2

General Review of Relevant Work

2.1 Introduction

The work described in this thesis is the first attempt to create a computer system capable of evolving a wide range of solid object designs from scratch. Since there are no directly equivalent systems to compare with the work described in this thesis, this chapter will review and appraise significant research in all areas related to this subject.

Research relevant and related to the evolution of solid object designs can be divided into six main categories:

(i) Engineering design theory

(ii) Natural evolution of designs

(iii) Genetic algorithms

(iv) The optimisation of existing designs

(v) The creation of shapes and images by computers

(vi) The creation of new designs by computers

Further appraisal of literature concerning the more detailed problems encountered during the development of the system (e.g. the representation of solid objects, multiobjective optimisation within GAs, variable-length chromosomes, and the evaluation of specific types of design) is performed in later chapters.

2.2 Engineering Design Theory

2.2.1 Classification of Human Design

Before attempting to automate the design process, it seems sensible to try to define and understand the human design process. Unfortunately, there is no universally accepted definition of engineering design (Dym and Levitt, 1991). However, for the purposes of this work, the process of design can be adequately summarised by the words of Dym:

> Engineering Design is the systematic, intelligent generation and evaluation of specifications for artefacts whose form and function achieve stated objectives and satisfy specified constraints. (Dym and Levitt, 1991)

Although the precise definition of design is often debated, there is perhaps more agreement concerning the functional composition of design. Hence, the engineering design process performed by a human designer can be generally summarised as consisting of the following stages (Pahl and Beitz 1988, Dym and Levitt 1991, Goldberg 1991b, Pham and Yang 1993):

1. conceptual or preliminary design
2. detailed design
3. evaluation or analysis
4. iterative redesign, if the evaluation results are unsatisfactory

To date, computers have been used successfully for all of these stages except the first: conceptual or *creative* design (Dym and Levitt 1991, Goldberg 1991b). As stated by Goldberg: "The creative processes of engineering design have long been regarded as a black art. While the engine of analysis steamrolls ever forward, our understanding of conceptual design seems locked in a timewarp of platitudes, vague design procedures, and problem-specific design rules." (Goldberg, 1991b).

Modern research has started to provide the beginnings of a more rigorous theory of human conceptual design (French 1994, Goldberg 1991b). Discussions viewing design as evolution (French, 1994) and evolution as design (Thompson 1961, Tributsch 1982) are common. Goldberg goes further by comparing the work of inductive designers, who integrate combinations of previous designs in an attempt to create improved designs, with the similar workings of the genetic algorithm. He concludes that "...genetic algorithms can be thought of as a bounding model of discriminative and recombinative invention. As human designers recombine bits and pieces of previous designs to form new, possibly better proposals, GAs recombine bits and pieces of artificial chromosomes to search for globally optimal solutions." (Goldberg, 1991b).

By allowing the system to create new designs from scratch, this project is attempting to create a system capable of generating new conceptual designs. However, as stated by Pham: "While the above-mentioned four major stages [of engineering design] are usually recognisable within a design process, in general no clear boundaries between them can be defined." (Pham & Yang, 1993). Hence, with no easy way to determine when creative design stops and the optimisation of the design begins, the system being created for this project will perform the whole design process without explicitly dividing the process into stages.

2.2.2 Evolution of Human Designs

It is clear that human designs have progressed over time, but can this progressive refinement accurately be described as evolution? The answer is yes - in a very real sense our designs evolve (French, 1994). From the moment of their construction, designed objects must perform their function to survive. If they are found to be inadequate, they are discarded and forgotten quickly. As with living creatures, our artefacts have limited lifespans before they wear out. In a process very similar to reproduction in nature, these objects are replaced by new artefacts that use or 'inherit' design features from existing designs in addition to the occasional new invention or 'mutation' (Goldberg 1991b, French 1994). Just as in nature, where the environment is constantly changing (e.g. new predators, variations in terrain or climate), the problem that the design is intended to solve is often constantly changing (Dym and Levitt, 1991). For example, at first the only purpose of the car was to carry passengers, then comfort and safety became important, and today a car must also be economical and not pollute the environment.

As the 'population' of new designs grows, so 'selection pressure' is increased. In other words, the more designs that exist to solve a single problem, the quicker some designs will be discarded. Perhaps one of the most extreme examples of this is with computer processors, where designs become obsolete and are replaced with faster, more efficient models barely eighteen months after being introduced. Likewise, in nature, the more trees in a wood that attempt to consume the limited resource of sunlight, the quicker the less tall varieties will die - the selection pressure is so strong that evolution is compelled to create trees that grow evermore faster and taller. Moreover, just as the struggle between predator and prey can trigger rapid evolution and counter-evolution in nature, the struggle between armies often triggers an explosion in new designs of weapons and counter-weapons in war (Thompson 1961, French 1994).

2.3 Natural Evolution of Designs

2.3.1 Creatures as Designs

Just as human designs evolve, it is often conversely argued that natural evolution produces designs. This is perhaps argued most eloquently by Dawkins, who compares natural selection with a blind watchmaker, "...blind because it does not see ahead, does not plan consequences, has no purpose in view. Yet the living results of natural selection overwhelmingly impress us with the appearance of design as if by a master watchmaker, impress us with the illusion of design and planning." (Dawkins 1986, p.21).

