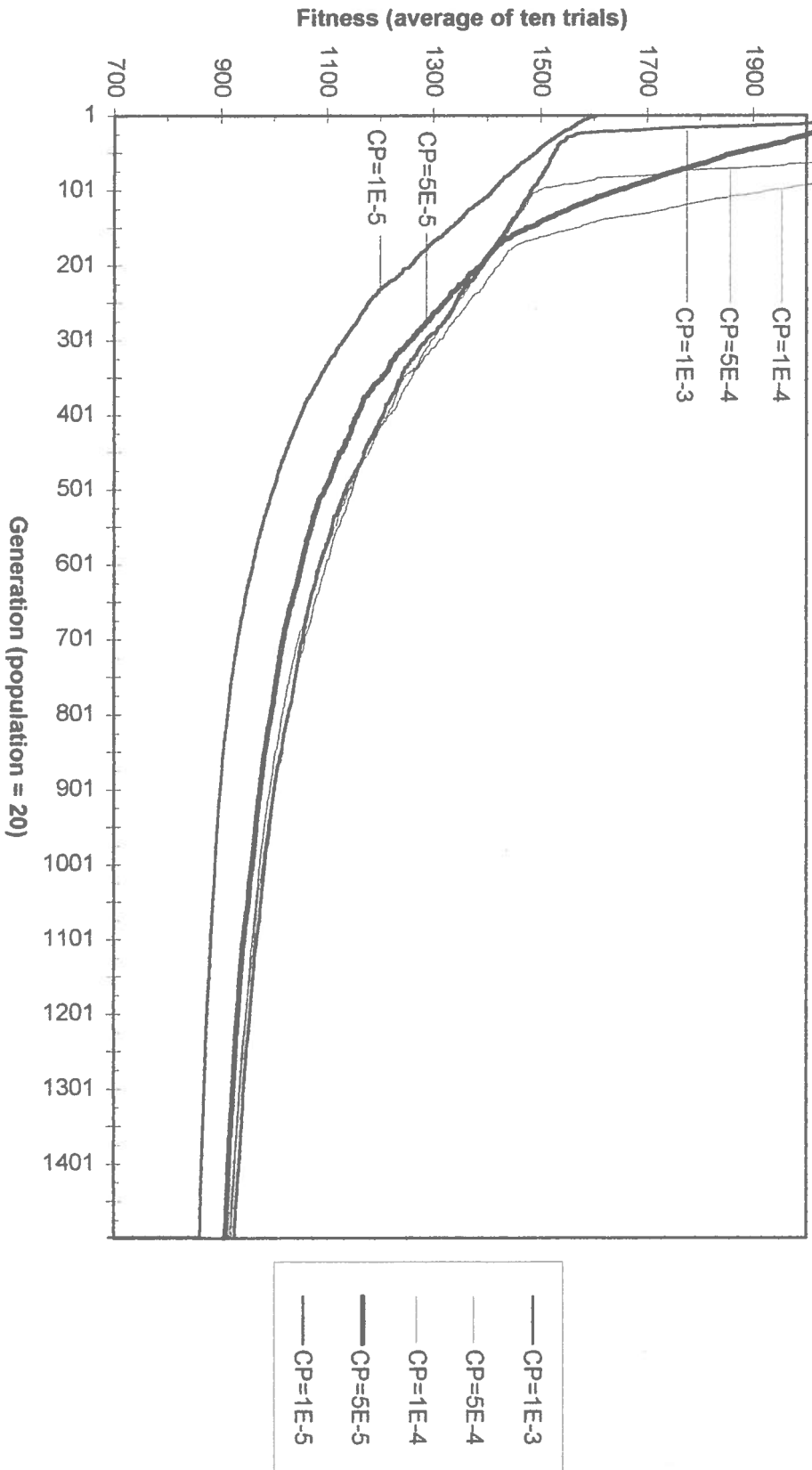


Figure 3.7
Effect of the Constraint Penalty Multiplier on Fitness Values

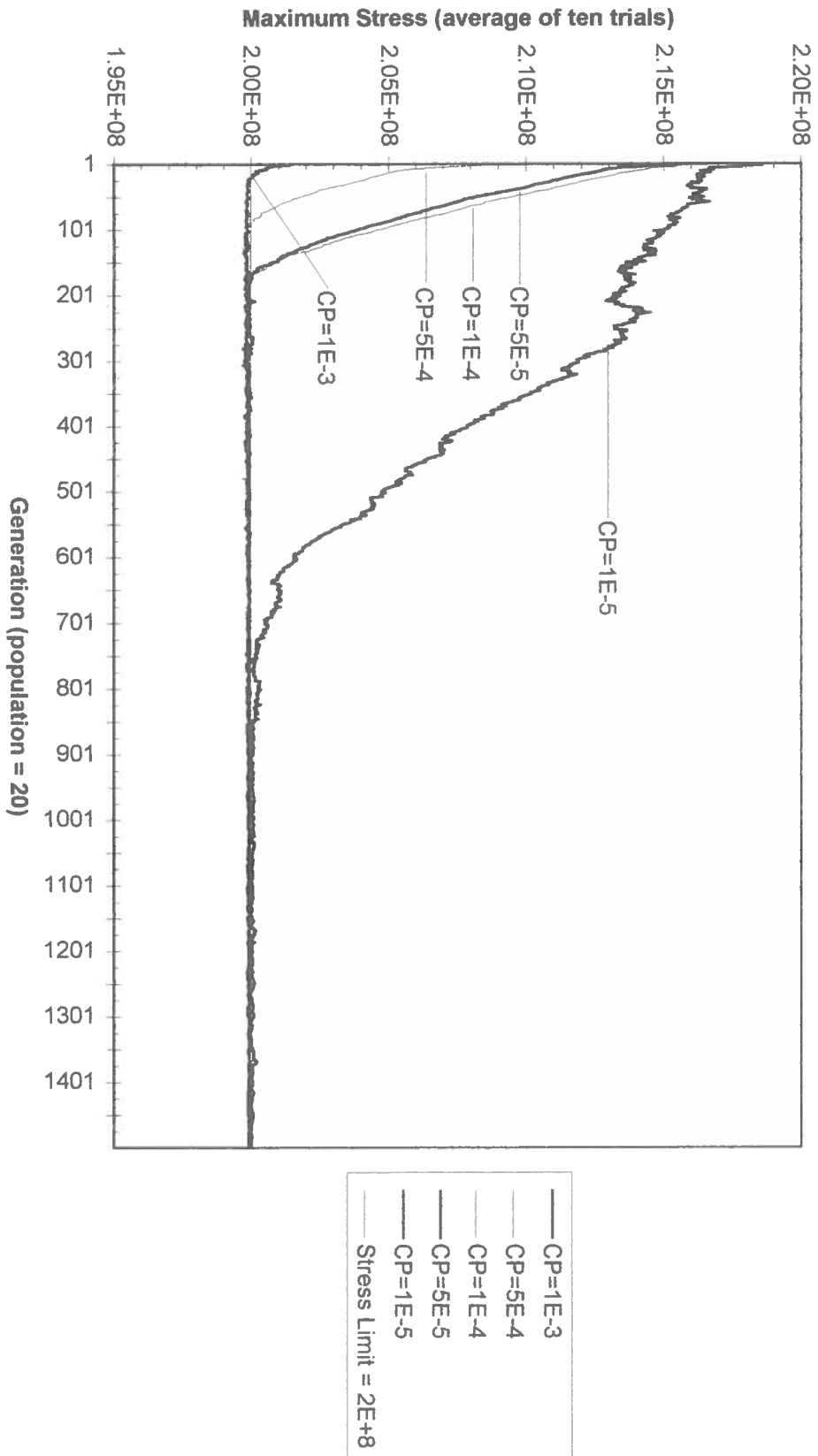
UNITED STATES DEPARTMENT OF AGRICULTURE



UNITED STATES DEPARTMENT OF AGRICULTURE



Figure 3.8
Effect of the Constraint Penalty Multiplier on Maximum Stress Levels



THE UNIVERSITY OF CHICAGO
DIVISION OF THE PHYSICAL SCIENCES
DEPARTMENT OF CHEMISTRY
5708 SOUTH CAMPUS DRIVE
CHICAGO, ILLINOIS 60637
TEL: 773-936-5000
WWW.CHEM.UCHICAGO.EDU



To be able to see the results for the 1E-3 and 1E-4 penalty values a considerable expansion of scale is required, Figure 3.9 shows the data from Figure 3.8 expanded about the constraint limit line (which is again included in the graph). Here even though the Maximum Stress average is very noisy data, we can see that the higher penalty values cause maximum stress to be further below the constraint limit than the lower penalties. The average for the penalty value 1E-4 actually crosses the constraint limit several times, but always returns to viable solutions after a short while.

