The Table: An Illustration of Evolutionary Design using Genetic Algorithms

PETER J BENTLEY & JONATHAN P WAKEFIELD

DIVISION OF COMPUTER AND INFORMATION ENGINEERING,
SCHOOL OF ENGINEERING, UNIVERSITY OF HUDDERSFIELD, QUEENSGATE,
HUDDERSFIELD, WEST YORKSHIRE HD1 3DH, UK
email: p.bentley@eng.hud.ac.uk

ABSTRACT

This paper describes an attempt to enable computers to generate truly novel conceptual designs of solid objects by using genetic algorithms (GAs). The current capabilities of the system are illustrated by the example of designing a table. Each individual table has its functionality specified by an explicit objective function, which is utilised by GAs to evolve candidate designs. These designs are represented using spatial partitions of 'stretched cubes'. The effect that varying the number of spatial partitions in each design has on evolution is investigated. Additionally, a method of producing symmetrical designs is explored.

1. INTRODUCTION

The genetic algorithm (GA) is a highly flexible and powerful search algorithm that uses principles of evolution observed in nature to direct its search [1,2]. GAs can tackle optimisation problems if these problems are formulated in terms of search, where the search space is the space containing all possible combinations of parameter values, and the solution is the point in that space where the parameters take optimal values. Indeed, GAs are commonly used to optimise designs, with some remarkable results [3]. However, in this paper we aim to demonstrate that GAs are able to do more than just optimise existing designs - we propose that they can create entirely new designs from scratch.

The area of design creation using genetic algorithms is a relatively unexplored area. Using GAs to create new designs has the potentially great benefit of new conceptual designs being automatically created in addition to optimal designs. So far research has been performed in limited ways, such as the use of GAs to create new conceptual designs from high-level building blocks [4] although this often seems to be little more than the optimisation of connections between existing designs rather than true design by the GA. Also the related area of Genetic art is becoming more popular, with various voting systems now being on-line on the internet (e.g. John Mount's 'Interactive Genetic Art' at http://robocop.modmath.cs.cmu.edu.8001).
Other art-evolution systems using humans as design evaluators have been created [5,6], but as yet very few, if any, systems exist that can evolve a design from scratch with no human input during the evolution process. This paper describes an early prototype of an evolutionary design system, capable of doing just that. To demonstrate this, the system is set the task of designing a table. The table was chosen in order to allow investigation of some fundamental aspects of design and because it is a recognisable everyday object.

2. REPRESENTATION

Evolving designs from scratch rather than optimising existing designs requires a very different approach to the representation of designs. When optimising an existing design, only selected parameters need have their values optimised (e.g. for a jet-turbine blade, such parameters could be the angle and rotation speed of the blade). To allow a GA to create a new design, the GA must be able to modify more than a small selected part of that design - it must be able to modify every part of the design. This means that a design representation is required, which is suitable for manipulation by GAs. Many such representations exist, and some are in use by the evolutionary-art systems: a variant of constructive solid geometry (CSG) is used by Todd & Latham [6], others use fractal equations (e.g. John Mount - see the WWW address given above), and Dawkins [5] uses tree-like structures. For a system capable of designing a wide variety of different solid object designs, however, a more generic representation is needed.

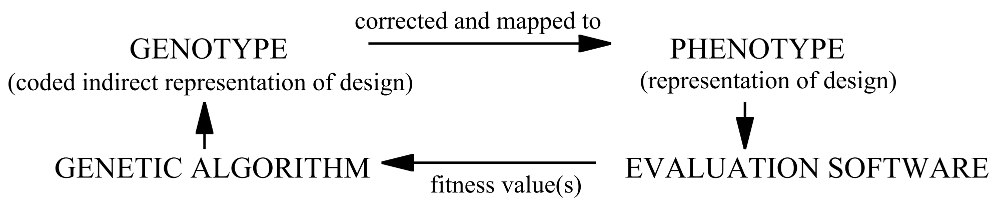

Fig. 1 Operation of the prototype evolutionary design system.

A variant of spatial-partitioning representation has been developed for this work [7]. This representation uses spatial partitions of variable sizes, each one being a 'cuboid' with variable width, height, depth and position in space. Each primitive shape can also be intersected by a plane of variable orientation, but this refinement to the representation will not be considered in this paper. One of the many benefits of using such a variable-sized primitive shape as a spatial-partition is that few partitions are required to represent designs. Significantly, the fewer the partitions in a design, the fewer the number of parameters that need to be considered by the GA. A disadvantage, however, is that it is possible to represent illegal designs using the representation, i.e. it is possible for two spatial-partitions to overlap each other causing redundancy and ambiguity.

