DESIGN 2006

INTRODUCING MACHINE LEARNING WITHIN AN INTERACTIVE EVOLUTIONARY DESIGN ENVIRONMENT

A. T. Machwe and I. C. Parmee

Keywords: interactive evolutionary design, machine learning

1. Introduction

The overall research focuses on the development of an Interactive Evolutionary Design System (IEDS) [Parmee 2002] which integrates machine-based evaluation of engineering criteria and rule-based aesthetic criteria with the designer's subjective aesthetic evaluation of design solutions. The research described in the paper relates to user-centric intelligent design systems and creativity in design. While research relating to artificial design environments is evident in the literature [Gero 2002], [Bentley 1999] there is little evidence of the integration of user evaluation, evolutionary search and exploration and machine learning. Furthermore, integration of aesthetic criteria within computer-based design has been limited to the development of theoretical models with little evidence of application based research [Moore et al. 1996; Saunders 2001]. The IEDS in this case addresses the conceptual design of bridges. For a more detailed description of the system as illustrated in Figure 1, the reader is directed to Machwe et. al. [2005]. The research decribed in the following sections concentrates upon the machine learning sub-system .
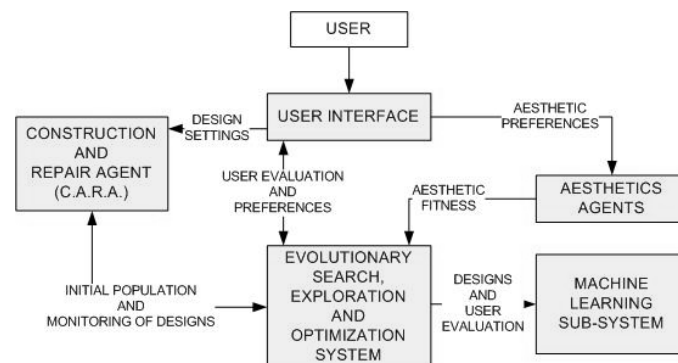

Figure 1. Interactive Bridge Design System

The designer interacts with the evolutionary process through the subjective evaluation of those designs considered high performance in terms of the machine-based criteria. The system displays images of the top N% (where N is preset by the designer) of designs based on engineering analysis, materials usage and simple rule-based aesthetic analysis. The user then assigns a user-defined fitness value to each design (User-assigned fitness: Ufit). The main purpose of the Machine Learning Sub-system is the on-line assimilation of the designer's subjective aesthetic preferences. This addresses a major problem in interactive evolutionary design systems relating to user fatigue caused by the evaluation of

excessive numbers of solutions. The intention is that, as the generations progress, the system reduces its dependence on human interaction and increasingly produces aesthetically pleasing solutions based upon the assimilated user preferences. This would result in a reducing degree of user-interaction and, in later generations, a completely machine-based process once user preferences have been adequately learned by the system.

2. Background

The nature of the interaction in the IEDS allows the use of supervised learning. The user ranking of generated solutions provides the required inputs and outputs of an online learning system [Mitchell 1997], [Bigus and Bigus 2004]. In this paper we look at three techniques implemented within the IEDS namely: Fuzzy rule based learning systems, Radial Basis functions (RBF) and finally Case based reasoning (CBR).

Design representation presented an initial problem in the research. Various authors [e.g. Mitchell 1997; Kolodner 1993] point out that the learning ability of any algorithm is only as good as the representation of the information to be learned. Thus, the essentially pictorial design has to be represented to suit various machine learning techniques such as back propagation neural networks and fuzzy rule based systems. If the representation is too rich then the machine learning system would be overloaded with the instances to be learned. If the representation is too lean then the system could miss out small but important differences between designs.
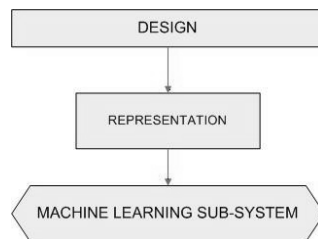