Higher penalty values prevent recurrent violation of the stress constraint, but very high penalties reduce the available search space by forcing solutions to stay further within the limits. Even in the long trials the higher penalty values have this restrictive effect. This is in keeping with the results found by [Chapman *et al.* 94].

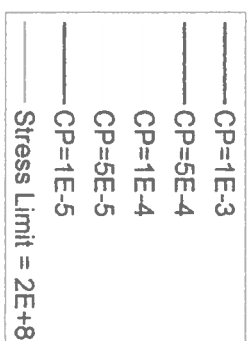
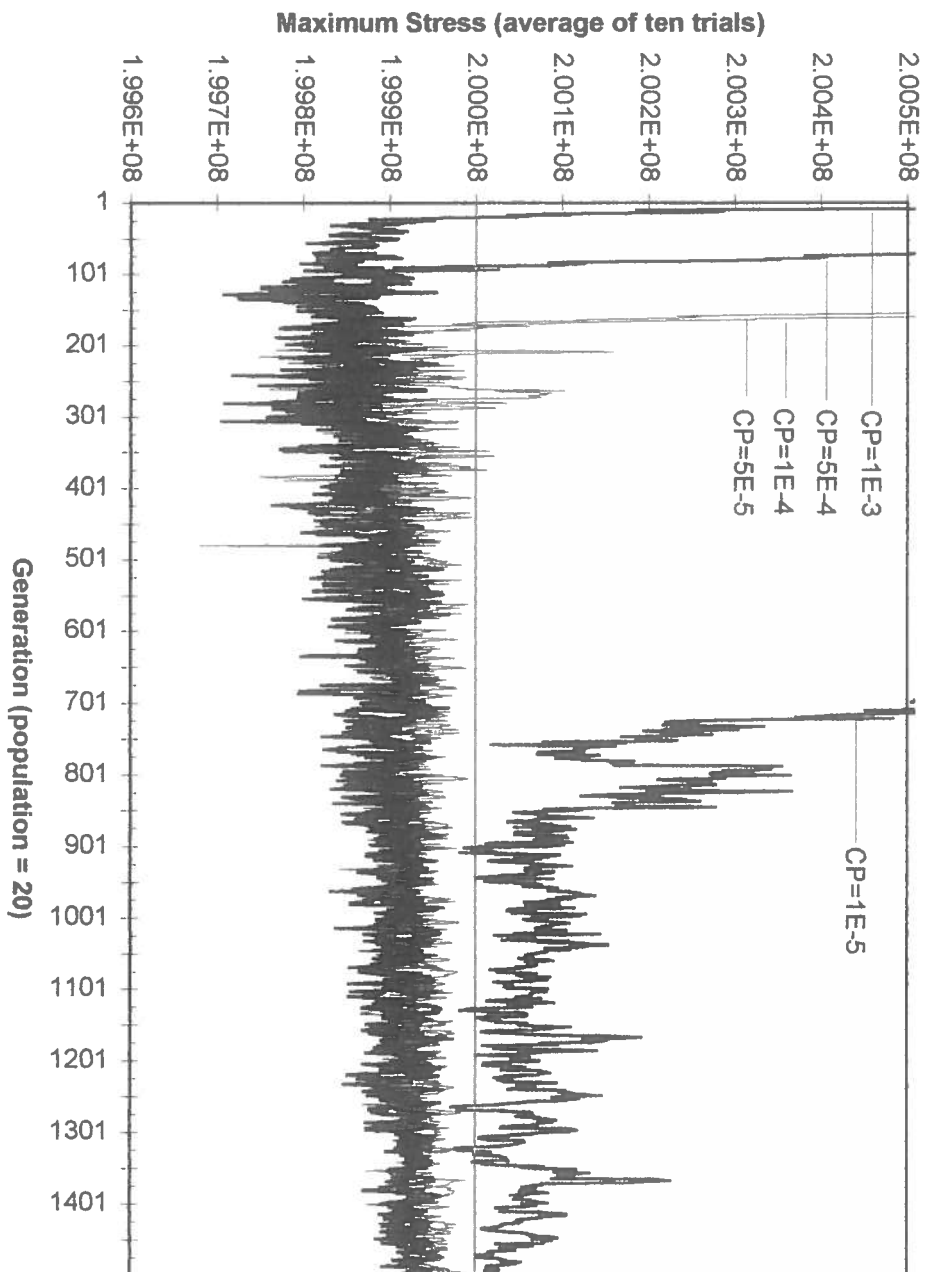
As the standard genetic algorithm verifies the viability of the initial population by using the fitness evaluation function (which incorporates the penalty value), the fitness/generation curves do not start at the same place for different trials but instead indicate that a lower average initial fitness can be achieved by using lower penalty values. This effect can be extremely useful, as a more fit initial population reduces the length of search required in order to achieve a given quality of result.

An ideal constraint penalty would allow the search to utilise the possibilities beyond the constraint limit at the beginning of the search and would cause more solutions to be valid as the search progresses towards its conclusion. This would encourage a fitter starting population, allow maximum population diversity initially and ensure that the eventual result population contains useful solutions.

These results suggest that a dynamically varying penalty value may result in more rapid optimisation in the early stages of the search whilst avoiding the problem of non-viable final solutions. This possibility was investigated and the results are included in the next section.



Figure 3.8
Effect of the Constraint Penalty Multiplier on Maximum Stress Levels - Detail



100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100

100



3.3.5 Experiments with a dynamically changing constraint penalty multiplier.

These experiments investigate the efficiency of using a dynamically changing constraint penalty multiplier rather than the static values used in the previous set of experiments. The rationale for this approach is that it may be good for the search if it is allowed to break the constraints during the early stages of the optimisation, but that the results at the end of the run *must* contain viable solutions.

The constraint penalty multiplier must start off low and increase as the optimisation continues, therefore the generation number will be used as the variable which controls the function:

$$f(\text{generation}) = \text{constraint penalty multiplier}$$

Equation 3.1

A linear function could be used to test this approach, however it is apparent that not only should the optimisation end with viable solutions, it should spend some time optimising the viable solutions before the end of the run. A more suitable curve function is:

$$\text{constraint penalty multiplier} = \text{constraint base} * (1.0 - e^{(-\text{alpha} * n)})$$

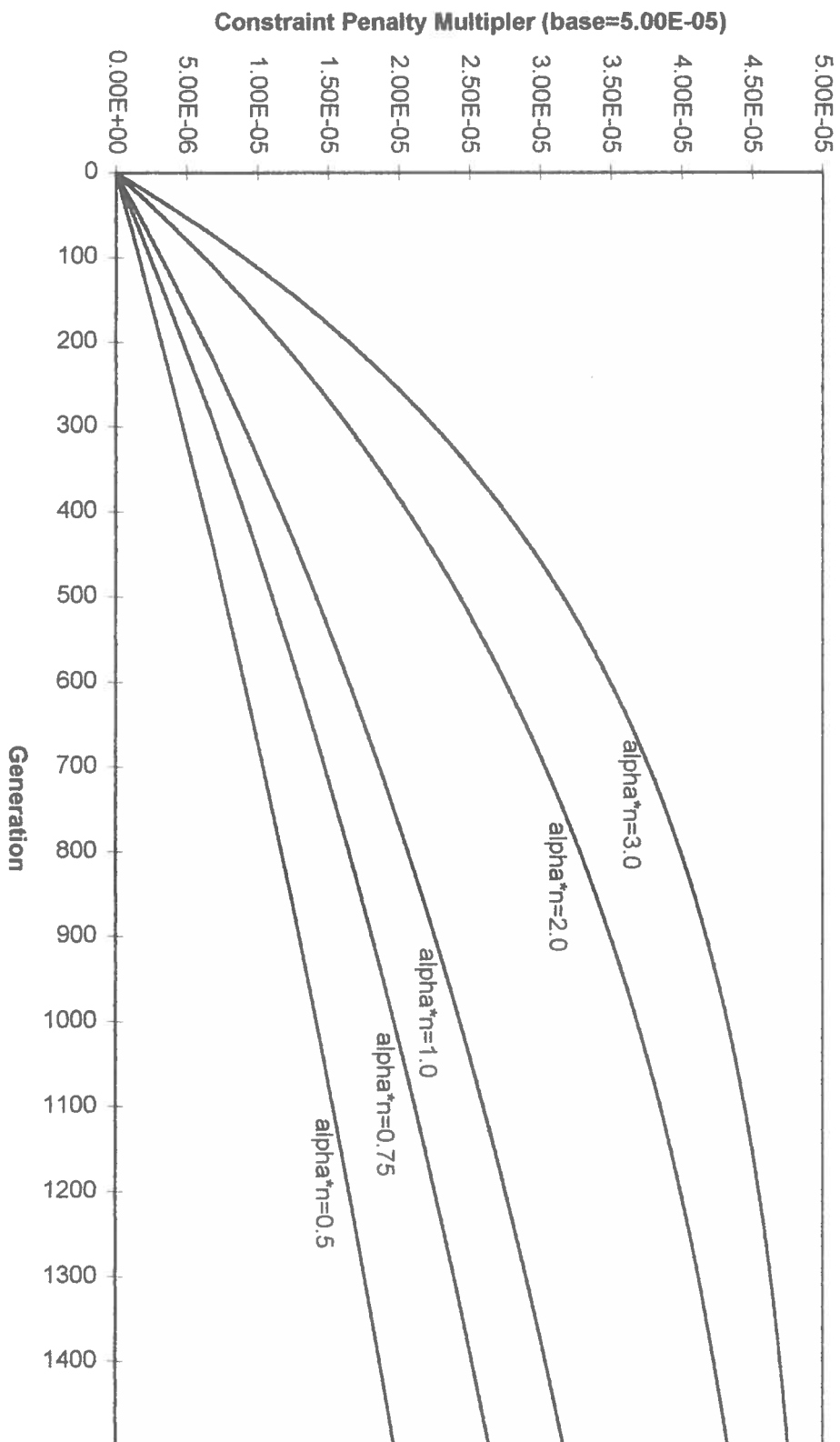
where n is the generation number.

