Proceedings of ASCE 2005 International Conference on Computing in Civil Engineering, Cancun, Mexico; July, 2005

OVERCOMING REPRESENTATION ISSUES WHEN INCLUDING AESTHETIC CRITERIA IN EVOLUTIONARY DESIGN

Azahar T. Machwe[1], Ian C. Parmee[2] and John C. Miles[3]

ABSTRACT

It is well known that for any sort of evolutionary search we must represent the problem solution in a suitable manner since the choice of representation has a large impact on the type and efficiency of the evolutionary search procedure applied. Usually in evolutionary design applications either bit strings or real number parameters are used to encode the problem. However, during the initial design phase a 'design' may not be decomposable into real number parameters since the nature of the search space is not well understood and / or the designer wishes to maintain a highly flexible approach whilst establishing an initial configuration. The paper introduces the overall objectives of the project and discusses representation issues before presenting an object-based representation which tries to incorporate the ambiguity present during the initial design phase by working with design elements and objects as members of the chromosome. Ambiguity is particularly acute in this case as the overall project objective is the development of a user-centric evolutionary design system that includes aesthetic criteria evaluation.

KEYWORDS

Agents, Engineering Design, Interactive Evolutionary Design, Representation techniques.

---

[1] ACDDM Lab, CEMS, University of the West of England, Bristol, UK, Phone: +44-7748600178, azahar.machwe@uwe.ac.uk

[2] ACDDM Lab, CEMS, University of the West of England, Bristol, UK. Phone: +44 (0)117 328 3137; Fax: +44 (0)117 328 2587, ian.parmee@uwe.ac.uk

[3] Cardiff School of Engineering, Cardiff University, Cardiff UK, CF24 0YF, Phone: +442920875694, MilesJC@cf.ac.uk.

INTRODUCTION

The research described in the paper relates to user-centric intelligent design systems and creativity in design. Creativity is initially considered through the inclusion of aesthetics as additional design criteria within a planned, semi-autonomous machine-based design environment. The proposed system brings together agent-based machine learning, evolutionary computing and subjective evaluation in design space search and exploration for aesthetically pleasing, structurally feasible and usable designs. The aim is that the system will be capable of learning basic characteristics relating to aesthetically pleasing designs from user-evaluation within an evolutionary search process. It is intended that, as the search progresses there will be a gradual lessening of the degree of user interaction allied with an increasing degree of autonomous machine-based solution evaluation involving both aesthetic and structural criteria.

Although research relating to artificial design environments is evident in the literature (Gero (2002), Rosenman (1997b) ,Bentley (1999), Parmee (2002)) there is little evidence of the integration of user evaluation, evolutionary search and exploration and agent-based machine learning With respect to the addition of aesthetics into computer-based design, much theoretical work (in the form of the development of computer models) is evident in this field but little application-based research has been done (Moore et. al. 1996; Saunders 2001).

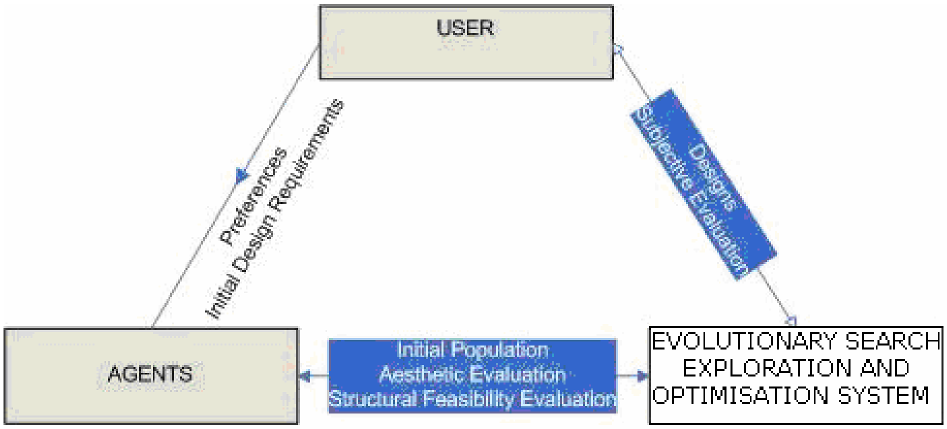

Figure 1: The Proposed System

The three main components of the proposed system are shown in figure 1, i.e. the user, the agents and the evolutionary search, exploration and optimisation system (ESEO). The primary purpose of the user is to define initial design requirements and to aesthetically evaluate the designs generated by the ESEO during the initial generations. The agents have multiple tasks which include the creation of the initial population based on design requirements and the monitoring of designs for feasibility during the ESEO processes. The ESEO identifies design solutions that can be considered high performance in terms of multiple criteria such as structural stability and cost.