The simplest way to overcome the problem of illegal designs is to remove them from the population if any are created. Unfortunately, it seems that a high proportion of all new offspring are illegal. Thus, a large amount of the time spent by the GA is wasted producing illegal designs which are immediately discarded. An alternative method is to correct the genotypes of any illegal designs in the population, making them legal. This is performed by a correction routine, which compares each primitive in a design with every other primitive, and squashes any overlapping primitives until they touch. However, once again, this causes problems. It seems that a high proportion of all new offspring differ only from their parents in that they have one or more primitives that overlap. By correcting them, these offspring become identical to their parents, thus undoing the evolution.

Thus, some form of 'safe' correction is needed, to convert overlapping primitives into non-overlapping primitives without affecting evolution. The solution is to correct the designs during the mapping of the genotypes to the phenotypes, rather than directly correct the genotypes of the designs and interfere with the evolution process by performing 'genetic engineering'. This means that the genotype of a design no longer directly corresponds to the phenotype (i.e. the design) - the shape of the design is now defined by the rules of the representation as well as its genes [8].

Hence, the GA evolves new designs by manipulating coded indirect representations in the genotypes, which are mapped to the direct representation of the phenotypes, Fig. 1. Since the phenotypes are evaluated, not the genotypes, the GA manipulates the shape of the designs indirectly. Despite this fact, the GA is able to take into account the restriction of the design representation, and compensate.

Some form of guidance is necessary to direct the evolution of the designs. This is provided by evaluation software - effectively a software version of the design specification. As will be shown below, the evolutionary design system can evolve new designs from scratch, guided only by such evaluation software.

3. EVALUATION OF A TABLE

A series of previously documented experiments [8] have shown that a table can be adequately specified by the combined use of six evaluation criteria. These are:

SIZE
Perhaps the most basic requirement for a table is that of appropriate size. The size of the design is specified by minimum and maximum extents for the left, right, back, front, top and bottom of the design. The fitness of a candidate design decreases the further it is from the ideal size.

MASS
Another basic, but vital requirement is that of mass. An ideal mass is defined, the fitness of the design decreasing the more the actual mass differs from the ideal mass.

UNFRAGMENTED DESIGNS
The easiest way for the GA to create designs of lower mass is to reduce the dimensions of the primitive shapes that make up the designs. However, this can produce fragmented designs, where primitives become

disconnected from the main part of the design. (Fragmented designs are detected by creating a network of the primitives and their connections in a design, and traversing it recursively. Any primitive that is not part of the main design will not be visited, meaning that the design is fragmented.)

Since this work concentrates on the evolution of a single object at a time, such fragmented designs are heavily penalised. The fitness of fragmented designs is only based on the sum of the distance of each primitive from the origin (normally the centre of the design).

STABILITY
A more complex requirement is stability under gravity - the table must stand upright, and not fall over under its own weight. The stability of a candidate design is determined by calculating whether the centre of mass of the design falls within the outer extents of the primitive(s) touching the ground. The more unstable the design is, the worse its fitness becomes.

FLAT SURFACE
Perhaps the most important requirement of a table is its flat, upper surface - the table top. A good table should have a flat surface at an appropriate height from the ground, and covering an appropriate area. These criteria are evaluated by randomly sampling, the top of the design within the required area of the table top, and checking how much the height at these points differs from the required height. The greater the difference, the less fit the design is.

SUPPORTIVENESS AND STABILITY
One final requirement for a table is that it must be able to support objects on its flat upper surface. Although tables evolved with the first five criteria do stand up on their own, if an object was to be placed on them, most would still topple over. Greater stability is therefore required. This is achieved by a simple extension to the 'stability under gravity' constraint, which requires the table to be able to support an object placed at the very edges of its extremities without falling over.

4. EVOLUTION OF A TABLE

The genetic algorithm used for the experiments described in this paper remains unchanged throughout, the only variation being the number of generations it runs for. A basic canonical GA was used [9], with primitive 'roulette wheel' partner selection [2], and real coding [10]. The population was initialised with primitives of random size and location at the beginning of every test run (i.e. starting from scratch). Initially, two experiments were performed, to discover whether the GA could evolve designs for tables comprised of five primitives.