Equation 3.2

This can be seen for several values of $(\text{alpha} * n)$ in Figure 3.10 which shows the constraint penalty multiplier for each generation to fifteen hundred. The function



Figure 3.10
Dynamically Changing Constraint Penalty Multiplier



at the beginning of the year.

at the beginning of the year.

The following table shows the results of the survey. The first column shows the number of respondents, the second column shows the percentage of respondents, and the third column shows the number of respondents who answered "yes".

Question	Number of respondents	Percentage of respondents	Number of respondents who answered "yes"
1. Do you have a car?	100	100%	75
2. Do you have a house?	100	100%	60
3. Do you have a job?	100	100%	85
4. Do you have a bank account?	100	100%	90
5. Do you have a credit card?	100	100%	70
6. Do you have a car insurance?	100	100%	80
7. Do you have a health insurance?	100	100%	95
8. Do you have a life insurance?	100	100%	65
9. Do you have a travel insurance?	100	100%	55
10. Do you have a fire insurance?	100	100%	70



describes a smooth curve from zero to a maximum, which is a proportion of the constraint base value determined by the value of α^n used. A higher value of α^n makes the curve more pronounced and results in a higher proportion of the constraint base value at the final generation.

The Genetic Algorithm used was as described in Section 2.2.3 with a population size of 20 and an initial run length of 1500 generations. The optimisation was performed ten times with each of the α^n values 0.5, 0.75, 1.0, 2.0 and 3.0.

The graph (Figure 3.11) shows the average fitness over ten trials for each of the tested α^n values plotted against the generation number. This graph shows the effect of very low constraint penalty multipliers in the first half of the run; the fitness curves go up from reasonable starting values to very poor results during the first four hundred generations, peaking at various points depending on the α^n value being used. It can be seen that the higher α^n values result in worse early results, but better results later on in the run, although the $\alpha^n = 0.75$ line is anomalous in that it leads to a better final result than the 1.0 line.

The reason for the poor fitness values early in the run from higher α^n values is that the constraint penalty multiplier increases faster for the higher α^n values, thus the solutions are being penalised more strongly than those with lower values. The better final results appear to be due to the rapidly increasing penalty multiplier values forcing the search into a viable search space earlier, thereby giving the search more time to do fine-tuning on the final solution.

Comparing the fitness graph results with the data in Figure 3.12 which shows the maximum stress values over the same time period leads to some further insights into the behaviour of the fitness curves. The lower values of α^n lead to greater violations of the stress limit constraint, to the point where $\alpha^n = 0.5$ doesn't produce any viable solutions at all during the 1500 generations. Typically the higher values lead to viable solutions sooner, with the exception of $\alpha^n = 1.0$ which takes longer to descend than $\alpha^n = 0.75$. This can be seen more clearly in Figure 3.13 which shows a detailed view of the second half of the optimisation period and



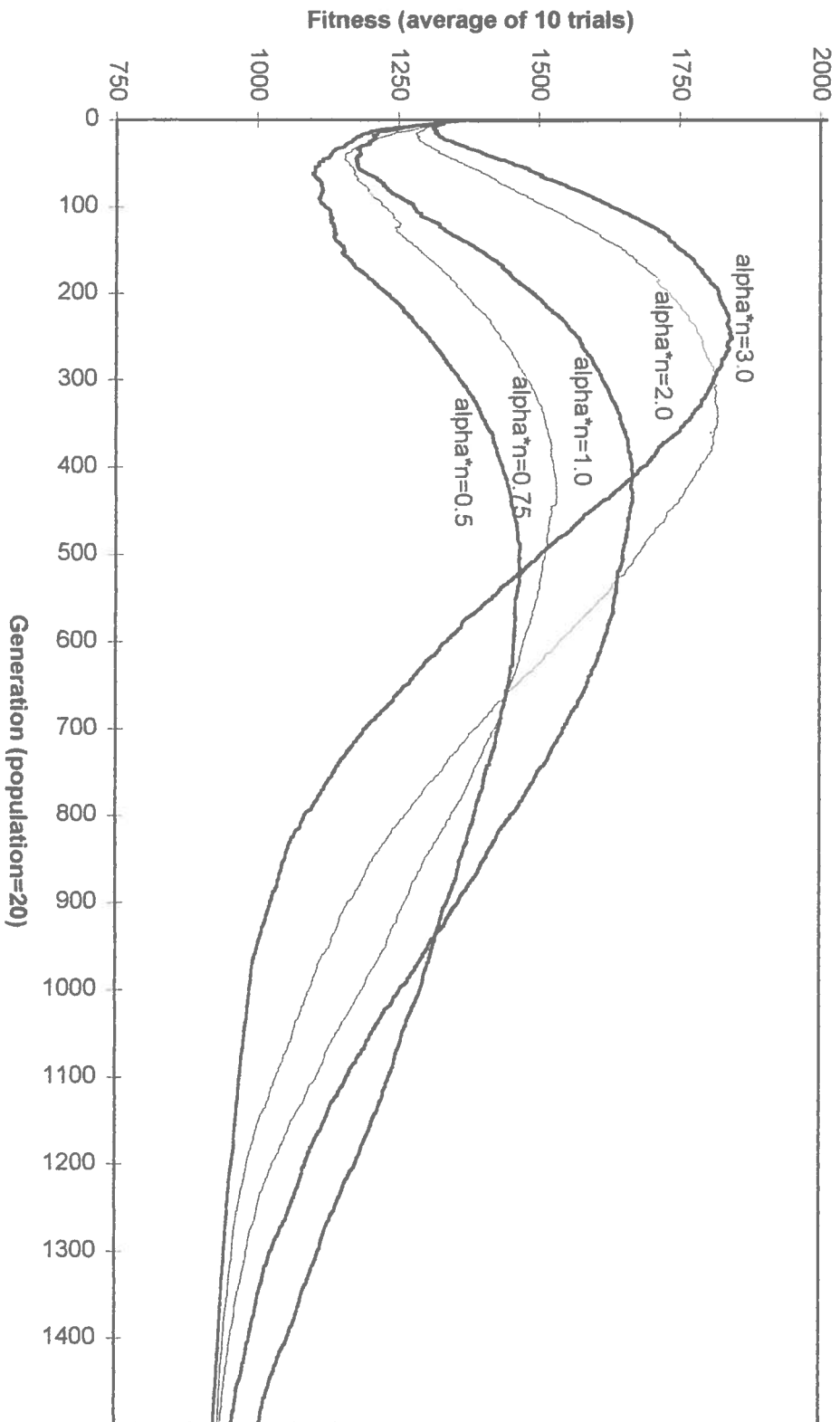


Figure 3.11
Effect of Dynamically Changing Constraint Penalty Multiplier on Fitness