The project is initially considering three test domains: bridges; liquid containers such as vases, wine glasses; chemical tanks, etc and street furniture in the form, initially, of bench type structures. These three domains have been chosen because of their differing design criteria and the need for differing forms of representation. The first domain is highly constrained, the second potentially requires some complex, non-linear shapes and the third looks similar to bridges but is actually far less constrained and offers interesting challenges re flexibility of reasoning

This joint research project involves the ACDDM Lab at UWE, Bristol and the Institute of Machines and Structures at CU, Cardiff. The paper concentrates upon the primary stages of the overall research project and upon the first test domain i.e. bridges. These early stages have largely involved the development of highly flexible and robust representations of simple bridge structures and the subsequent identification of high-performance solutions via basic evolutionary algorithms. Evaluation of solutions has been solely in terms of spatial feasibility, simple structural analysis and cost / material weight considerations.

THEORETICAL OVERVIEW

'A recurring issue in all design research is the issue of representation' (Rosenman (1997a)). This statement best describes the problem at hand. The issue of representation is central to any evolutionary system since the efficiency of the search as well as effectiveness of the evolutionary operators such as mutation and crossover is directly linked with the representation (Goldberg (1989), (Rosenman (1997a)).

Traditionally, in the case of genetic algorithms (GA), evolutionary strategies (ES) and evolutionary programming (EP) representations are based on binary or real number variable parameter strings. Such representations have their own advantages and disadvantages but many alternatives are also available (Goldberg (1989), Holland (1975) and Rosenman (1997a)). Bentley (2000) states that component based representations 'allow increased freedom for evolution' Cramer (1985), Rosenman (1996) and others have proposed hierarchical representations since design objects are complex entities with many related sub-systems / components that cannot be efficiently represented by a simple variable string. Peysakhov et al (2003) use a messy GA to assemble structures from LEGO™ blocks using a representation based on assembly graphs. These representations indicate that for a representation to be flexible as well as robust a component based hierarchical representation is necessary. Coupled with this we also need to examine what kind of stochastic search process would best suit this representation in terms of efficient negotiation of the design space which is equally important as the efficient encoding of the variable parameters.

Assuming we wish to achieve a high degree of solution search and exploration population based approaches (i.e. approaches that search from many trial points initially well-distributed across the design space) would seem most appropriate. Genetic algorithms (GA), evolutionary programming (EP) and evolutionary strategies (ES) appear to offer high utility. The basic difference between the three is the usage of evolutionary operators. GAs use crossover as the main exploratory operator (Deb (2001)). ES is similar to a real parameter GA without crossover although 'recent ES studies have introduced crossover like operators' (Deb (2001)). Finally EP (Fogel (1988)) is a purely mutation based evolutionary algorithm where mutation is the only exploratory operator.

EP could be considered to represent the simplest of the above algorithms since it has just one operator namely mutation. Thus the representation is not restricted by the need to support crossover between differing variable strings. GAs are at the other end of the spectrum where the representation has to be robust enough to handle repeated crossovers while ensuring the validity of the children. ES in its many flavours and types such as (μ, λ) and $(\mu + \lambda)$ lies somewhere between them. It is clear that it will be important to assess the utility of the above algorithms in terms of the representations under development. Since EP and GA represent two ends of the spectrum it would be prudent to test the representation on these two before moving on to ES and hybrid methods.

REPRESENTATION ISSUES

In any evolutionary optimisation process we find that the representation chosen plays a very important role in determining search efficiency and the quality of the solutions obtained. Mainly there are two classes of representations, string based and tree based. While tree based representations are used with Genetic Programming systems, variable strings are used with GA/EP/ES systems.