Indeed, before the revolutionary theories of Darwin (Darwin, 1859), every living thing was widely believed to have been the direct creation of God. Some still hold this view today, often using arguments stating that, for example, all the parts of an eye must have been created together, for an eye without a cornea, lens, and retina all working in perfect synchrony is of little survival value (Hitching, 1982). In other words, it is argued that a more primitive eye, with say 5 per cent of the function of our eye, would be of no use to the organism and thus could never evolve to the intricate complexity of our eyes. However, as Dawkins argues vehemently: "Vision that is 5 per cent as good as yours or mine is very much worth having in comparison with no vision at all. So is 1 per cent vision better than total blindness. And 6 per cent is better than 5, 7 per cent better than 6, and so on up the gradual, continuous series." (Dawkins 1986, p.81). Indeed, creatures with more primitive eyes (e.g. insects), and creatures with more advanced eyes than our own (e.g. hawks) are well known in nature, illustrating that there is great survival value in having eyes of any degree of function (Dawkins, 1982). Consequently, today the widely accepted view is that the extraordinary designs of life were evolved, and not consciously created (Dawkins, 1995).

2.3.2 Designs Evolved in Nature

From the perspective of engineering design, it is clear that nature far exceeds us in countless aspects. For example, one of the most common and familiar living things we see every day is a marvel of design: the tree. As described by French, the tree exhibits a number of elegant design solutions: structurally trees are stressed in tension at the outside and in compression in the middle, at all stages of growth to give great strength overall. A cascade of chemical reactions driven by the sun's radiation provides carbohydrates for 'food'. Water is ingeniously *pulled* up from the ground in long free-hanging threads, rather than being pumped or sucked. Numerous control systems open and close the millions of tiny pores or stomata in the leaves (which collect carbon dioxide), to prevent loss of water (French, 1994).

A more widely appreciated design in nature is that of bats. As described by Dawkins, in addition to their unique position as the only flying mammals, bats have extraordinary sonar.

Many types of bats have independently developed sonar, and some have perfected it to such an extent that their faces have become highly distorted to help receive and direct the ultrasound echoes to their sensitive ears. Because their ears are so sensitive, bats have muscles to 'disengage' their ears while they emit their loud echo-location calls. The calls are ultrasound as the higher frequency helps to resolve smaller objects and hence provides clearer detail in the returning echoes to the bat. Usually pulses of sound are emitted, for some species at 10 per second in normal flight and up to 200 a second when homing in on prey, to provide quicker updates of the world. In some bats, the frequency of each pulsed call is changed, like a high-speed wolf whistle (similar to our chirp radar) to allow them to detect echoes of echoes. In addition to this, horseshoe bats rapidly move their outer ear flaps to gain still further information from the returning sound. Yet despite the amazingly complex design of their echo-location, no bat is ever confused by the echo-location of the thousands of other bats it lives with. (Dawkins, 1986)

Clearly, even from just two brief descriptions of the countless designs produced by natural selection, evolution has much to offer human designers. Natural evolution forms the inspiration for this work, and it is hoped that an artificial evolutionary design system using a genetic algorithm will have some of the potential of natural evolution, and will be able to create novel, elegant and successful designs in the same way.

2.4 Genetic Algorithms

2.4.1 Algorithm

Natural evolution acts through large populations of creatures which reproduce to generate new offspring that inherit some features of their parents (because of random *crossover* in the inherited chromosomes) and have some entirely new features (because of random *mutation*). Natural selection (the weakest creatures die, or at least do not reproduce as successfully as the stronger creatures) ensures that, on average, more successful creatures are produced each generation than less successful ones. As described previously, evolution has produced some astonishingly varied, yet highly successful forms of life. These organisms can be thought of as good 'solutions' to the problem of life. In other words, evolution optimises creatures for the problem of life.

In the same way, within a genetic algorithm (GA) a population of solutions to the problem is maintained, with the 'fittest' solutions (those that solve the problem best) being favoured for 'reproduction' every generation, during an otherwise random selection process. 'Offspring' are then generated from these fit parents using random crossover and mutation operators, resulting in a new population of fitter solutions (Holland, 1975).

Genetic algorithms differ from traditional algorithms in four ways (Goldberg, 1989):

1. GAs usually work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge
4. GAs use probabilistic transition rules, not deterministic rules.

Coded parameters are normally referred to as *genes*, with the values a gene can take being known as *alleles*. A collection of genes in one individual of the population is held internally as a string, and is often referred to as a *chromosome*. The entire coded parameter set of an individual (which may be anything from a single gene to a number of chromosomes) is known as the *genotype*, while the solution that the coded parameters define is known as the *phenotype*.

The simple or canonical GA is summarised in figure 2.1. Typically, populations are initialised with random values. The main loop of the algorithm then begins, with every member of the population being evaluated and given a fitness value according to how well it fulfils the objective or fitness function. These scores are then used to determine how many copies of each individual are placed into a temporary area often termed the 'mating pool' (i.e. the higher the fitness, the more copies that are made of an individual). Two parents are then randomly picked from this area. Offspring are generated by the use of the crossover operator which randomly allocates genes from each parent to each offspring. For example, given two parents: 'ABCDEFG' and 'abcdefg', and a random crossover point of, say, 4, the two offspring generated by the simple GA would be: 'ABCDefg' and 'abcdEFG'. Mutation is then occassionally applied (with a low probability) to offspring. When it is used to mutate an individual, typically a single allele is changed randomly. For example, an individual '111111' might be mutated into '110111'. Using crossover and mutation, offspring are generated until they replace every parent in the population. This entire process of evaluation and reproduction then continues until either a satisfactory solution emerges or the GA has for run a specified number of generations. (Holland 1975, Goldberg 1989, Davis 1991, Fogel 1995).