EXPERIMENT 1
Designs of five primitives were evolved, using the first five evaluation criteria mentioned previously (i.e. the tables were required to remain standing under their own weight, but not to support any additional weight). The population size was set at 100, and the GA was allowed to run for 300 generations.

The resulting designs were usually very good, but compromises were being made by the GA. It was not always able to fully satisfy all of the evaluation criteria since, in order to have a large flat surface, the size of the primitives must be increased, but to have a low mass, the size of the primitives must be decreased. Results were often designs with excellent table tops, but too massive (Fig. 2), or designs with unusual multi-level table tops with the right mass (Fig. 3). Nevertheless, all the designs were the right size, stable, and unfragmented.

EXPERIMENT 2
Again, designs of five primitives were evolved, this time with all six evaluation criteria being used. The GA was allowed to run for 500 generations, and produced some remarkably fit designs, coming up with two main methods for increased stability: a very low centre of mass (Fig. 4) or a very wide base (Fig. 5).

EXPERIMENT 3
Having found that excellent designs can be evolved for tables comprised of five primitives, an experiment to evolve tables using fewer primitives was performed. For this experiment, designs of only three primitives were evolved, again using all the evaluation criteria. The GA ran for 600 generations, but the resulting designs were not as expected: most of the time, insufficiently stable tables evolved, Fig. 6. It seems that, because of the random initial positions of the primitives, designs with three primitives on top of each other are rare, meaning that the expected design (Fig. 7) could not evolve most of the time. Nevertheless, the GA produced the best it could, given the poor initial population. As is clear from Fig. 6, the table is the right size, mass, is unfragmented, has a good table top and is very stable in one axis at least.

EXPERIMENT 4
To further investigate the effect that varying the number of primitives has on evolution, designs of ten primitives were evolved. After 600 generations, the evolved designs were typically as shown in Fig. 8. Clearly, ten primitives are too many to evolve satisfactory

	P1	P2	P3
P4	1*	2	4
P5	2	3*	5
P6	4	5	6*

Table 1 Order to check and correct overlapping primitives (after reflection in x = 0)

	P1	P2	P3	P4	P5	P6
P7	1*	2	4	1	2	4
P8	2	3*	5	2	3	5
P9	4	5	6*	4	5	6
P10	1	2	4	1*	2	4
P11	2	3	5	2	3*	5
P12	4	5	6	4	5	6*

Table 2 Order to check and correct overlapping primitives (after reflection in x = 0, then z = 0)

designs; the GA is unable to incorporate all of them in the design, with the result that the table is not well formed and has unnecessary parts. When the redundant primitives were manually removed, the design was much improved (see Fig. 9). Despite still being too massive, this table does have a good table top, and a very stable 'T' shaped footprint.

5. SYMMETRICAL DESIGNS

Although it should be apparent that the GA can indeed create novel, usable designs for tables using five primitives, almost all of the evolved tables have just a single support, unlike the four-legged table most of us are used to. The missing feature is symmetry - all of the tables evolved are highly asymmetrical. Enforcing symmetry would allow the GA to produce tables with two or four legs.

Symmetrical designs could be requested by the addition of another criterion to the evaluation software, i.e. the less symmetrical a design is, the less fit it is. However, previous experience indicates that the GA would rarely evolve designs that fully meet the strict requirements of symmetry. Additionally, the calculations to determine the degree of symmetry in a design would be complex. A more attractive method is to enforce symmetry by reflecting the design in one or more planes. This reflection is performed in the genotype to phenotype mapping, meaning that the genotype need only specify the non-reflected portion of the design. This has the advantage of always producing symmetrical designs, with the GA only needing to manipulate half or a quarter (depending on the number of reflections) of the primitives in each design.

Reflection can be performed in the x = 0, y = 0 and z = 0 planes. A design can intersect a plane and still be reflected in it: all primitives on one side are reflected to the other. Since this can (and often does) produce designs that have primitives overlapping, the reflected design must be corrected once again.

For example, consider a design consisting of primitive shapes P1 to P3. The genotype of the design will always only hold coded versions of these three, no matter how many reflections take place later. During the mapping of genotypes to phenotypes, the three primitives are checked against each other and corrected should any overlap. The design is then reflected in the plane x = 0, producing a symmetrical design consisting of primitives P1 to P6. The two halves: P1 to P3 and P4 to P6 are then checked for overlaps and corrected if necessary. This checking and correction process must be performed in a symmetrical manner, to ensure both halves of the design are corrected identically. (To correct overlapping primitives, the primitives are squashed in the direction which ensures the least change occurs to them. However, there is a special case when correcting a primitive that overlaps its mirror image - it must always be squashed in the direction normal to the plane of reflection.) Table 1 shows the order in which the checks must be performed, with each matching number representing a check to be performed simultaneously and each special case denoted by an asterisk.