Our initial goal was to create a representation which not only was flexible in terms of the possible designs that it could represent but also robust enough to be used for design search, exploration and optimisation. To ensure a high degree of flexibility pure string based real number representation needs to be avoided. Thus we decided to use a collection based object oriented representation. Here a single population member (chromosome) is a collection of primitive elements that represent a design. For example any structure made up of LEGO™ bricks can be represented as a collection of primitive design objects each with a specific **x** and **y** position and a pre-defined length (along X) and height (along Y). Here the LEGO™ bricks are the primitive design objects which when used again and again at different positions and orientations give us a complete structure. We also have the flexibility of using different elements with different design properties by just including them in the set of possible design primitives. When it comes to the evaluation of fitness of the structure and checking the structural integrity we use secondary properties of the particular primitive element type. An argument was initially made for the use of a design grammar based GP system but initial investigations in this direction indicated that such a system would take away the flexibility by trying to force the mapping of the design onto a tree structure and by the complexity relating to maintaining feasibility during evolutionary operations.

Figure 2 further clarifies the idea of using an object based representation. A simple bridge design is basically divided into two separate collections. These are the **Span Element**

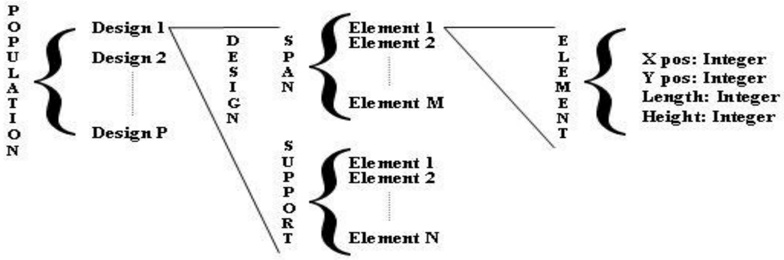

Figure 2: Details of the object based representation.

collection containing elements which form the span of the bridge and the **Support Element** collection containing elements which form the support of the bridge. In the case of a free span bridge the **Support Element** collection will be empty.

Each Element is basically a rectangle with properties as previously described. An Element can either be part of a supporting element collection or a span element collection. Since initially we are looking at a simple beam span bridge with and without supports and a bridge with angled beam span sections there are only two basic types of Elements required. These are the angled section Element (to be used as a span element only) and a simple rectangle Element which can be used as both spanning and supporting element. To extend the design into the third dimension all elements have a constant width. Thus only the profile of the bridge is relevant. An addition benefit of using an object based representation is that it can take advantage of the principles of object oriented programming such as inheritance. Thus if we wanted to add a new kind of element, say curved span section, we could easily do so by extending the basic properties of Element and adding the extra properties required for a curved section. To further elaborate on the representation we describe the manner in which a mutation operator acts on it.

MUTATION

Let us assume the chromosome to be mutated represents the following design:

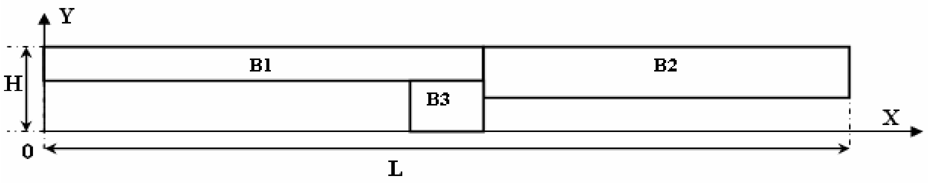

Figure 3: Design before mutation.

The above is a simple beam bridge with a single support (B3) and two span elements (B1, B2). L is the span and H is the maximum height of the bridge.

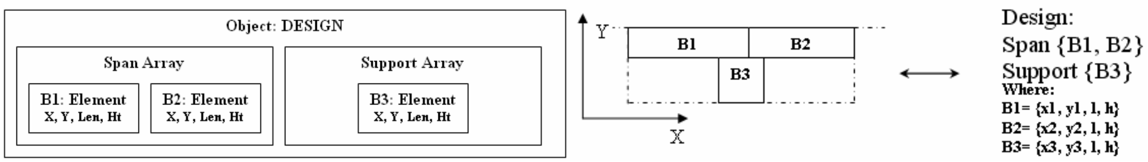

Figure 4: Structure of a design chromosome.