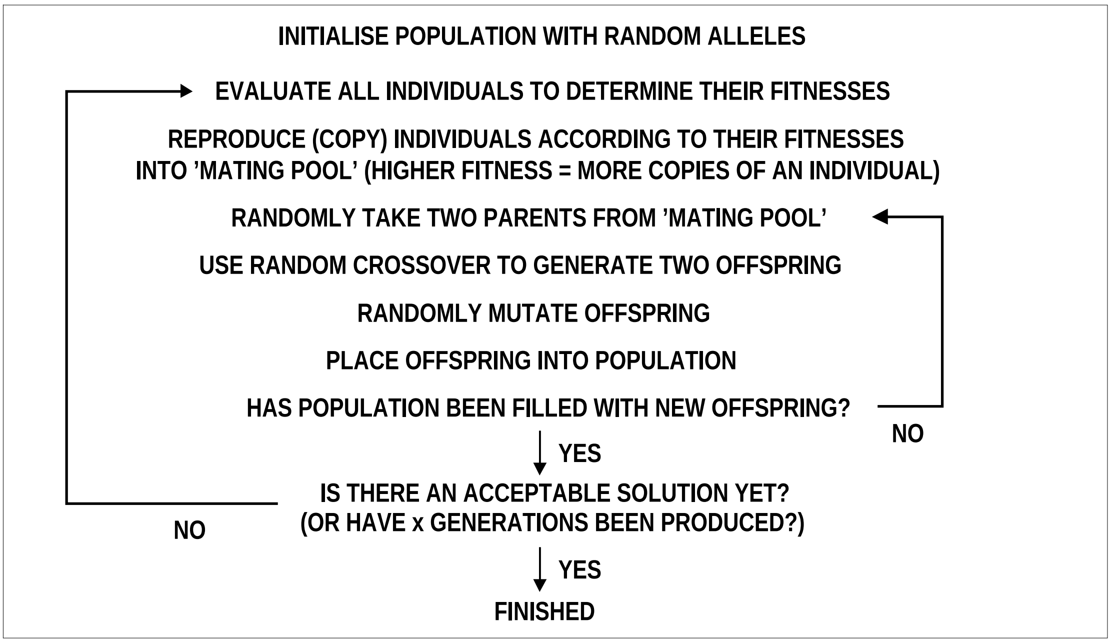

Fig. 2.1 The simple genetic algorithm.

The simple GA is just that - very simple and a little naive. This GA is favoured by those that try to theoretically analyse and predict the behaviour of genetic algorithms, but in reality, typical GAs are usually more advanced. Common features include: more realistic natural selection (i.e. automatic selection without human guidance), ability to detect when evolution ceases, and overlapping populations or elitism (where some fit individuals can survive for more than one generation) (Davis, 1991). Because of this improved analogy with nature, the term *reproduction* is normally used as it is in biology to refer to the entire process of generating new offspring, encompassing the crossover and mutation operators. (This is in contrast to the somewhat confusing use of the word 'reproduction' to mean an explicit copying stage within the simple GA).

2.4.2 Theory

The genetic algorithm is perhaps the most well-known of all evolution-based search algorithms. GAs were developed by John Holland over twenty-five years ago in an attempt to explain the adaptive processes of natural systems and to design artificial systems based upon these natural systems (Holland 1973, 1975). Whilst not being the first algorithm to use principles of natural selection and genetics within the search process (others include Evolutionary Programming: Fogel, 1963 and Evolutionsstrategie: Rechenberg, 1973), the genetic algorithm is today the most widely used. More experimental and theoretical analyses have been made on the workings of the GA than any other evolutionary algorithm. Moreover, the genetic algorithm (and enhanced versions of it) resembles natural evolution more closely than most other methods.

Having become widely used for a broad range of optimisation problems in the last ten years (Holland, 1992), the GA has been described as being a "search algorithm with some of the innovative flair of human search" (Goldberg, 1989). GAs are also very forgiving algorithms - even if they are badly implemented, or poorly applied, they will often still produce acceptable results (Davis, 1991). GAs are today renowned for their ability to tackle a huge variety of optimisation problems (including discontinuous functions), for their consistent ability to provide excellent results and for their robustness (Holland 1975, Goldberg 1989, Davis 1991, Fogel 1995).

For a search algorithm to be robust, it must be capable of producing good solutions to a broad range of problems. In his book, Goldberg compares traditional search methods (calculus-based, enumerative, and random) with genetic algorithms (Goldberg, 1989). He concludes: "while our discussion has been no exhaustive examination of the myriad methods of traditional optimisation, we are left with a somewhat unsettling conclusion: conventional search methods are not robust." (Goldberg, 1989, p.5). The general view of many researchers appears to be in agreement with this, although it is often debated exactly which algorithm would give the best results for specific problems. Typically, it is argued that a traditional algorithm designed specifically for a problem will provide better results for that problem than a GA could, but that a GA will provide good solutions for a much broader selection of problems, compared to such problem-specific methods (Fogel, 1995).

Significantly, the design system described in this thesis requires an algorithm capable of consistently finding good solutions to a range of design problems, i.e. the evaluation software specifying new design tasks for the system could consist of literally any type of function. For this reason it is felt that the genetic algorithm is an appropriate choice to form the core search-engine of the system.

Whilst there is no formal proof that the GA will always converge to an acceptable solution to a given problem, a variety of theories exist (Holland 1975, Kargupta 1993, Harris 1994), the most accepted of these being Holland's Schema Theorem and the Building Block Hypothesis (Holland 1975).

Briefly, a *schema* is a similarity template describing a set of strings (or chromosomes) which match each other at certain positions. For example, the schema *10101 matches the two strings {110101, 010101} (using a binary alphabet and a metasymbol or *don't care* symbol *). The schema *101* describes four strings {01010, 11010, 01011, 11011}. As Goldberg (1989) elucidates, in general, for alphabets of cardinality (number of alphabet characters) k, and string lengths of l characters, there are $(k + 1)^l$ schemata.