Figure 2. Representing Bridge Designs within the Machine Learning Sub-system

3. Representation of solutions

As fully described in Machwe et al [2005], the simple bridge design is basically divided into two separate collections namely the **Span Element** collection (containing elements forming the span of the bridge) and the **Support Element** collection (containing elements forming the support of the bridge). In the case of a free span bridge the **Support Element** collection will be empty. Figure 3 further clarifies the idea of using an object based representation.
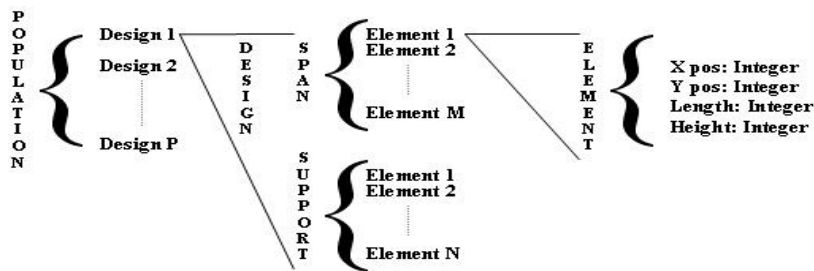

Figure 3. Details of the object based representation

Each Element is basically a rectangle with properties as shown in Figure 3. 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

METHODS AND TOOLS IN DESIGN PRACTICE

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.
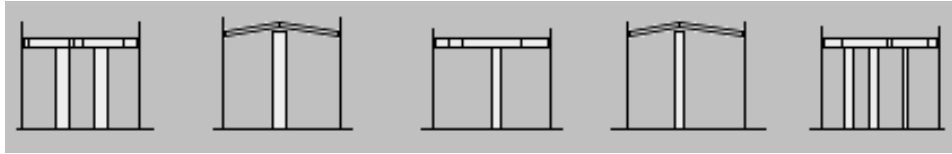

Figure 4. Two classes of bridges: Simply supported and Angled section spans

There are three kinds of bridges possible, two of which can be seen in Figure 4. The third type is simply supported at either end with no columns. Each type is treated as a separate class. Thus at the first level, this classification of designs helps the machine learning system categorise test cases whilst during the matching process classes ensure the comparing of like-with-like.

For the fuzzy rule based system each design class is represented by a set of fuzzy variables. Thus each class has its own fuzzy variable based representation within the machine learning sub-system. In case of simply supported spans the following fuzzy variables are used to represent the design:

Avgerage Span Thickness
Standard. Deviation of Span Thickness
Average Distance between supports
Standard. Deviation of distance between supports
Avgerage Thickness of supports
Standard Deviation of thickness of supports

For the angled section we use the following fuzzy variables to specify the aesthetic model:

Peak: Left, Central, Right
Difference in Span Thickness (Delta-Thickness)
Average Thickness
Column Thickness

The User assigned fitness (Ufit) fuzzy variable is the result of the rule. Thus the numeric fitness value assigned by the user to a design is translated into a fuzzy value and added as the result of that particular rule (generated from the design). A typical rule for an angled section bridge would look something like:

IF peak = Left AND delta-thickness = Right AND avg. thickness = High AND col. thickness = Low THEN ufit = Low

In the case of the RBF and Case based reasoning implementation a much more flexible representation has been utilised. The solutions are simply stored but, at the time of retrieval in CBR and during training in RBF, each solution is decomposed using geometrical techniques rather than the characteristic-based variable approach utilised in the fuzzy inference method. In the case of the angled span bridge the solution is decomposed by taking the geometrical properties of the left and right span element including length and thickness and the thickeness and horizontal (x) position of the support. In the case of the supported span the span section is divided into ten equal parts and the average thickness calculated. This gives an approximate profile of the span section. The greater the number of parts used the higher the resolution and more computationally expensive the retrieval becomes. The supports are again represented by their thickness and horizontal (x) positions.
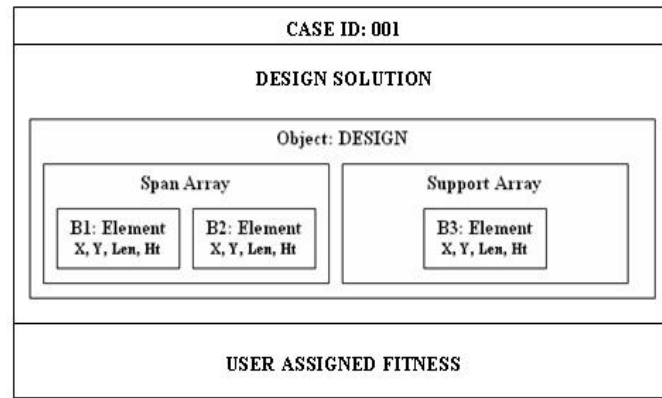
Figure 5. Structure of Test Data