The mutation is rule based and a rule is selected randomly. There are separate rules for the two arrays of elements. The supports can only move left or right. Their height is based upon the thickness of the spanning element they support . Hence there are only 4 rules for supports i.e. two rules for left and right movement and two for increasing and decreasing width. The depth of each span element can vary but they must have a level upper surface and must be continuous with no overlap or space between them. Thus for a simple Element in a span there are just two rules namely to increase or decrease the span depth. Now for example if the selected rule for support (B3) is to move it left by a constant distance (say 2 units) and for span to decrease thickness of B2 support by constant units (say 2 units again) then the B3 object in the support array will have its X value attribute decreased by 2 units and B2 object in span array its height value attribute decreased by 2 units. The height attribute of the support will be automatically adjusted to make it continuous and remove any overlap at its new position.

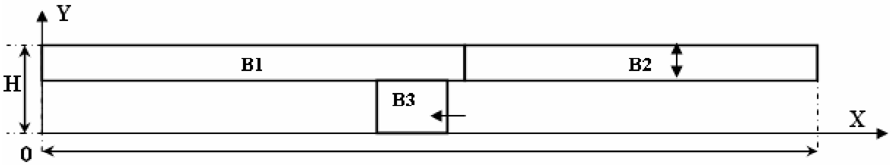

Figure 5: Design after mutation.

ADVANTAGES OVER BIT / REAL NUMBER REPRESENTATION

The major argument against using real number/bit-string chromosomes is that at the start of the design activity it may not be desirable or possible to strictly parameterize a design. As the design process continues designs make the transition from abstract concepts to well defined specifications. Object based representation has the advantage that it can represent designs at all the levels. At the abstract level by using high level objects like Elements and at the design maturity level as a set of specifications. Their functionality can be added to, based on the changing requirements. Thus objects offer a relatively straightforward way to cover design activity at all levels. To produce the same in a simple string based chromosome we would require additional checks to ensure consistency of the chromosome is not violated once a bit or real number is added or removed or its functionality modified. The overall system would be overly complex and difficult to manipulate.

INTRODUCTION OF AGENCY

Initial testing of the representation involved freeform assembly of simple structures such as a linear span and stepped arch using GA, EP and agents (for comparison). It was found that agents are able to assemble free form structures quite easily and with respect to agent based assembly, evolutionary methods are slow. Thus we decided to merge the evolutionary and agency approach. Since agents were good at building structures and evolutionary methods have been known to be efficient in terms of search and optimisation it was decided that the agents would create the initial population of bridge structures and an evolutionary system would perform search, exploration and optimisation within the space of possible structures. Also during these potentially disruptive processes any changes made in the structure would be monitored by the construction agent to ensure that the resulting structure is correct.

The agents at present have a simple task of assembling various structures with various sizes and shapes of span and supports. There are no restrictions on the placement of supports and other design characteristics. But as part of the future work agents will be given specifications of the designs to be built including restrictions on placement of supports and types of span sections used. Thus the agents will be told what the design environment is and they will create initial population designs within it. Then the evolutionary process will take care of the SEO process with the construction agent keeping a check on design changes.

INTRODUCING SIMPLE STRUCTURAL CRITERIA

Having demonstrated how the object based representation can be used to assemble simple structures it is now necessary to test the feasibility of the approach in terms of subsequent SEO processes. The construction agents can currently create three kinds of bridges. These are: Simple beam bridges without support (Type 1a), Simple beam bridges with supports (Type 1b) and Simple beam bridges with sloping span sections and supports (Type II).

Thus an initial population can consist of a mixture of three designs. In future work these designs will be assessed in terms of the designer's aesthetic preferences in addition to structural and cost criteria. However, during this initial establishment of a flexible, feasible representation solution fitness is assessed by applying simple length depth ratios whilst also minimising material used.. Column design is assessed via simple buckling criteria. It is not considered necessary to include loading other than beam weight during this preliminary work which is merely evaluating the developed representation.