The *order* of a schema is the number of fixed characters in the template, e.g. the order of schema *1*110 is 4, and the order of schema *****0 is 1. The *defining length* of a schema is the distance between the first and last fixed character in the template, e.g. the defining length of 1****0 is 5, the defining length of 1*1*0* is 4, and the defining length of 0***** is 0.

Holland's Schema Theorem states that the action of reproduction, crossover and mutation within a genetic algorithm ensures that schemata of short defining length, low order and high fitness exponentially increase within a population (Holland, 1975). Such schemata are known as building blocks.

The building block hypothesis suggests that genetic algorithms are able to evolve good solutions by combining these fit, low order schemata with short defining lengths to form better strings (Goldberg, 1989). However, this still remains an unproven (though widely accepted) hypothesis.

2.4.3 Analyses

Experimental results show that for most GAs (initialised with random values), evolution makes extremely rapid progress at first, as the diverse elements in the initial population are combined and tested. Over time, the population begins to converge, with the separate individuals resembling each other more and more (Davis, 1991). Effectively this results in the GA narrowing its search in the solution-space and reducing the size of any changes made by evolution until eventually the population converges to a single solution (Goldberg, 1989). When plotting the best fitness value in each new population against the number of generations, a typical curve emerges, fig 2.2 (Parmee and Denham, 1994).

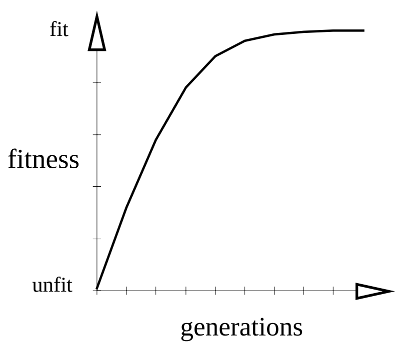

Fig. 2.2 Typical curve of evolving fitness values over time.

Theoretical research to investigate the behaviour of the various varieties of GAs for different problems is growing rapidly, with careful analyses of the transmission of schemata being made (De Jong 1975, Kargupta 1993). The use of Walsh function analysis (Goldberg 1989, Goldberg and Rudnick 1991, Deb et. al. 1993) and Markov Chain analysis (Horn 1993, Mahfoud 1993a, 1993b) has led to the identification of some 'deceptive' and 'hard' problems for GAs (Deb and Goldberg 1992, 1993, Goldberg, Horn & Deb 1992a, 1992b).

2.4.4 Advanced Genetic Algorithms

When applying GAs to highly complex applications, some problems do arise. The most common is *premature convergence* where the population converges early onto non-optimal local minima (Davis, 1991). Problems are also caused by deceptive functions, which are, by definition, 'hard' for most GAs to solve. In addition, noisy functions (Goldberg et. al. 1992a, 1992b, Lomborg 1991) and the optimisation of multiple criteria within GAs can cause difficulties (Fonseca and Fleming, 1995). In an attempt to overcome such problems, new, more advanced types of GA are being developed (Goldberg, 1993). These include:

- Parallel GAs, where multiple processors are used in parallel to run the GA (Adeli and Cheng 1994c, Levine 1994).
- Distributed GAs, where multiple populations are separately evolved with few interactions between them (Whitley and Starkweather 1990, Mühlenbein 1992)
- GAs with niching and speciation, where the population within the GA is segregated into separate 'species' (Horn 1993, Horn and Nafpliotis 1993, Horn et. al. 1994).
- Messy GAs (mGA), which use a number of 'exotic' techniques such as variable-length chromosomes and a two-stage evolution process (Deb 1991, Deb and Goldberg, 1991).
- Multiobjective GAs (MOGAs), which allow multiple objectives to be optimised with GAs (Schaffer 1985, Srinivas and Deb 1995, Bentley and Wakefield 1996).
- Hybrid GAs (hGAs), where GAs are combined with local search algorithms (George 1994, Radcliffe and Surrey 1994a).

- Structured GAs (sGAs), which allow parts of chromosomes to be switched on and off using evolveable 'control genes' (Dasgupta and McGregor 1992, Parmee and Denham 1994).

2.4.5 Applications of Genetic Algorithms

The genetic algorithm has only become popular for optimisation problems in the last ten or fifteen years (Holland 1992, Goldberg 1994). However, in that time, literally thousands of different problems in many different areas have had solutions successfully optimised by GAs. As will be shown in the following sections of this chapter, one of the most common problem areas is design, with perhaps more design problems having been optimised by GAs than any other type of problem. To avoid repetition, there follows a brief list of just some of the problems tackled by GAs that are *not* directly related to the design of physical objects:

- Machine learning (Goldberg 1989, Goldberg et. al. 1992b, Holland 1992, Smith and Goldberg 1992a, Horn et. al. 1994).
- Strategy acquisition (Greffenstette, 1991).
- Ordering problems (Kargupta et al. 1992, Schaffer & Eshelman 1995).
- Control systems (Krishnakumar and Goldberg 1992, Lansbury et al. 1992, Husbands et al. 1996, Morris and Martin 1996).
- Fault-tolerant systems (Thompson, 1995).
- Scheduling (Yamada and Nakano, 1995).
- Data mining (Radcliffe and Surrey 1994b).
- Artificial life (Bedau et. al. 1992, Cliff et. al. 1994, Sims 1994a, 1994b).
- Game playing (Axelrod 1987, Lomborg 1991, Adachi and Kazuhiro 1992, Albin 1992, Mühlenbein 1991, Vincent 1992).
- Set covering and partitioning (Beasley and Chu 1994, Levine 1994).
- Signal timing (Foy et. al., 1992).
- Composition of music (Horner and Goldberg, 1991)

2.5 The Optimisation of Existing Designs

2.5.1 Application-Specific Optimisation of Designs

The development of non-generic optimisation systems, capable of optimising explicitly parameterised parts of existing designs, has been attempted for many years (Dym and Levitt 1991, Adeli 1994). Traditionally, calculus-based approaches have been favoured, for example the designs of structures such as bridges or transmission towers have been optimised using optimality criteria methods (Rozvany and Zhou, 1994).