For an RBF network we can assume the following formulation of the problem:

$$F(\mathbf{x}) = \text{Ufit} \tag{1}$$

Where **x** is the vector containing the average thickness of the ten sections that the span is divided into and F is the RBF approximated function [Orr 1996]. The user supplied ranking (Ufit) and the corresponding design gives the training data for the RBF. The basic storage structure to store data for RBF and CBR systems is shown in Figure 5.

4. Fuzzy Rule Based Learning System

The fuzzy rule based system uses the Information Gain heuristic to build a decision tree from the rules in the rule base. The decision tree is then used to evaluate new examples. The Information Gain heuristic as given by Shannon and Weaver [Bigus and Bigus 2004] is given by the following set of equations:

$$I(\frac{p}{p+n},\frac{n}{p+n}) = -(\frac{p}{(p+n)})\log 2(\frac{p}{(p+n)}) - (\frac{n}{(p+n)})\log 2(\frac{n}{(p+n)}) \tag{2}$$

$$\text{Remainder } (A) = \sum_{m}(\frac{p_i + n_i}{p+n})I(\frac{p_i}{p_i + n_i},\frac{n_i}{p_i + n_i}) \tag{3}$$

$$\text{Gain } (A) = I(\frac{p}{p+n},\frac{n}{p+n}) - \text{Remainder } (A) \tag{4}$$

Here p and n give the total number of positive and negative examples (rules with positive or negative results) in the data set to be used for decision tree creation. In case we want to find the information gain for attribute A (or variable A) which takes m different values (say) we just calculate the Remainder of A (where p_i and n_i give positive and negative examples with A taking the ith value out of the possible m). The Gain always takes a value between 0 and 1.

4.1 Results

The information gain was calculated for each variable in both the models using a set of generated rules.

METHODS AND TOOLS IN DESIGN PRACTICE

Tables 1a and 1b Information Gain data for Angled and Simply supported bridges

Angled Beam Bridges	
Variable Name	Gain
Skew	*0.218*
Delta-Thickness of Two span elements	*0.170*
Average Thickness of Span	*0.255*
Column Thickness	*0.334*

Simply Supported Beam Bridges	
Variable Name	Gain
Std. Dev. Of Span Thickness	*0.077*
Avg. Distance between Supports	*0.009*
Std. Dev. Of distance between supports	*0.029*
Average Thickness of columns	*0.030*
Std. Dev. Of thickness of columns	*0.027*

A positive example is a rule where the **Ufit** value was greater than five and a negative example when the **Ufit** value is less than or equal to five.Rules are collected during a run and analysed offline for the information gain of each variable. The information gain data for a standard run is given in Table 1a and 1b, from which we can see that information gain is low when we use fuzzy variables to decompose the design into a set of rules.

5. Radial Basis Functions

Radial basis functions are used for supervised learning. An RBF network consists of an input layer fully connected to a hidden RBF layer which is made up of *h* number of hidden units. The output of the hidden layer is added using weighted sum and given to the output layer (consisting of a single unit). Each hidden unit represents a radial basis function which is characterised by a center (c) and radius (r) of coverage (if using Gaussian functions as the radial function).

$$H(\vec{x}) = \exp\left(-\frac{(\vec{x} - \vec{c})^2}{r^2} \right)$$
(5)

To completely define an RBF we need to first find the optimal number of hidden units with center and radius defined and the weight vector for the hidden units. Once the centers and radii are defined the weight vector can be obtained using the *normal equation*. The RBF once setup can be tested using a validation technique such as Generalised Cross Validation (GCV).

There are multiple techniques for setting up an RBF such as backward elimination and forward selection. Forward selection was used in this study since it offers many advantages including not requiring the number of hidden units to be fixed in advance and being computationally efficient [Orr 1996].

The hidden units are initialised with random centers and radii for each run and added one by one. After each addition the GCV value is calculated and recorded. As each hidden unit is added the sum square error is reduced but there is a danger of overfit taking place. The GCV value on the other hand stops decreasing after a certain number of units are added and starts increasing which signifies overfitting.