FITNESS EVALUATION FOR TYPE 1A

Type 1a consists of a simple concrete span bridge without supports. This is treated as a simple beam deflection problem under UDL. For analysis purposes we use a simple heuristic that the ideal length to height ratio for a span element is 20:1. The closer a span section is to this deal ratio better is its fitness.

$$F_i = | R - (\frac{l_i}{h_i}) | \qquad Stability = \frac{1}{(1+\sum F_i)}$$

where

l_i,h_i = length and height of ith span

R = ideal l to h ratio

F_i= fitness of ith span element

To ensure the overall integrity of the structure the above equations are used. As we can see the closer the dimensions of the span elements are to the ideal ratio (*R*) the lower will be the value of F_i. At the minimum all F_i's are equal to zero and thus stability is equal to one. Taking into account material usage (M) the net fitness function then becomes:

$$\boxed{Fitness = Stability + \frac{1}{Material\_Usage}}$$

FITNESS EVALUATION FOR TYPE 1B AND TYPE 2

Type 1b is a simple span bridge with supports. Here we take into account the buckling in the columns due to the weight of the loaded beam. The formula for buckling load is:

$$\boxed{P' = \frac{\pi^2 EI}{H^2}}$$

where P' = maximum allowable

I = moment of inertia

H = Height of column

E = modulus of elasticity

Thus if the load on a column is greater than P' it will buckle. The load on column is simply determined by first finding the length of the beam between the supports on left and right of the main column and then calculating the load on that length of the beam and the weight of the section (using density of concrete). This is then divided by two to give the loading for the central column. If a column has thickness sufficient to prevent buckling then it can both increase and decrease in thickness. Otherwise when selecting mutation rules for a support which is in danger of buckling, the only option available is to increase the thickness. Here the fitness of the structure is calculated as above. In the case of the sloping element the only difference is that the length taken is the sloping length and the height taken is the 'thickness' attribute of the sloping element (instead of the actual height).

TEST RESULTS

The test problem was to span a 100m gap. A simple EP system was used with a population size of 100 designs. Tournament selection was used as the selection operator with a tournament size of 10. The system was run for 100 generations. A few members from the initial population are shown below.

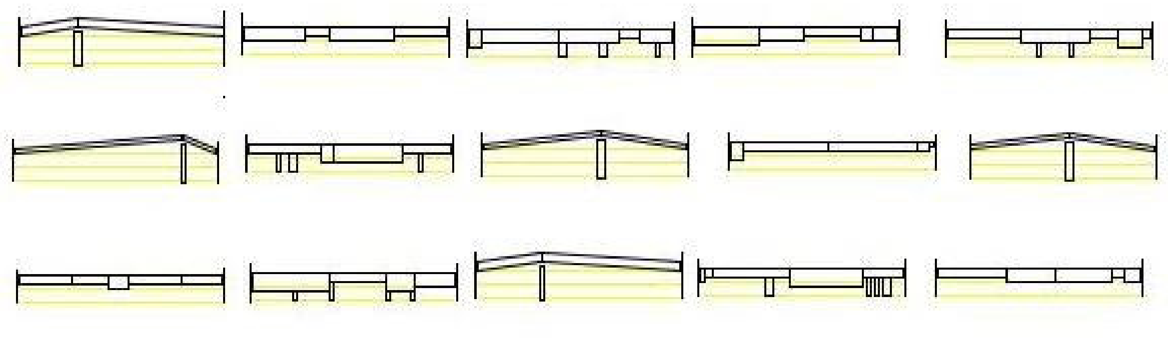

Figure 6: Sample of mixed initial population of bridge shapes.

We can see that the initial population consists of three different kinds of designs namely a simple unsupported span, a simple supported span and a slanted span bridge. After 100 generations the optimal designs shown in figure 3 are achieved.

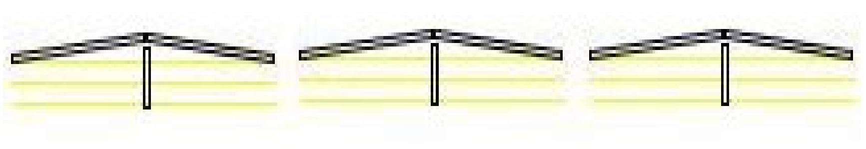

Figure 7: Run 1 optimized bridges.

Here we can see that slanted span bridges turned out to be most efficient in terms of stability and material usage. This means that the other two design types have been evolved out of the population. A second run again produces optimal slanted span bridges but in a different configuration suggesting that there may be several optimal configurations in this category.