These and many other techniques have been tried to optimise designs (Dym and Levitt, 1991), but in recent times the use of GAs and similar adaptive search methods for such problems has become widespread (Parmee and Denham 1994, Rayward-Smith 1995). Results from comparisons between GAs and other algorithms inevitably vary depending on the application, but for a typical design optimisation task, it seems that the GA can outperform other traditional algorithms. For example, Tennant compares a GA with simulated annealing and downhill simplex algorithms, in the design of microwave absorbers, with only the GA locating the maximum of the function correctly every run (Tennant and Chambers, 1994). Similar results were found by Husbands, who compared a GA with simulated annealing and gradient descent algorithms in the optimisation of the structural design of a wing boxes (Husbands et. al., 1996).

Today, numerous examples of the optimisation of designs exist, most using GAs or other adaptive search methods. For example, computers have been used to optimise:

- Adaptive antenna arrays and radar absorbers (Chambers et. al., 1995).
- Airfoil and aircraft geometries (Bouchard et. al. 1988, Bramlette and Bouchard 1991, Obavashi and Takanashi 1995, Husbands, Jermy, McIlhagga, & Ives 1996).
- Analogue filters (Reeves et al., 1994).
- Architecture of buildings (Glaskin 1995, describing the work of Cawthorne).
- Building heating systems (Dickinson and Bradshaw, 1995).
- Aesthetic bridges (Furuta et. al., 1995).

- Low noise engine blocks (Fisher, 1995).
- Encastré beams (McMahon et al., 1994).
- Floorplans (Koakutsu et al., 1992).
- Sizes of gas pipes (Boyd, 1994).
- Hydraulic networks (Savic and Walters 1994, Donne et al., 1994).
- Microwave absorbing materials (Tennant and Chambers, 1994).
- Minimum length nozzle design (King et al. 1993).
- Satellite Booms (Keane and Brown, 1996).
- Servo and micro motors (Hameyer and Belmans, 1996).
- Spacecraft systems (Garipov et. al., 1994).
- Structural topology (Shankar and Hajela 1991, Adeli and Cheng 1994a, 1994b, 1994c, Rozvany and Zhou 1994).
- Transmission towers (Cai and Thierauf, 1996).
- VLSI layouts (Schnecke and Vornberger 1995).

Many of these applications are simple demonstrations and are often implemented crudely, with seemingly little knowledge of the genetic algorithm used. However, some systems have been used with considerable success to optimise real-world problems. For example, as described by Holland (1992), a design of a high-bypass jet engine turbine was typically optimised in eight weeks by an engineer; the genetic algorithm optimised a design in only two days, "with three times the improvements of the manual version".

Whilst the wide variety of applications being tackled shows that computers can be used to successfully evaluate and optimise many different types of design, every one of these optimisation systems, without exception, suffers from two major drawbacks. Firstly, every one can only optimise existing designs - it would be quite impossible to use any of them to create a new design. Secondly, every one is application-specific - they can only optimise the single type of design they were created to optimise, and no others.

2.5.2 Generic Optimisation of Designs

Generic design optimisation (i.e. the optimisation of more than one type of design by a single system) is a less common subject for research. Kant's 'CONFIG' system (Kant, 1988) allows a variety of different designs to be analysed, but performs no optimisation. Similarly, Libes' 'EXPECT' (Libes, 1990) allows multiple application analysis tools to be linked, but does not incorporate optimisation. Bouchard's 'Engineer's Associate' (Bouchard et. al., 1988) provides a limited generic framework to work with systems that can be represented by equations. Culley's general purpose optimisation system 'GPOS' (Culley and Wallace, 1994) consists of a toolbox of optimisation algorithms, capable of optimising a range of different applications (once interfaced appropriately).

However, Tong's 'Engineous' (Tong, 1992) is perhaps the most successful generic system, having been demonstrated on over 20 design optimisation tasks, including the optimisation of 3D turbine blades, cooling fans, DC motors, power supplies and a nuclear fuel lattice. A large portion of the system consists of complex interfacing software to allow the use of existing design evaluation packages. The system relies heavily on expert systems containing much application-specific knowledge to guide the evolution of a GA, which has to be changed for every new application. Tong claims that "the current version of Engineous has demonstrated the profound impact such a system can have on productivity and performance" (Tong, 1992).

2.6 The Creation of Shapes and Images by Computers

2.6.1 Evolution of Art

The use of computers to create art (again, often with GAs and similar adaptive search algorithms) is growing in popularity amongst some artists. For example, Stephen Todd and the artist William Latham have successfully evolved many three dimensional 'artistic' images and animations (Todd and Latham, 1992). Their system uses an elegant artificial embryology known as 'Form Grow' to allow the definition of intricate three-dimensional shapes and textures. These shapes are composed of a number of primitive shapes (e.g. spiral, sphere, torus) selected to give a distinctive 'biological' appearance to the images, see fig. 2.3. A simple evolution-based search algorithm, known as 'Mutator' (not unlike an Evolutionary Strategy (ES), Rechenberg, 1973) allows the creation and modification of the shapes, directed by the user of the software. This work has now been redeveloped by Latham and Atkinson, to create a commercially available product known as 'Organic Art' (as described by Boxer, 1996).