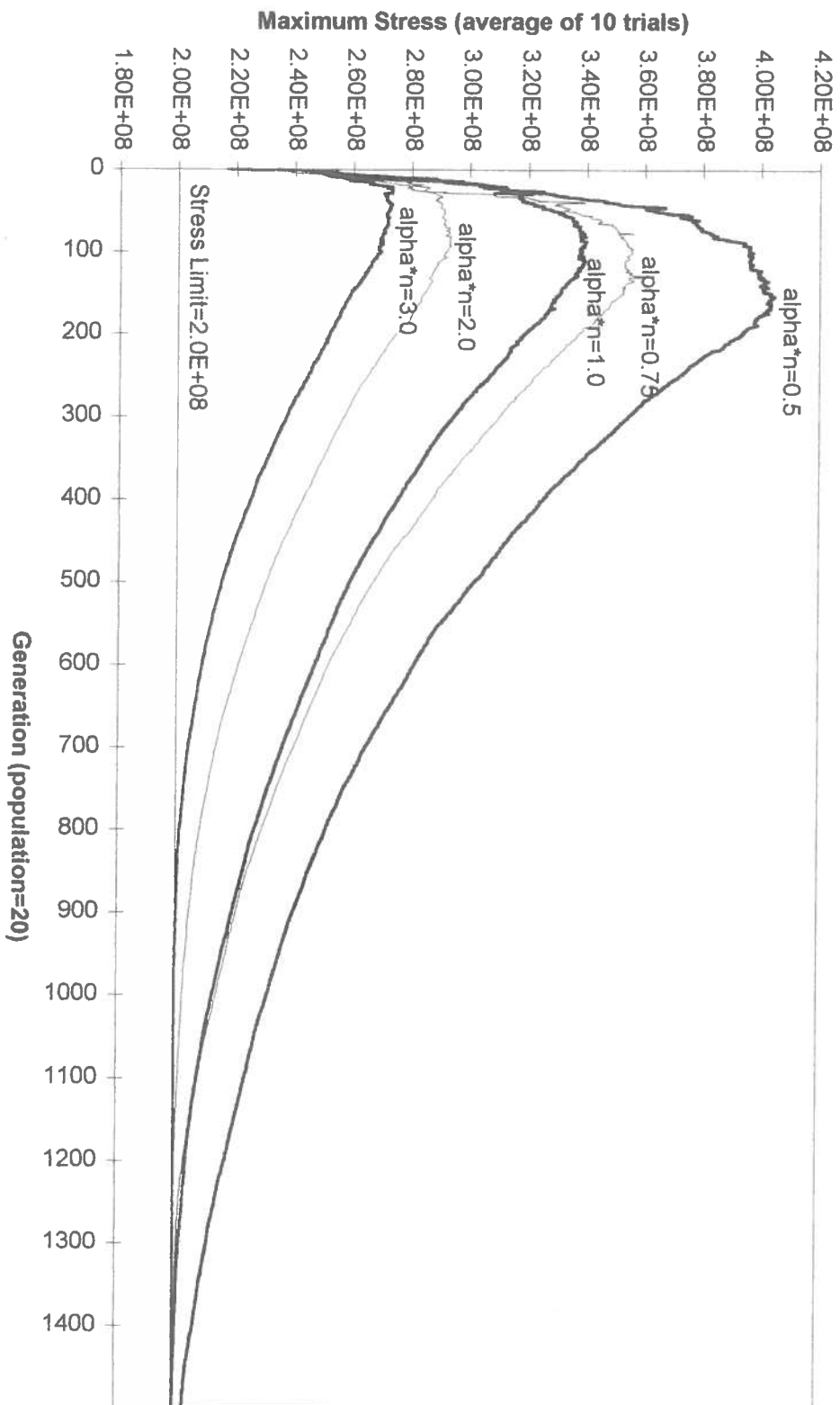
1. Formulas for integrals

Let $f(x)$ be a function defined on the interval $[a, b]$. The definite integral of $f(x)$ from a to b is denoted by $\int_a^b f(x) dx$.



The area under the curve is given by the definite integral $\int_a^b f(x) dx$.

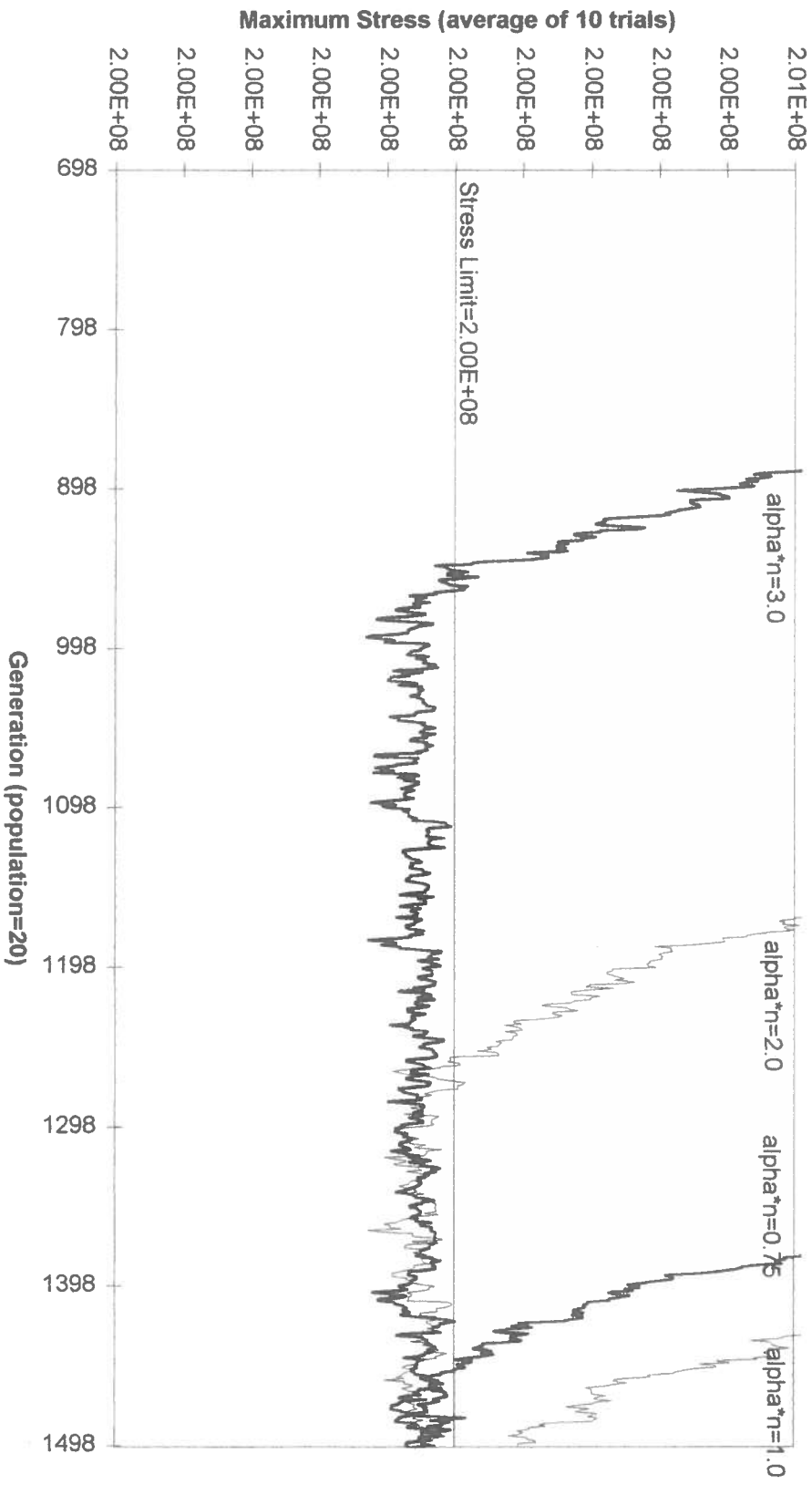
Figure 3.12
Effect of Dynamically Changing Constraint Penalty Multiplier on Maximum Stress Level



Handwritten text, possibly a title or header, located at the top of the page.



Figure 3.13
Effect of Dynamically Changing Penalty Multiplier on Maximum Stress Level
(Detail)



The following table shows the results of the experiment. The data indicates that the reaction rate increases with the concentration of the reactants.



Figure 1: Reaction rate vs. concentration

indicates that at $\alpha \cdot n = 1.0$ there were no viable solutions produced during the run.

Figure 3.13 shows that once the solutions enter the viable region by descending below the stress limit they rarely exceed that limit again, and indeed tend to remain at a fairly constant distance from the limit. This is possibly due to the constraint penalty multiplier value which, by continuing to increase, makes it less and less likely that a non-viable solution will be more fit than a viable one.

The fitness values at the end of the optimisation are not very much different than those obtained for good constant values for the constraint penalty multiplier. Table 3.2 contains the end of run fitness values for both constant penalty multipliers and the dynamic penalty multipliers, entries followed by a * are invalid solutions which exceeded the constraint penalty at the end of the run.

Constant constraint penalty multiplier values:

0.001	0.0005	0.0001	0.00005	0.00001
925.999	916.599	919.843	908.123	861.650 *

Dynamic constraint penalty $\alpha \cdot n$ values:

0.5	0.75	1.0	2.0	3.0
1004.231 *	936.266	957.609 *	932.259	924.466

End of Run Values for Average Fitness.