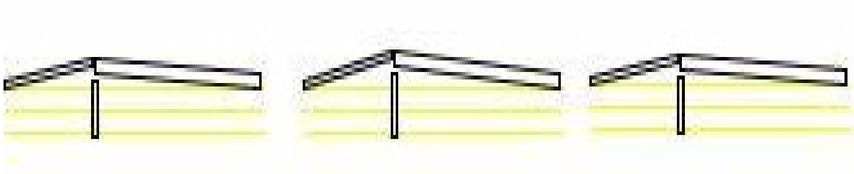

Figure 8: Run 2 optimized bridges.

FURTHER WORK

The results shown above plus additional thorough testing of the developed representation confirm a significant potential although it is intended to further explore alternative representations that may offer increased utility in terms of flexibility and robustness. In addition, further possible bridge types will be considered for inclusion in the initial population. It will be necessary to develop the structural evaluation procedures to some degree whilst avoiding conflicts relating to usability and computational expense. Ultimately the system will be required to give a comparative indication in terms of aesthetically pleasing design and likely cost whilst indicating structural feasibility. Best alternatives generated by the system could then be subjected to further, more rigorous analysis off-line.

Concurrently, we intend to utilize the construction and repair agent / EP representation (subsequently referred to as CARA-EP) to commence the development of the user evaluation process and the multi-agent learning environment. In this next stage of the research the designer will be actively involved in the evolutionary process by ranking the CARA-EP generated designs in terms of personal aesthetic criteria. This ranking will then result in the fitness of the more aesthetically pleasing solutions of each population being appropriately increased. Such user-involvement could ensure the survival of unsupported and supported bridge types of the test population of figure 6 from disappearing as was the case in runs one and two by increasing their competitiveness. Alternatively, if the slanted sections were still considered the most aesthetically pleasing then user-involvement could bias the evolutionary process to one or the other of the high performance solutions of figures 7 and 8.

The introduction of such an interactive process poses many questions such as:

- How many designs from each population should be presented to the user?
- How should these be selected?
- How many evaluations can a user be expected to perform before becoming bored and / or fatigued?
- Should our evolutionary algorithm be able to cope with small populations to reduce new solution numbers or would a steady-state approach be more appropriate?

etc

These questions are not new and have been repeatedly posed but seldom successfully addressed within the interactive evolutionary computing (IEC) community. The reader is directed to http://www.ad-comtech.co/Workshops.htm where output from three recent IEC

Workshops held at the Genetic and Evolutionary Computing Conferences (2001,2002,2003) can be found along with extensive references to IEC applications. Takagi (2001) provides an excellent overview of the area in his review paper. However, more recent developments are also of interest.

The CARA-EP representation will allow us to commence further exploration of these and other issues. However, the intention of this project is to significantly decrease the evaluation load on the designer as early as possible in the evolutionary process by introducing a multi-agent based learning environment that supplements and eventually takes over the aesthetic criteria evaluation.

The following procedure is envisaged. A 'negotiating agent' approach (Cvetkovic &, Parmee, 2003) will be adopted to identify those aspects of given sample designs which do or do not have aesthetic merit. The software agents will each represent a particular established aesthetic such as those proposed by Moore et al (1996b), Ngo et al (2003) and Stamps, (1999) and these will be ranked by the user in terms of preference (Cvetkovic & Parmee, 2001). The agents will negotiate to determine the relative performance of the solutions of each population and return the 'best' solutions to the user in rank order. If the user disagrees with the ranking then changes to aesthetic preferences can be made. The system will then analyse the user preferences and adjust the agents' weightings to match. Thus rather than assessing each solution the designer can scan the characteristics of the best designs, assess whether the aesthetic preferences are operating appropriately and adjust as necessary. This procedure may initially occur at every generation but at a lesser frequency as the agents converge upon the designer's aesthetic preferences. The eventual outcome should be the identification of solutions that satisfy structural and cost criteria whilst also being considered by the designer to be aesthetically pleasing.