A similar system was recently developed by Husbands, Jermy, McIlhagga and Ives. This uses superquadrics as primitive shapes in combination with a recursive shape description language, allowing the specification of a variety of three-dimensional shapes. Guided by a human observer, a distributed genetic algorithm was used to evolve free-form shapes resembling corkscrews or propellers (Husbands, Jermy, McIlhagga, & Ives, 1996).

Another example is the work of John Mount, who shows his 'Interactive Genetic Art' on the internet (at http://robocop.modmath.cs.cmu.edu:8001). He employs a GA to modify fractal equations that define two dimensional images. Visitors are then invited to vote on how attractive each image is, which provides the GA with fitness scores to allow a new generation of theoretically more attractive images to be produced.

Fig. 2.3 Evolved art: ’Breeding Forms on the Infinite Plane’

(reproduced with kind permission of William Latham)

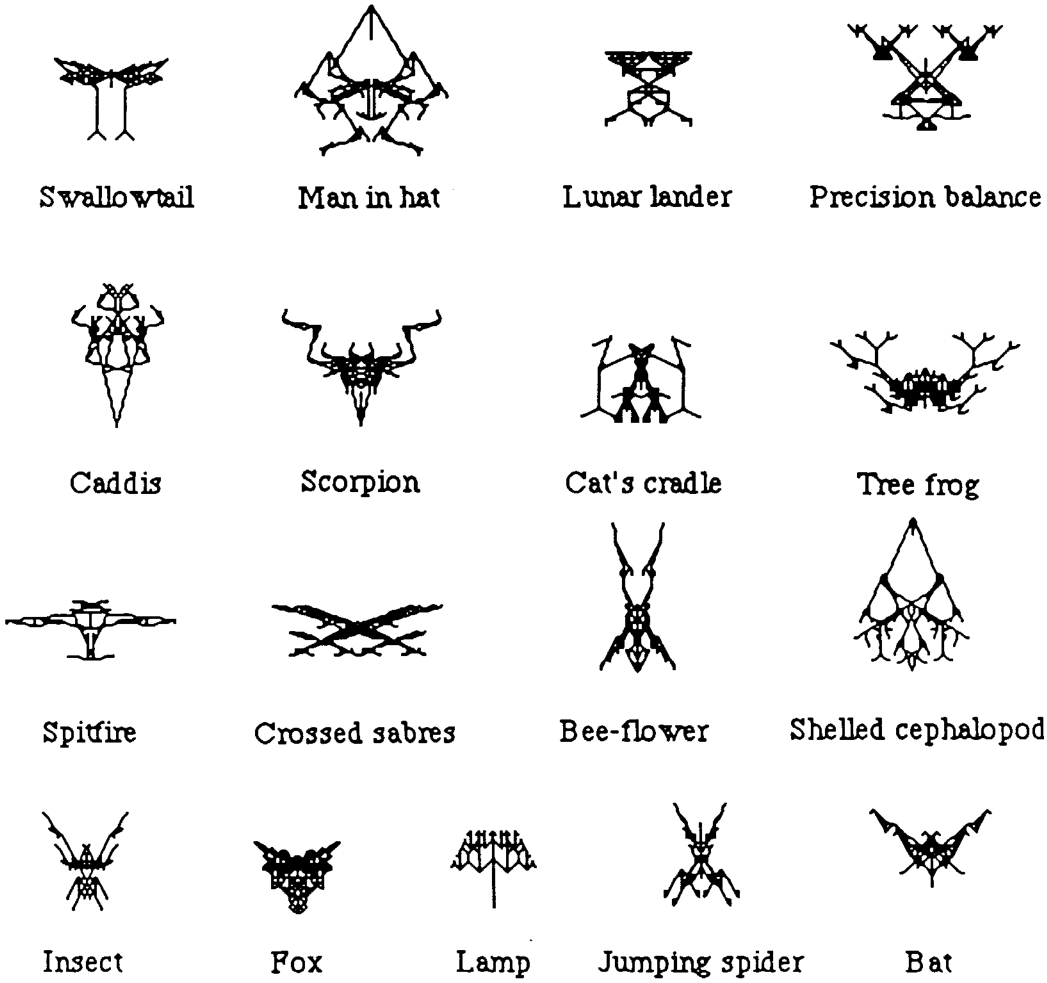

Fig. 2.4 A selection of evolved 'biomorphs'

(reproduced with kind permission of Richard Dawkins)

Finally, the biologist Richard Dawkins has demonstrated the ability of computers to evolve shapes resembling those found in nature (Dawkins, 1986). Using hierarchical tree-structures to define shapes and a simple mutation-based program that modifies the shapes, he has produced images (or 'biomorphs') resembling the shapes of life-forms, e.g. 'spiders', 'beetles', and 'flowers', see fig. 2.4.

All of these systems can be said to *create* images and shapes, since none involve the modification of existing images, i.e. all images are evolved from random beginnings (from scratch). However, all of these systems require the images being evolved to be evaluated by a human (i.e. artificial selection instead of natural selection). Moreover, because of limited representations, most of these systems cannot produce anything more than 'pretty pictures'. Nevertheless, such systems do demonstrate the ability of computers using evolutionary search to generate a wide range of different two and three-dimensional images. In addition, and perhaps more importantly, by evolving a variety of highly original and unusual-looking images, these systems all demonstrate creativity by a computer.

2.6.2 Evolution of Shape and Behaviour

Typical work in the field of Artificial Life consists of investigations into the creation of control systems (or 'brains') for robots, capable of producing behaviour such as action selection, planning and learning (Meyer and Guillot, 1994). Recent work involves the evolution of connections between artificial neurons, and in some cases, the evolution of the form of the artificial creature or 'animat'. For example, Harvey has successfully evolved both the 'brains' and the visual morphology (i.e. position and size of three visual receptive fields) of robots, resulting in robots capable of visually distinguishing between a rectangle and a triangle (Harvey et. al., 1994).