Table 3.2

The intention of this approach was to allow maximum population diversity initially, but ensure that the eventual resultant population contains viable solutions, and in this regard the concept appears to be successful. As long as the $\alpha \cdot n$ value is set to ensure a suitable constraint penalty multiplier value at the end of a run, the prevalence of invalid solutions in the early stages of the optimisation is not a



problem for the type of problem being addressed here. This technique is obviously not suitable for optimisations of uncertain length or when the optimisation may be stopped before convergence and a viable solution will be required.

The fitness curve graph (Figure 3.11) shows that the eventual fitness of the best individuals varies very little, which indicates that the precise setting of the α^n value is not critical. However the evidence of both the fitness curves and the maximum stress curves is that a high value of α^n is better for the search, resulting in better final solutions and avoiding the danger of non-viable populations by a larger margin. Further experiments which use dynamically varying constraint penalty multipliers would have used an α^n value of 2.0, however, as it can be seen from the Table 3.2, the constant constraint penalty multipliers consistently perform better than the dynamic ones, therefore constant constraint penalty multipliers were used during the following experiments.

3.3.6 A new operator : smoothing.

The smoothing operator experiments were an attempt to directly address some of the weaknesses of the voxel representation by devising a new specialised operator which should aid the search by reducing the number of small holes and ragged edges produced by the GA. The new operator was intended to be capable of easy expansion from two-dimensions to n-dimensions in order that it would continue to be useful in the case of three- (or more) dimensional problems using the voxel representation. This would be an important issue for shape optimisations which are not capable of being represented by a simple cross-section, for example if a beam was to have a varying cross-section along its entire length then it would be necessary to represent it using three-dimensional voxels.

The GA parameters used were the same as the standard GA described in Section 2.2.3 and the new operator was applied *in addition* to the previous mutation



and cross-over operators. The population size was set to twenty, and the number of generations was limited to fifteen hundred.

The new operator was designed to set a region of voxels to the average value of the previous contents of the voxels. Thus, if a region consisted primarily of 'on' voxels then the whole region would be set 'on' and similarly for 'off'. This would achieve the desired results by filling isolated holes in solid regions, and would also eliminate the ragged edges where they overlap into the empty regions.

The size and location of the region to be affected was selected at random, however the size was restricted to a minimum of 2 voxels per dimension and a maximum of 1/4 of the dimensions of the voxel grid. These restrictions were imposed in an attempt to ensure that the smoothing operator would always have an effect when applied, and that the effect would not be to wipe out a huge area of the voxel grid. Because the location of the region was selected randomly, the operator was written in such a way that the region would 'wrap-around' to the opposite side of the voxel grid if it exceeded any of the grid's boundaries.

In order to check the efficiency of the new operator, the modified GA was run ten times with each of the probabilities 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0 of smoothing being applied on a per chromosome basis.

The results of the smoothing operator on the population can be seen in Figure 3.14 which plots the average fitness values of the best member of the population over the ten trial runs against the generation number, for each of the six different smoothing probabilities applied. This graph shows that the smoothing operator definitely increases the rate at which the fitness improves during the early part of the runs; the increased steepness of the curves and the generally lower final values at generation 1500 are good evidence that the operator is beneficial. There *is* some 'ideal' rate of application for this operator as witnessed by the improvement from 0.0 through to 0.6 and then the decrease in effectiveness from there to the 1.0 probability level. The best result from these trials was gained at 0.6 and so that will be the probability of smoothing used in further experiments.



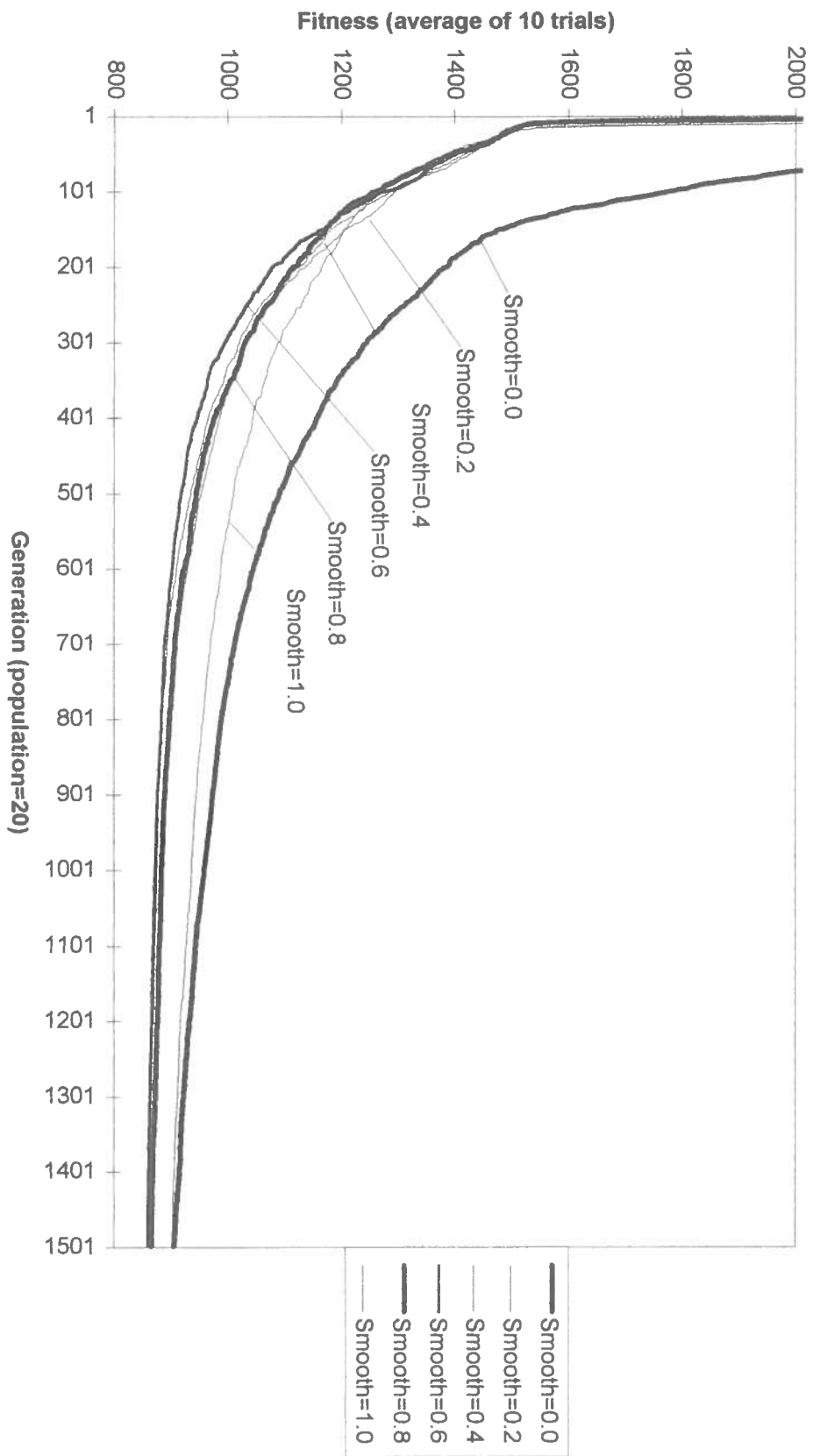


Figure 3.14
Effect of Smoothing Operator on Fitness

1000



1000

1000

Comparing Figure 3.15 which displays some typical end-of-run population members with the earlier results in Figure 3.2, shows just how effective this domain specific approach to operator design has been, especially at eliminating isolated holes and reducing ragged edges.

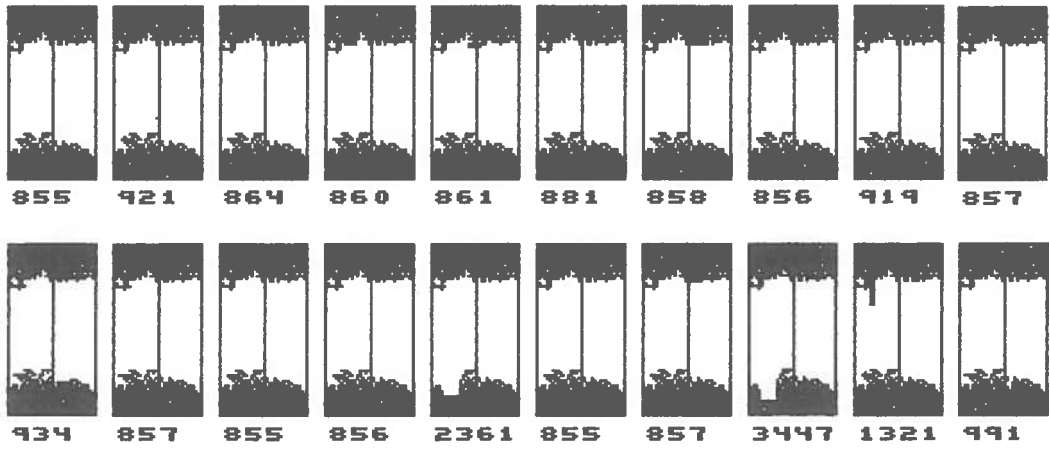
This set of experiments was intended to address two separate issues in operator design; firstly, will specially designed operators function as wanted when reducing particular undesirable effects during the optimisation process; secondly, can such operators be designed so that they can be expanded to n-dimensions with relative ease? The answer to both of these questions is a definitive 'yes'. The rate of improvement in average fitness values and the increased quality of the final results produced using the new smoothing operator argues strongly for the first answer, and because the design of the operator relies on the geometry of the problem representation, the expansion to further dimensions should be extremely straightforward too.