The basic thrust of the work is to establish whether or not this coupling of multi-agent activity to evolutionary search can be developed into a learning capability where agent memory contributes to preference selection. This would lead to semi-autonomous activity that, whilst significantly reducing the load upon the designer, ultimately results in viable and aesthetically pleasing solutions. The introduction of more than one designer each independently assessing aesthetic value is a natural progression which will add further layers of complexity that require investigation.

CONCLUSION

A comparative investigation of a number of evolutionary algorithms and associated problem representations has been carried out. An object based representation that allows the flexible generation of simple structures whilst being easily implemented within an evolutionary process (EP) has been developed. The effectiveness of an associated rule-based agency approach to develop initial solution populations and maintain and repair solutions in subsequent generations has been illustrated. The overall implementation has been tested in terms of the construction of three basic bridge structures, their simple analysis and the evolution of high performance solutions.

The tests showed that the CARA-EP system performed as expected. Thus this it is worth further exploring this representation. Specifically there is need to formalise this representation and the surrounding framework to develop a standard EP based system. There is also need to expand the library of shapes used to include more complicated shapes to truly harness the systems potential.

The next stage of the research relating to the introduction of user evaluation of aesthetic criteria has been introduced and related to the preliminary results from the CARA-EP implementation.

REFERENCES

Bentley, P. J. (2000). “Exploring Component-Based Representations - The Secret of Creativity by Evolution?” In Proc. of the *Fourth International Conference on Adaptive Computing in Design and Manufacture* (ACDM 2000), April 26th - 28th, 2000, University of Plymouth, UK. I. C. Parmee (ed), pp. 161-172.

Bentley, P.J. ed. (1999). *Evolutionary Design By Computers.* 1^{st} Edition. Morgan-Kaufmann, USA.

Cramer, N.L. (1985) “A representation for the adaptive generation of simple sequential programs”, Proc. of *an Int. Conf. on Genetic Algorithms and their Application,* 1985, pp.183-187.

Cvetkovic D., Parmee I. C. (2003) “Agent-based Support Within an Interactive Evolutionary Design System”. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing Journal*; Cambridge University Press, 16(5), pp 331-342

Cvetkovic D., Parmee I. C. (2001) “Preferences and their Application in Evolutionary Multi-objective Optimisation”. *IEEE Transactions on Evolutionary Computation*, **6**(1), pp 42-57.

Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms,* John Wiley and Sons.

Fogel, D.B. (1988). “An Evolutionary approach to the travelling salesman problem”. *Biological Cybernetics* 60(2), 139-144.

Gero, JS (2002) “Computational models of creative designing based on situated cognition”, in T Hewett and T Kavanagh (eds), Creativity and Cognition 2002, New York, ACM Press, USA.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass.

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

Moore CJ, Miles JC & Evans SJN, (1996a) “Establishing a knowledge base for bridge aesthetics”, Structural Engineering Review, 8(2/3), 247-258.

Moore C. J., Miles J. C., Evans S. J., (1996b) “Innovative Computational Support in Bridge Aesthetics”, Transp. Res Record, 1549, 1-11.

Ngo DCL, Teo LS & Byrne JG, (2003) “Modelling interface aesthetics”, Information Sciences (152), 25-46.

Parmee, I.C. (2002). “Improving problem definition through interactive evolutionary computation”. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2002), 16, pg. 185-202. Cambridge University Press, Printed in USA.

Peysakhov, M and Regli, W.C. (2003) , *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 11 Apr 2003, vol. 17, no. 2, pp. 155-68.

Rosenman, M. A. (1997a). “An exploration into evolutionary models for non-routine design”, *Artificial Intelligence in Engineering*, **11**(3):287-293

Rosenman, M. A. (1997b). “The generation of form using an evolutionary approach”, *in* D. Dasgupta and Z. Michalewicz (eds), *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, Southampton and Berlin, pp.69-85.

Rosenman, M.A. (1996) “A growth model for form generation using a hierarchical evolutionary approach”, *Microcomputers in Civil engineering, Special Issue on Evolutionary Systems in Design*, 1996, **11**:161-172.

Saunders, R. (2001) *Curious Design Agents and Artificial Creativity*, Ph.D. Thesis (Electronic Edition), Faculty of Architecture, The University of Sydney.

Stamps A. E., (1999) “Architectural detail, Van der Laan Septaves and Pixel Counts”, Design Studies(20), 83-97.