For a design symmetrical in two planes, the process is repeated. The design consisting of primitives P1 to P6 is reflected in z = 0. The new design P1 to P12 must then be corrected again, with these checks and corrections (if necessary) occurring in a new symmetrical manner as shown in Table 2.

Despite the seemingly large number of checks that need to be performed for each design during the genotype to phenotype mapping, by checking at each stage, the number has been reduced. In the example above, if every primitive shape was checked for overlaps *after* both reflections, 11+10+9+...+2+1 = 66 checks would be required. By checking at each stage of reflection, the number of checks is reduced to 3+9+36 = 48.

Using this method of enforcing symmetry, two more experiments were performed.

EXPERIMENT 5
Designs of five primitives reflected about x = 0 (symmetrical ten-primitive designs) were evolved for 600 generations. The results were the best yet seen, being judged practically perfect by the evaluation software (see Figs 10 & 11). As expected, introducing symmetry made evolution to good designs a much easier task for the GA. A symmetrical design is inherently more stable and it is much easier for crossover and mutation to produce designs of the right size, mass and with good table tops.

Although the designs do consist of ten primitives, since many are 'doubled up' (i.e. the table tops in both figures shown are made up from two, not one primitive), there is no serious problem of too many primitives as was seen in the fourth experiment.

EXPERIMENT 6
Designs of five primitives reflected in x = 0 and z = 0 (symmetrical twenty-primitive designs) were evolved for 600 generations. The results were again excellent, but for some the high number of primitives did reduce the fitness slightly. Perhaps the greatest problem with so many primitives in each design was one of aesthetics - the tables tended to look somewhat cluttered. However, since the evaluation software judges tables purely on functionality, not artistic features, some results of this type are to be expected. The GA favoured two main types of design: the table with one large base, Fig. 12, and the four-legged table, Fig. 13.

6. CONCLUSIONS

The genetic algorithm is capable of more than design optimisation - it can evolve entirely new designs. The GA can modify coded, indirectly represented designs and compensate, producing some excellent results. It can evolve symmetrical, almost perfect designs of tables (as judged by the evaluation software), despite the fact it only indirectly manipulates such designs.

It is clear that the number of primitive shapes permitted in a design strongly effects the ability of the GA to evolve good results. To solve this problem, it is anticipated that the GA can be made to evolve not only the position and dimensions of the primitive shapes, but also the number of primitives making up a design. Thus the number of primitives in a design could be increased or reduced during evolution, perhaps by a new mutation operator, until optimal.

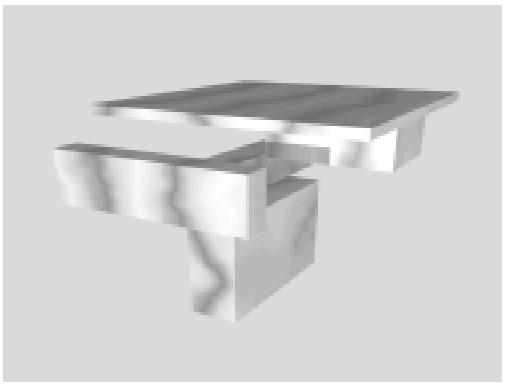
Fig. 2 Experiment 1: evaluating size, low mass, stability and flat top

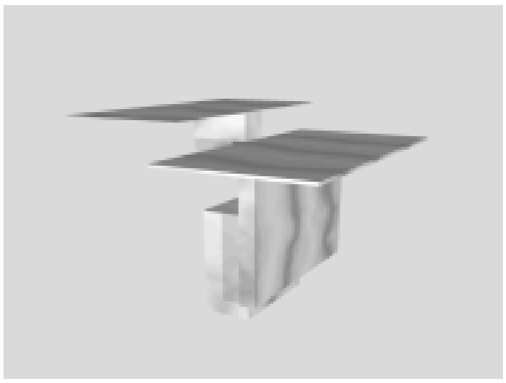
Fig. 3 Experiment 1: evaluating size, low mass, stability and flat top