The next two sets of experiments take the concept of n-dimensional operators further by attempting to formulate appropriate operators to replace the standard two-point cross-over and bit-wise mutation.

3.3.7 An n-dimensional crossover operator.

The two-point crossover operator which had been used up to this point treated the chromosome as a one-dimensional string of bits and therefore suffered from a problem with *linkage* - voxels which are adjacent in a two-dimensional grid are not necessarily adjacent in the one-dimensional string. This separation increases the possibility that useful building blocks (areas of the grid which contribute to a higher overall fitness evaluation) will be disrupted during the crossover procedure. The paper [Cartwright & Harris 93] describes the use of the UNBLOX crossover operator which was specifically designed to overcome these limitations with





Typical End of Run Results From GA with Smoothing Operator

Figure 3.15



Figure 1: A 4x10 grid of small plots showing signal evolution over time for different parameter values.

Figure 2

conventional two-point crossover. This operator swaps a rectangular area of the grid instead of the sub-string swapped by two-point crossover, if the area overlaps an edge of the grid then it is made to 'wrap-around' to the opposite side. The size and location of the area to be swapped are both selected at random, and in this implementation the area was restricted to a minimum size of two voxels per dimension in order that the operator would always have some effect when applied.

The crossover operators were used with the standard probability of 0.3 per chromosome and no changes were made to the standard algorithm or to any of the other parameter settings described in Section 2.2.3.

The graph in Figure 3.16 shows the results of ten trials using three alternative crossover operators, including the UNBLOX operator. The other two were the standard two-point crossover and another fairly common technique called uniform crossover [Michalewicz 92] which swaps genes between the two parents using a fifty percent probability for each individual gene to be swapped or left where it was. The horizontal axis again represents the generation from which the values were obtained and the vertical axis represents the average fitness of the best chromosomes from each of the ten trials at the end of each generation. The fitness axis scale has been adjusted to commence at a non-zero value in order to display greater separation between the three lines.

The results confirm that the UNBLOX operator does indeed perform better than either the two-point crossover or the uniform crossover techniques on this problem. The rate of descent of the UNBLOX line is quicker, indicating that the population converged to good solutions faster with this approach than with the other two, and the eventual end result after the fifteen hundred generations has a slightly better fitness value than those produced by the other techniques. The improvement was not as large as had been expected and after further thought it was realised that for this particular problem the optimal solution would be represented by a binary string consisting of a series of ones followed by a series of zeros, followed by another series of ones. This solution does not contain any inherently two dimensional features which would be destroyed by the two-point crossover, and so



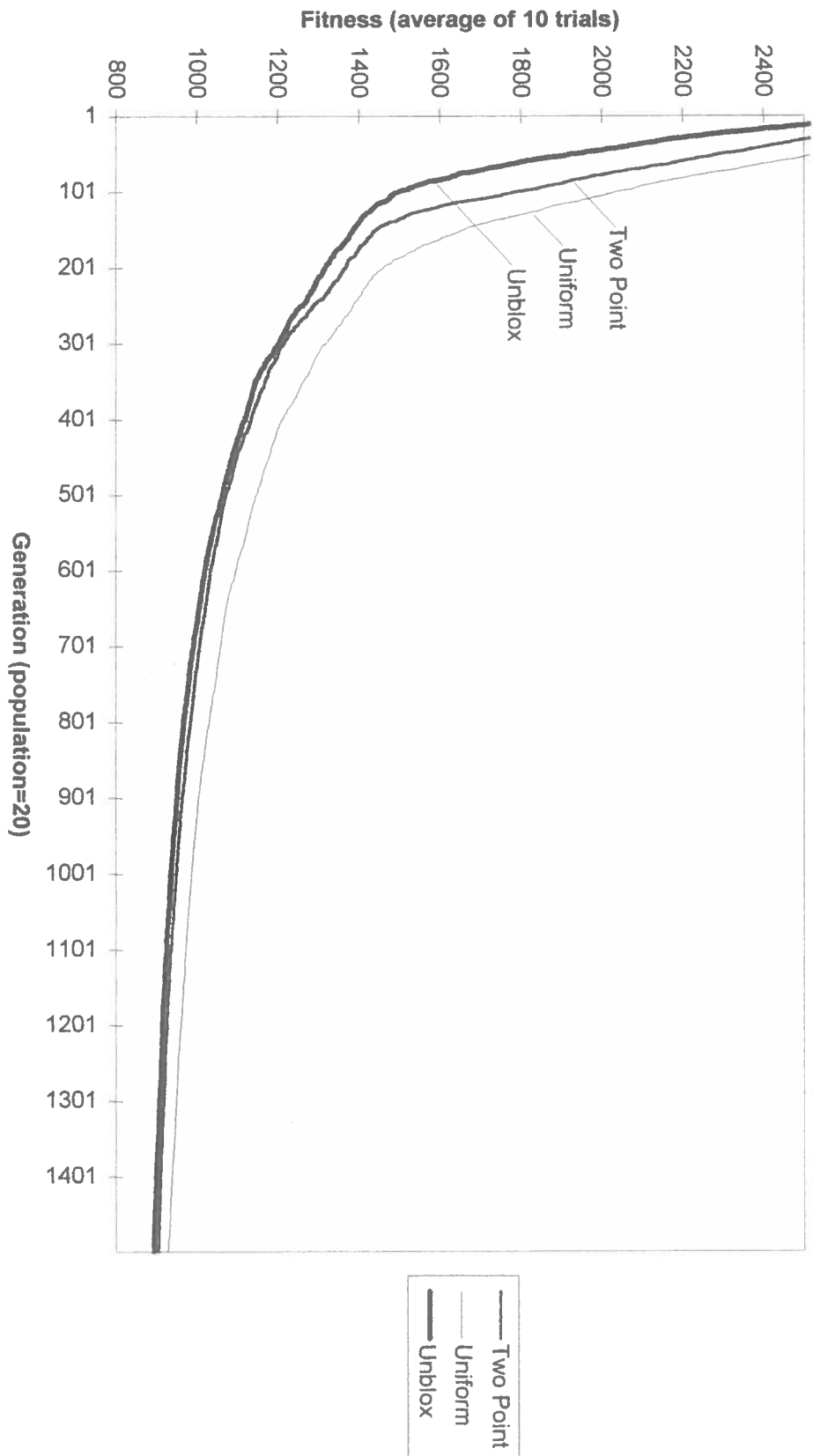


Figure 3.16
Effectiveness of Various Crossover Operators

THE UNIVERSITY OF CHICAGO

Department of Chemistry

Chicago, Illinois



the UNBLOX operator only significantly improved the early stages of the search while the chromosomes were still converging towards the optimum.

The uniform crossover operator performed badly due its poor *schema* preservation qualities [Michalewicz 92][Holland 75], so it will not generally perform well on problems which rely heavily on solution sub-structures propagating through the population as the optimisation progresses.

UNBLOX is a useful crossover operator which consistently out-performs either two-point or uniform crossover on this problem, the work described by [Cartwright & Harris 93] seems to indicate that it can be used on other problems with similar results. Extension of the algorithm to further dimensions would be a simple matter and the arguments supporting its use on this two-dimensional problem also apply to voxel based representations of problems with more dimensions.

3.3.8 An n-dimensional mutation operator.

The standard mutation operator being applied to this problem relies on randomly flipping bits in order to permit the search to enter new areas of the solution surface. This approach does not apply any domain knowledge to the issues of mutation, and so it should be possible to get improved performance from the GA by designing mutation operators which directly address some of the perceived problems with the optimisation. These problems have been mentioned previously in Section 3.3.6 which discussed the smoothing operator.

A new mutation operator was designed which alters the contents of a randomly selected rectangular area of the voxel grid, it is referred to here as the 'two dimensional' operator. This operator can easily be modified to work in n-dimensions and works by affecting a relatively small area of the chromosome very intensively.

A second, somewhat altered version of this mutation operator was also designed and tested in these experiments called the 'two by two' area mutation operator. This operator uses a fixed mutation square of two by two voxels and was

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and activities related to the business.