5.1 Results with RBF

Table 2. Data from Radial Basis Functions using Forward Selection

Total Number of Cases (C) ~ 700	CASE SIZE			
	C	C/2	C/3	C/4
Minimum Number of Functions	51	24	19	29
Maximum Number of Functions	53	57	42	38
Minimum GCV	11.58	24.43	40.613	48.21
Maximum GCV	11.88	25.425	42.892	52.74
Standard Deviation (Number of Functions)	1.15	15.59	9.47	3.49
Standard Deviation (GCV)	0.16	0.45	0.99	1.8

When testing with radial basis functions a total of 700 design cases were collected over multiple evolutionary runs with the same user. The maximum and minimum values for error are shown for different test case sizes (when using Forward Selection) in Table 2. As the test case size is increased the error reduces and the number of hidden units required converges to approximatily 50. We also see that with increasing test case size the standard deviation of error obtained from multiple runs also reduces. In case of small test case size (C/3 ~ 233 cases and C/4 ~ 175 cases) the minimum error is quite large where as with a case size of around 350 cases the error is halved and with case size of around 700 cases the error is halved again. The results suggest that the required level of on-line learning is not achievable and that the RBF is more suited to an off-line approach since the RBF based learning system requires a large number of cases to achieve low error rates. The typical case library size for an RBF would be around 350 cases (see Table 2.). For a user ranking ten solutions per generation it would take atleast thirty five generations to achieve low error rates. The cognitive load of continuously evaluating ten solutions per generation for thirty five generations defeats the purpose of having a learning system to assist the designer.

6. Case Based Reasoning

Due to the problem of representation Case Based Reasoning (CBR) is one of the most promising techniques [Mitchell, 1997]. A major advantage of using CBR is that the design information can be stored as it stands without using any other representations that can take away essential information [Kolodner 1993]. The retrieval part of the CBR uses nearest neighbour distance metrics to measure the difference between the new design and the designs in the case base. The design closest to the new design has its user assigned fitness (Ufit) awarded to the new design (Figure 6).
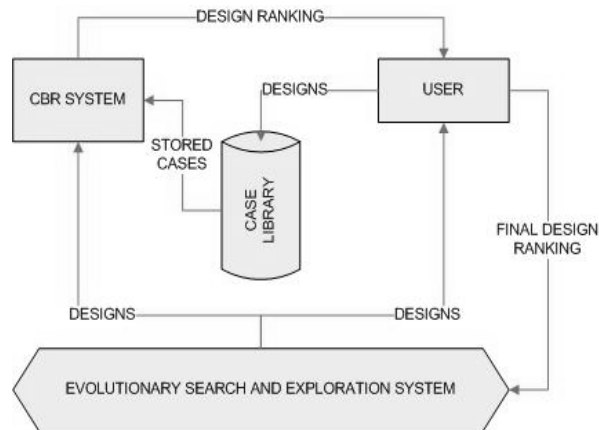

Figure 6. The Case Based Reasoning system integrated with the IEDS

METHODS AND TOOLS IN DESIGN PRACTICE

6.1 Results with CBR

The CBR learning is online and cases are not carried over from past runs. Initially the case library is empty. After the first generation of user evaluations the case library starts to fill. Solutions not examined by the user are assigned a zero fitness. Once there are a number of cases in the case library the machine learning system starts ranking solutions and the user has the option to change the machine assigned rank. The testing carried out consisted of recording the number of changes the user makes to the machine assigned rank in each generation in each of fifty seperate runs of variable length. The values then were grouped by generation number and then averaged.