However, perhaps the most notable work in this area is that of Karl Sims, who evolves both the 'brains' and the entire 'bodies' of virtual creatures (Sims 1994a, 1994b). Using rectangular blocks arranged hierarchically to define the simple shapes of the creatures and a genetic algorithm to evolve them, Sims has evolved turtle-like creatures with astonishing abilities to 'swim' and follow lights, as well as 'walking' creatures, and 'jumping' creatures (Sims, 1994a). By co-evolving two competing creatures simultaneously (each trying to 'grab' a virtual block before the other), Sims has shown how the evolution of many designs and counter-designs can develop, see fig. 2.5. For example, one creature might evolve a long arm to snatch the block quickly, only to have the other creature evolve an arm to block this movement (Sims, 1994b).

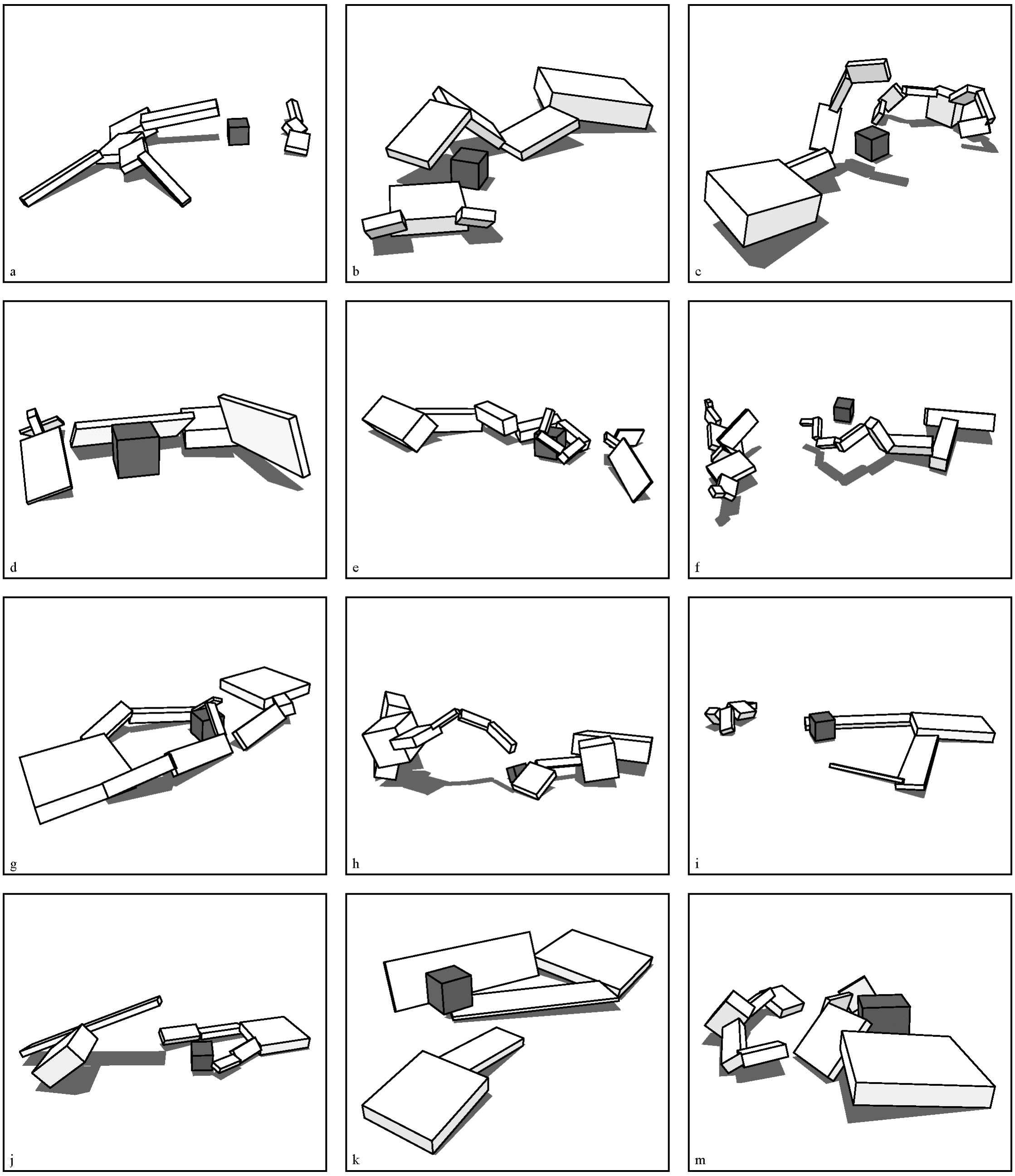

Fig. 2.5 Evolved Competing Creatures

(reproduced with kind permission of Karl Sims)

Whilst the evolutionary design system to be described in this thesis focuses on the shape of designs only, disregarding behaviour, the 'animats' such as those evolved by Sims, clearly demonstrate that computers are capable of evolving functional designs from scratch, evaluated without any human interaction.