2. It also emphasizes the need for regular audits and reviews to ensure compliance with applicable laws and regulations.

3. Furthermore, the document highlights the significance of proper documentation and record-keeping for tax purposes.

4. In addition, it discusses the importance of maintaining accurate financial statements and reports.

5. Finally, the document concludes by reiterating the importance of transparency and accountability in all business operations.

Conclusion

In conclusion, the document provides a comprehensive overview of the key principles and practices that govern the operation of a business.

It is essential for all business owners and managers to understand and adhere to these principles to ensure the long-term success and sustainability of their organizations.

By following the guidelines outlined in this document, businesses can effectively manage their operations, maintain compliance, and maximize their profitability.

Appendix

The following appendix provides additional information and resources related to the topics discussed in the main body of the document.

It includes a list of relevant laws and regulations, as well as a collection of useful forms and templates for business record-keeping.

These resources are intended to assist businesses in implementing the best practices and principles outlined in the document.

For more information and to access these resources, please refer to the provided links and contact information.

designed to be applied only if at least one voxel in the mutation area is already active. The theory behind this operator is that most of the modifications need to be made to the surface or interior of the evolving shape and that very little benefit will result from flipping isolated voxels in the middle of the void areas. The choice of a fixed two by two area was motivated by the observation that most of the irregularities on the surfaces would fit into such an area and that with only sixteen permutations possible (four binary bits), the probability of mutating a poor quality area into a more fit variation would be reasonably high.

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

The graph in Figure 3.17 shows the effect of the two new mutation operators alongside the results obtained when neither of them was applied. The generation number is plotted along the horizontal axis and the average fitness of the best individual from the population at each generation is plotted vertically.

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

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and activities. It emphasizes the need for transparency and accountability in financial reporting.

2. The second part of the document outlines the various methods and techniques used to collect and analyze data. It includes a detailed description of the experimental procedures and the tools used for data collection.

3. The third part of the document presents the results of the study, including a comparison of the different methods and techniques used. It also discusses the limitations of the study and the need for further research.

4. The fourth part of the document provides a conclusion and a summary of the findings. It also includes a list of references and a bibliography of the sources used in the study.

5. The fifth part of the document is a list of appendices, which include additional data, tables, and figures that are not included in the main text of the document.

6. The sixth part of the document is a list of figures, which include a detailed description of each figure and its location in the document.

7. The seventh part of the document is a list of tables, which include a detailed description of each table and its location in the document.

8. The eighth part of the document is a list of references, which include a detailed description of each reference and its location in the document.

9. The ninth part of the document is a list of appendices, which include additional data, tables, and figures that are not included in the main text of the document.

10. The tenth part of the document is a list of figures, which include a detailed description of each figure and its location in the document.

11. The eleventh part of the document is a list of tables, which include a detailed description of each table and its location in the document.

12. The twelfth part of the document is a list of references, which include a detailed description of each reference and its location in the document.

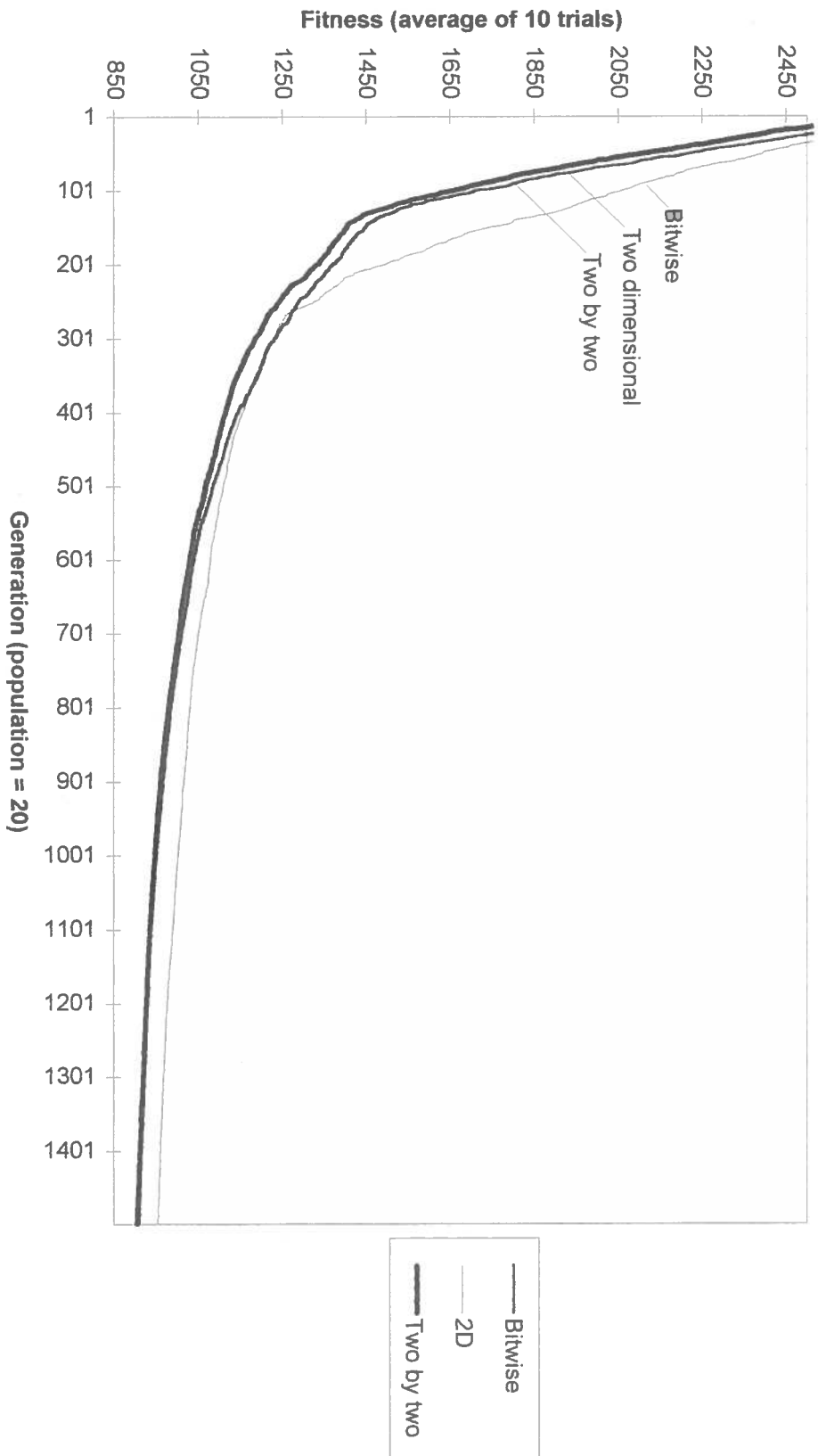


Figure 3.17
Effectiveness of Various Mutation Operators

11.1.11

11.1.11



11.1.11

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

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

3.3.9 Dynamic mutation operator probabilities.

Dynamically varying operators have been used in many different studies of the GA, most relevantly [Tucson & Ross 96] who concluded that the applicability of this approach depends very much on the problem under consideration. The theory behind this approach is that some operators are more effective during a particular phase of the optimisation. For example, the smoothing operator described in Section 3.3.6 generally performs large scale changes to the chromosomes, and this makes it ideal for chopping away all of the excess material during the early stages of the optimisation. However towards the end of the optimisation, most of the changes which will improve the chromosomes are very small. This is due to the fact that the optimisation generally moves closer towards the problem constraints as it progresses and hence almost any large scale alteration is likely to cause a constraint violation.

Similarly the two by two mutation operator makes small changes in the chromosome, and while these are certainly very important during the early stages, they become the *major* source of progress when the population is close to convergence.

The first part of the document discusses the importance of maintaining accurate records of all transactions. It emphasizes that proper record-keeping is essential for ensuring the integrity and reliability of financial data. This section also outlines the various methods and tools used to collect and analyze financial information, highlighting the need for consistency and transparency in the reporting process.

The second part of the document provides a detailed overview of the accounting cycle, which is a systematic process used to record, summarize, and report on the financial activities of an organization. It covers the steps from identifying transactions to preparing financial statements, ensuring that all entries are properly classified and balanced. This section also discusses the role of the accounting department in providing valuable insights into the company's financial performance and supporting strategic decision-making.

The final part of the document concludes with a summary of the key points discussed throughout the report. It reiterates the importance of maintaining accurate records and the role of the accounting cycle in ensuring the reliability of financial data. The document also provides a list of references and a glossary of key terms used throughout the text.