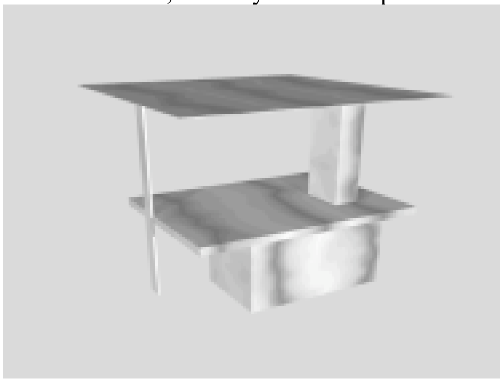
Fig. 4 Experiment 2: evaluating size, low mass, flat top and greater stability

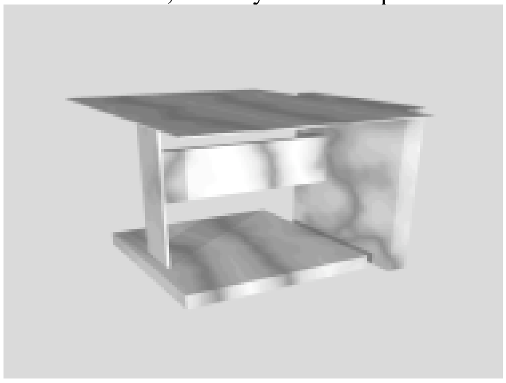
Fig. 5 Experiment 2: evaluating size, low mass, flat top and greater stability

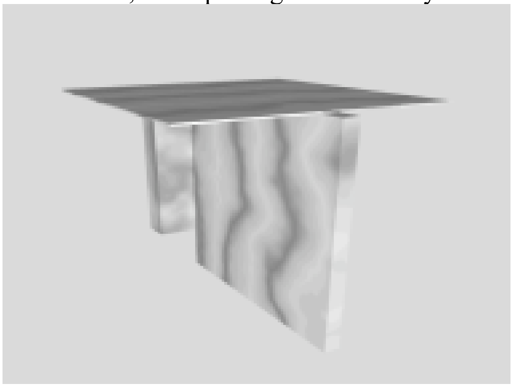
Fig. 6 Experiment 3: design evolved with 3 primitives.

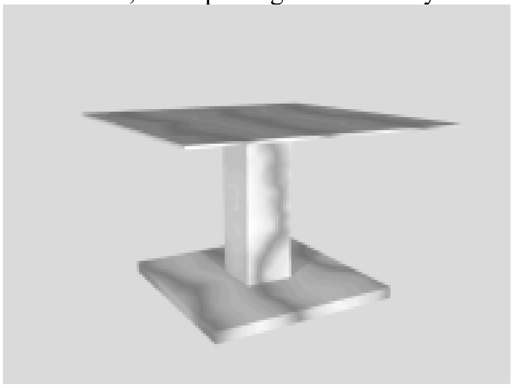
Fig. 7 Experiment 3: expected design with 3 primitives (not evolved).

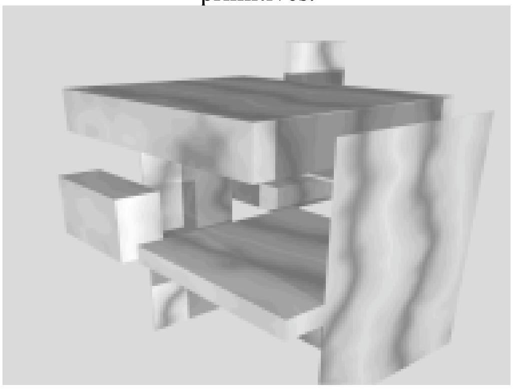
Fig 8 Experiment 4: design evolved with 10 primitives

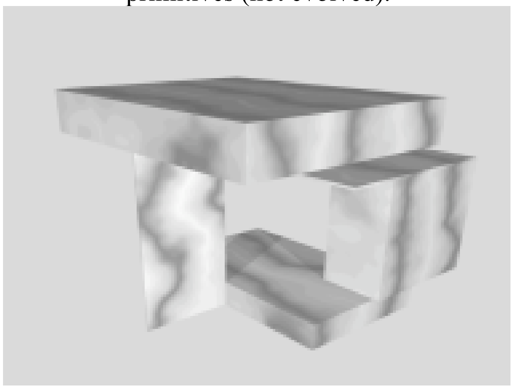
Fig. 9 Experiment 4: design evolved with 10 primitives, redundant primitives removed.