If the learning system is operating successfully then the number of changes made by the user would decrease rapidly with each generation as the machine assimilates user preference. This is assisted in part by the convergence of the population i.e. as the population converges it improves the resolution around the solutions initially preferred by the user. It must be kept in mind that without a machine learning sub-system the user would have to rank all the N% solutions shown to him consistently after every generation to ensure that the population converges to the desired designs.
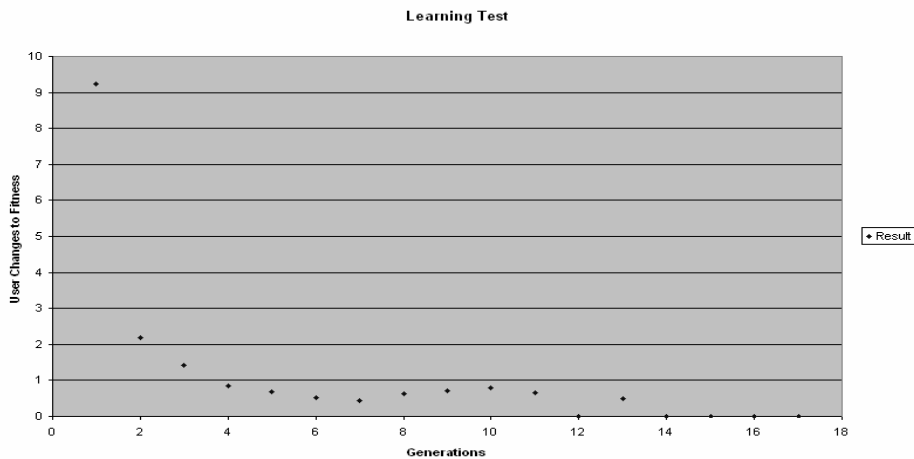

Figure 7. Number of user made changes to machine assigned fitness during different generations

Thus we can see that the user interaction decreases as the generations proceed. The user interaction required drops from the average of around 9.2 changes out of 10 solutions to circa 0.5 changes out of 10 solutions in generation six and drops to zero by generation fourteen. This illustrates that the CBR sub-system is operating satisfactorily within the IEDS (Figure 7).

7. Conclusion

We can draw several important conclusions from the tests carried out using the different techniques. The results confirm the findings of other researchers i.e. that even in the case of machine learning, representation plays an important role.. From the failure of fuzzy rule based systems we can conclude that when complex design properties have to be extracted it is difficult to do so using a simple set of variables. Perhaps a more comprehensive fuzzy model might lead to better results but the loss of information during decomposition would always remain a problem as can be seen from the success of the geometrical representation.

From the experiments with RBF we can see that RBF, with a little more sophisticated training technique, could provide an ideal machine learning sub-system in the offline, profile based interaction but for online learning, using a smaller training data set, the error between expected and actual output is too large.

The CBR approach does seem to provide a way forward in the case of online learning. Figure 7 provides us with insight into the performance of the learning system and also raises an interesting question. If we examine the graph between generations nine and eleven we find a slight increase in

required user interaction. Similarly between generations twelve and fourteen we find another (lower) rise. A possible reason for this could be a mid-course correction introduced during certain runs by the user. Such a mid-course correction can be caused by two designs being highly similar and yet having different past user assigned fitness values. This kind of 'confusion' can be attributed to the Euclidian distance measure being used to retrieve the cases.

Acknowledgement

References

Bentley, P.J. ed. "Evolutionary Design By Computers". 1st Edition. Morgan-Kaufmann, USA. 1999

Bigus, J. and Bigus, J. "Constructing Intelligent Agents Using Java: Professional Developer's Guide", 2nd Edition. John Wiley and Sons. 2004.

Gero, J.S. "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.2002

Kolodner, J. "Case-based Reasoning". Morgan Kaufmann Publishers 1993.

Machwe, A. Parmee, I.C. and Miles, J.C. "Integrating Aesthetic Criteria with a user-centric evolutionary system via a component based design representatio".. Proceedings of International Conference on Engineering Design 2005, Melbourne, Australia. 2005

Mitchell, T.M. " Machine Learning". McGraw Hill International.1997.

Moore C.J., Miles J.C. & Evans S.J.N., "Establishing a knowledge base for bridge aesthetics", Structural Engineering Review, 8(2/3), 1996. 247-258.

Orr, M.J.L. "Introduction to Radial Basis Function Networks", Center for Cognitive Science, University of Edinburgh, Edinburgh, Scotland, 1996, http://www.cns.ed.ac.uk/people/mark/manual.ps.

Parmee, I.C. "Improving problem definition through interactive evolutionary computation". Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2002), 16(3), 2002. pg. 185-202. Cambridge University Press, Printed in USA.

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

Azahar Machwe
ACDDM Group, Faculty of Computing, Engineering and Mathematical Sciences,
University of the West of England, Coldharbour Lane, Bristol, UK, BS16 1QY
Email: Azahar.machwe@uwe.ac.uk