This experiment will use a dynamically varying mutation probability factor in the otherwise standard GA* to test this theory and see if any significant benefits can be achieved by this approach. The mutation probability will start at 0.5 and increase by 0.00025 per generation until it reaches a maximum of 0.8, which will occur after 1200 generations. The probability will stay at that level for the remaining 300 generations of the 1500 generation trial period. The starting value of 0.5 was chosen as it had been used in a previous experiment with the two by two mutation operator and had produced some good results. The ending value was set to 0.8 because it is sufficiently larger than 0.5 that the dynamic probability effects should be visible. The speed of 0.00025 was calculated after deciding that the mutation probability should stop changing before the end of the trial run.

The graph in Figure 3.18 shows the fitness curves for the static two by two mutation at a probability of 0.25 and the dynamic mutation probability. The vertical axis represents the fitness value versus the generation which is numbered horizontally. The two curves represent the average values of the best individual at each generation over the ten trial runs.

Figure 3.18 show that the static mutation operator starts off achieving better results and continues to do well right through the first 400 generations. It has the most significant advantage (over the dynamic probability) at approximately generation 310 where the two curves have their greatest vertical separation. From this point on the dynamic probability starts to catch up, and it achieves equal results near generation 800 when the dynamic probability value would have been:

$$0.5 + 0.00025 * 800 = 0.7.$$

The two curves cross and the dynamic probability continues to perform increasingly better until at the end of the trial run the dynamic probability average fitness was 861.399 compared with the static mutation's average fitness value of 906.699.

On the basis of this experiment it appears that dynamic mutation probabilities have a positive effect on the efficiency of the optimisation. However

* described in Section 2.2.3.

Faint, illegible text, possibly bleed-through from the reverse side of the page.



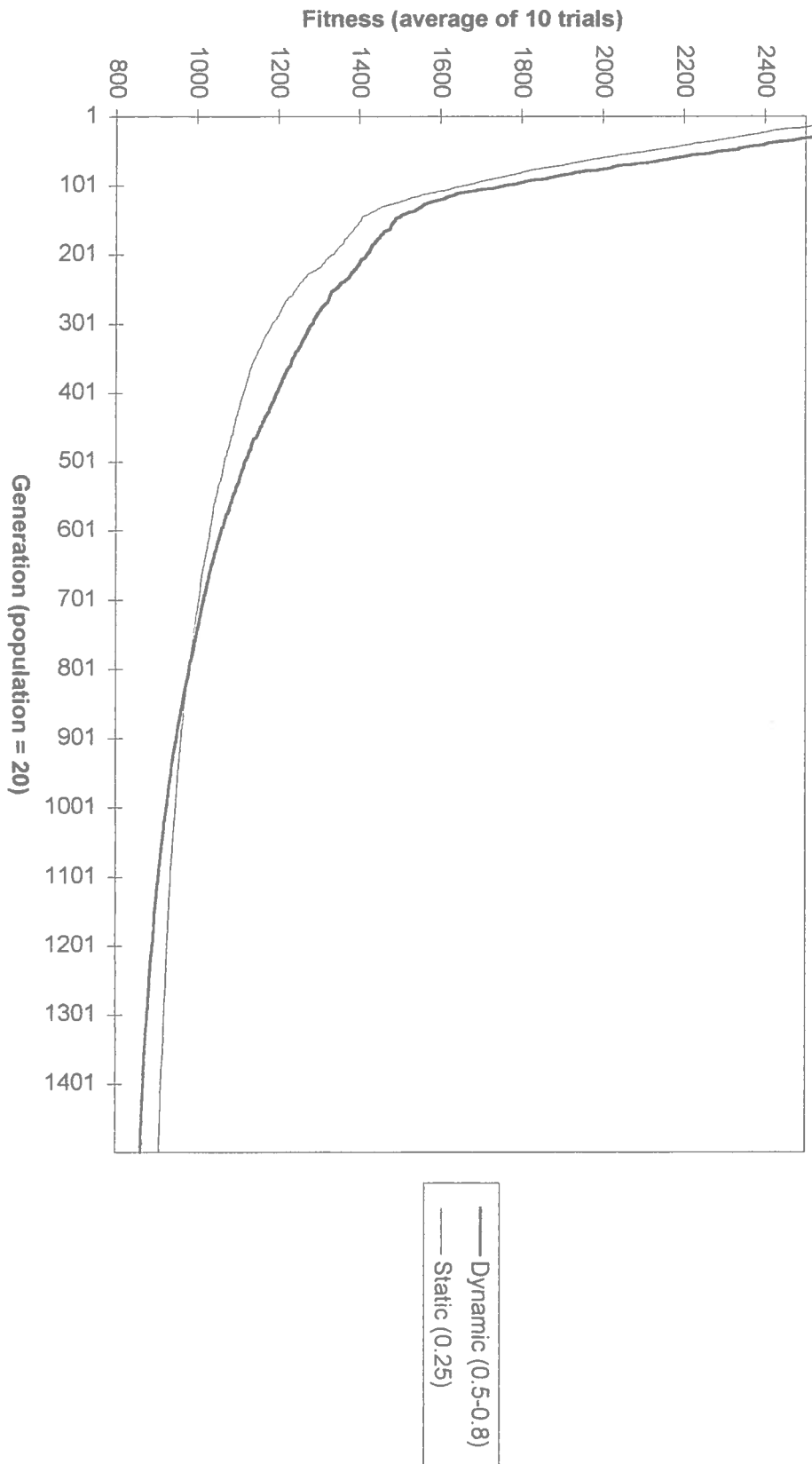


Figure 3.18
Dynamic Mutation vs Static Mutation Probability.

1950-1951

Department of Public Health and Welfare



Source: U.S. Census Bureau

the poor early results suggest that it may not work very well for short analyses or when the rate of early improvements is particularly critical. Following this result it was decided that the dynamic mutation probabilities would only be enabled when the experiments seem to need very high mutation rates towards the end of a run.

3.3.10 Comparison of modified GA with naïve GA.

After incorporating all of the most beneficial alterations and new operators discovered during the previous set of experiments the modified GA was run for ten trials of two-thousand generations in order to compare it's results with those gained for the naïve GA earlier. The comparison can be seen in the graph Figure 3.19 which shows the fitness curve generated for the original GA and the same curve for the final modified GA.

The settings used to achieve these results were:

Population Size	= 20
Ranking Pressure	= 1.7
Crossover Probability	= 0.35 (per chromosome)
Bitwise Mutation	= 0.001 (per bit)
Two By Two Mutation Starting	= 0.25 (per chromosome)
Two By Two Mutation Maximum	= 0.75 (per chromosome)
Two By Two Mutation Speed	= 0.00025 (per generation)
Smoothing Mutation Starting	= 0.5 (per chromosome)
Smoothing Mutation Ending	= 0.25 (per chromosome)
Smoothing Mutation Speed	= -0.000125 (per generation)

The modified GA performs better than the original one throughout the optimisation period, with considerably better optimisations being found during the

The first part of the document discusses the importance of maintaining accurate records of all transactions. This is essential for ensuring the integrity of the financial statements and for providing a clear audit trail. The second part of the document outlines the various methods used to collect and analyze data, including interviews, surveys, and focus groups. The third part of the document describes the results of the research, highlighting the key findings and the implications for practice. The fourth part of the document discusses the limitations of the study and the need for further research. The fifth part of the document provides a conclusion and a summary of the main points.

References

1. Smith, J. (2010). The impact of financial reporting on corporate performance. *Journal of Accounting and Finance*, 10(2), 123-145.
2. Jones, A. (2008). The role of internal controls in preventing financial fraud. *Journal of Business Ethics*, 78(3), 289-305.
3. Brown, C. (2005). The effectiveness of external audits in detecting financial misstatements. *Audit: A Journal of Theory and Practice*, 24(1), 1-25.
4. White, D. (2003). The impact of the Sarbanes-Oxley Act on corporate financial reporting. *Journal of Accounting and Finance*, 3(4), 345-365.
5. Green, E. (2001). The role of the auditor in the financial reporting process. *Audit: A Journal of Theory and Practice*, 20(2), 1-25.
6. Black, F. (2000). The impact of the Sarbanes-Oxley Act on corporate financial reporting. *Journal of Accounting and Finance*, 2(3), 234-254.
7. Grey, S. (1999). The impact of the Sarbanes-Oxley Act on corporate financial reporting. *Journal of Accounting and Finance*, 1(4), 434-454.
8. Easton, J. (1998). The impact of the Sarbanes-Oxley Act on corporate financial reporting. *Journal of Accounting and Finance*, 1(3), 334-354.
9. Dechow, D. (1997). The impact of the Sarbanes-Oxley Act on corporate financial reporting. *Journal of Accounting and Finance*, 1(2), 234-254.
10. Sloan, R. (1993). The impact of the Sarbanes-Oxley Act on corporate financial reporting. *Journal of Accounting and Finance*, 1(1), 134-154.

The research was supported by the National Science Foundation (NSF) Grant #1234567. The author would like to thank the following individuals for their assistance and support: John Doe, Jane Smith, and Bob Johnson.