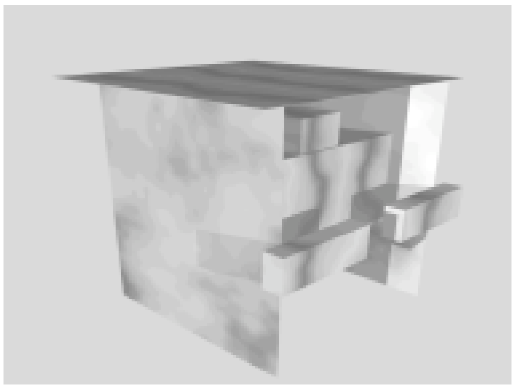

Fig. 10 Experiment 5: evolved design symmetrical in x = 0

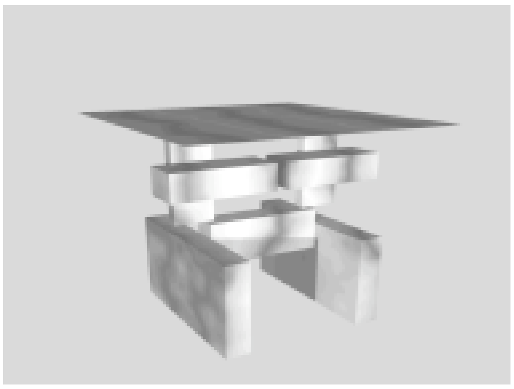

Fig. 11 Experiment 5: evolved design symmetrical in x = 0

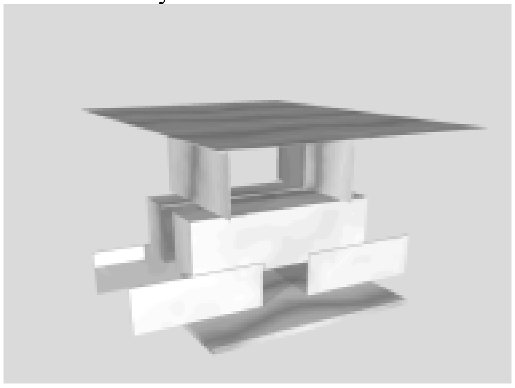

Fig. 12 Experiment 6: evolved design symmetrical in x = 0 and z = 0

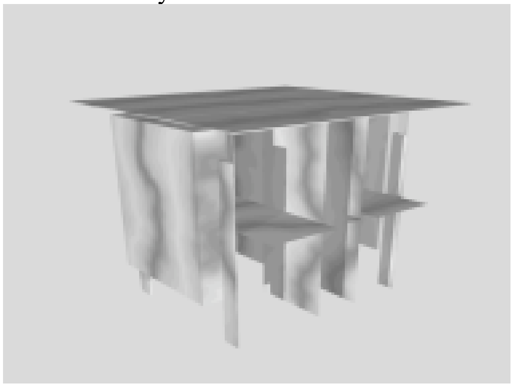

Fig. 13 Experiment 6: evolved design symmetrical in x = 0 and z = 0

7. REFERENCES

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

[2] Goldberg, D E (1989). *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley, USA.

[3] Parmee, I C & Denham, M J (1994). The Integration of Adaptive Search Techniques with Current Engineering Design Practice. Proc from *Adaptive Computing in Engineering Design and Control -'94*, Plymouth, 1-13.

[4] Pham, D T & Yang, Y (1993). A Genetic Algorithm based Preliminary Design System. *Journal of Automobile Engineers v207:D2*, 127-133.

[5] Dawkins, R (1986). *The Blind Watchmaker.* Longman Scientific & Technical Pub.

[6] Todd, S & Latham, W (1992). *Evolutionary Art and Computers.* Academic Press.

[7] Bentley, P J & Wakefield, J P (1994). Generic Representation of Solid Object Geometries for Genetic Search. *Microcomputers in Civil Engineering* (to appear).

[8] Bentley, P J & Wakefield, J P (1995). The Evolution of Solid Object Designs using Genetic Algorithms. Proc. *Applied Decision Technologies*, April 1995, London.

[9] Mühlenbein, H (1992). Darwin's continent cycle theory and its simulation by the Prisoner's Dilemma, Conf. Proc. *1st European Conference on Artificial Life Towards a Practice of Autonomous Systems.* Paris, 236-244.

[10] Goldberg, D E (1991). Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking. *Complex Systems 5*, 139-167.