2.7 The Creation of Designs by Computers

As described previously, the optimisation of existing designs is relatively common, with the creation of artistic images and artificial life growing rapidly. However, the *creation* of new designs seems to be a less common subject for research, with little literature in existence. Early work concentrated in the cognitive area of creative design automation (Dyer et al. 1986, Maher et. al. 1989, Dym and Levitt 1991), i.e. attempting to make a computer 'think' in the same way as a human, when designing. Such systems attempted to create descriptions of designs at an abstract level, typically using an expert system to 'design'. For example Dyer's 'EDISON' (Dyer et. al., 1986) represented simple mechanical devices such as doors and can-openers symbolically in terms of five components: parts, spatial relationships, connectivity, functionality and processes. A combination of planning and invention using 'generalisation', 'analogy' and 'mutation' attempted to modify these components to fulfil the design specification. Unfortunately, the abstract level at which reasoning was performed was too low, so the system was unable to handle any problems apart from the simplest cases (Pham and Yang, 1993). Another approach consisted of invention based on 'visualising potential interactions' (Williams, 1990). This generated descriptions of designs in terms of high-level components and the interactions between them, using qualitative reasoning and quantitative algebra. Again, the proposed system could only deal with highly simplified designs (Pham and Yang, 1993).

Some researchers do claim to have produced preliminary design systems (Ulrich and Seering 1987, Michielssen 1992, Pham and Yang 1993), but on closer inspection, such systems seem to either consist of the optimisation of connections between existing high-level building blocks or to be simply optimisation of existing designs. For example, Michielssen (1992) describes an approach for designing optimal multilayer optical filters that "does not require a preliminary design". However, the author then describes how a GA is used to determine "the thicknesses of the layers of the filter required for an optimal response". In other words, the GA is presented with a simple preliminary filter design consisting of a fixed number of layers, whose thicknesses are to be optimised. Alternatively, Pham describes a "preliminary design system" known as TRADES (TRAnsmission DESigner) (Pham & Yang, 1993). When given the type of input (e.g. rotary motion) and the desired output (e.g. perpendicular linear motion), the system generates a suitable transmission system to convert the input into the output. However, this GA is presented with a set of high level design building blocks (such as rack and pinion, worm gear, belt drive), reducing the design task to a simple ordering problem. In other words, the GA simply finds the optimal order of existing components within a design; this pre-defined and limited choice could reduce the potential for truly creative design by the system.

In a similar way, Chakrabarti has demonstrated a program capable of creating simple designs of devices and machines which involve motion (Chakrabarti, 1995). Using a knowledge-base and a search-engine, 'FuncSION' can generate a number of different designs by combining different basic elements such as pivots, rods and levers. So far the system has been used to suggest alternative designs for a door-handle and an arm support for sufferers of muscular dystrophy. However, again designs are generated by combining existing components. Moreover, the knowledge-base determines the operation and combinations of the components that are permitted, thus potentially limiting the scope for radically different conceptual designs.

Other researchers favour closer analogies with nature, using artificial embryologies that resemble natural embryology. In other words, genes are used as instructions on how shapes should be 'grown', rather than used to specify the shapes directly. For example, de Garis attempted to grow simple shapes using collections of 'cells' with a genetic algorithm (de Garis et. al., 1992). However, all results produced by the system were disappointing, with even a simple 'L' shape proving difficult. To show how difficult it was to evolve any more complex shapes, the final attempt was to evolve a 'turtle' shape (comprised of six filled circles: one for the body, one for the head, and four for the legs). The authors concluded: "The fitness was only 80% and does not look anything like a turtle. It looks more like a blob." (de Garis et. al., 1992).

Rosenman has had more success using a similar idea to evolve new floorplans for houses (Rosenman, 1996). This is perhaps the work that can most accurately be described as design creation, since actual designs are generated from scratch. Two dimensional plans are 'grown' using a simplified GA to modify 'cells' organised hierarchically using grammar rules (Rosenman, 1996). This spatial representation, although sounding conceptually elegant ('growing' cells to form designs), requires huge, complicated structures to define any design composed of more than just a few cells. Just as de Garis had difficulties, it is possible that this over-complexity will prevent the system from being scalable to allow the creation of more complex and realistic designs.

2.8 Summary

This chapter has examined six distinct areas of research relevant to the evolution of general solid object designs from scratch. First, by reviewing the theories of human design, it becomes clear that although the human design process can be divided into distinct stages, it is not usually explicitly performed in these stages. In reality human design is an evolutionary process, with designs being refined iteratively. Likewise, after reviewing natural evolution, it is clear that evolution can justifiably be thought of as a highly efficient design process, capable of inspiring and teaching human designers.

The genetic algorithm is the closest analogy in computer science to natural evolution. The review of the nature and properties of the GA makes it apparent that the GA does manage to 'borrow' some of the amazing search potential of natural evolution. The GA is one of the most effective and generic of search algorithms known, and has become popular in use for a long list of different applications, particularly design.

Finally, the three areas of research most related to generic evolutionary design were critically appraised. The first of these was the optimisation of existing parameterised designs, showing the ability of the computer, and often of the GA, to optimise many different types of design beyond the ability of a human designer. Another related area was the creation of artistic images by the computer (and again, often by the GA) which demonstrates that computers are able to create a wide range of intricate and detailed images when guided by a human. In addition, the evolution of artificial 'creatures' demonstrates that evolutionary search can also create functional designs from scratch with no human interaction. However, research into the automatic creation of designs is still in its infancy, with very few successful systems in existence. Moreover, as yet no system that creates new designs from scratch (rather than simply ordering pre-defined components) has been demonstrated in more than one problem domain.

From this extensive literature review it was identified that a generic evolutionary design system, capable of creating, from scratch, a range of different solid object designs and optimising those designs without human interaction, does not exist.

Such a system will be described in the remainder of this thesis. This system combines the creative evolutionary techniques pioneered by artists (and biologists) with the more rigorous methods of automatic creative design. This has resulted in a generic design system which has the 'creative properties' of the art systems and is capable of the generation of a wide range of useful designs. Furthermore, it is the 'innovative flair' (Goldberg, 1989) of the genetic algorithm that gives the system such capabilities.