

John S Gero and Mary Lou Maher (eds) (1998)  
*Computational Models of Creative Design IV*  
Key Centre of Design Computing, University of Sydney

## **ADAPTING EVOLUTIONARY COMPUTING FOR EXPLORATION IN CREATIVE DESIGNING**

JOHN GERO AND VLADIMIR KAZAKOV  
*Key Centre of Design Computing*  
*Department of Architectural and Design Science*  
*University of Sydney NSW 2006 Australia*

**Abstract.** *This paper introduces a modification to genetic algorithms which provides computational support to creative designing by adaptively exploring design structure spaces. This modification is based on the re-interpretation of the GA's crossover as a random sampling of interpolations and its replacement with the random sampling of direct phenotype-phenotype interpolation and phenotype-phenotype extrapolation. Examples of the process are presented*

### **1. Introduction**

Whilst the notion of creative designing is well established, there is no consensus on a formal model of creative designing processes. In design research one common approach to this is to distinguish two classes of designing: routine designing and non-routine designing (Gero, 1990). Non-routine designing maps onto creative designing. In routine designing all the variables which specify designs are given in advance. This means that the space of possible designs is known a priori, each point in this space can be constructed and evaluated directly. What needs to be done is to search this space in order to locate an appropriate or most appropriate design. The result here is the “best” design from this space. In non-routine designing the result is the “best” space of possible designs as well as the “best” design from this space. Processes which modify the design space of the search problem are called exploratory processes. The computational modelling of exploratory processes remains an important problem in creative designing. Here we will consider one type of exploratory process, one which affects only one component of design state space, namely the structure space. The behaviour and function components of design space remain unchanged. We consider only such processes where the structure space can be enlarged during an exploration.

The corresponding process of design structure space evolution over time is shown in Figure 1.

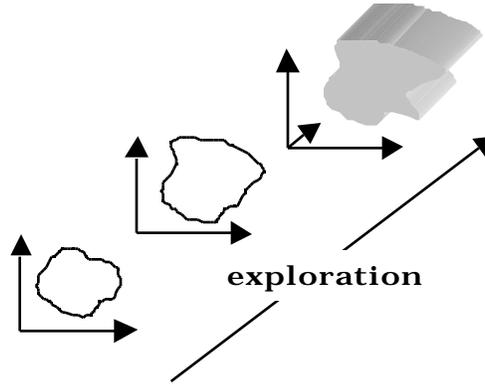


Figure 1. Non-routine design as an evolution of design space

In this paper we assume that when routine designing is modeled then design spaces are searched using genetic algorithms (GAs). We will construct an extended version of GAs which can be used to model a creative designing process. We will demonstrate that it is able to operate in a pure search mode when the design space stays the same, in a pure exploratory mode when the design space is being modified or in a mixed mode when both these processes occur simultaneously.

Initially, we will briefly characterise genetic algorithms prior to introducing a re-characterisation of the crossover operation in genetic algorithms as a class of interpolations. This will allow us to draw on other members of that class.

## 2. Genetic Algorithms

### 2.1 STANDARD GA

In a standard GA a search problem is represented using two spaces. The first space,  $\mathbf{G}$ , is the space of ordered character strings,  $\mathbf{g} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$  (which are called genotypes), whose components,  $\mathbf{e}_i$ , are drawn from some fixed finite alphabet,  $\mathbf{A}$ . For simplicity, we assume that all genotypes have the same length,  $n$ . The second space,  $\mathbf{P}$ , is the state space of the solutions (design structure space in our case). Its elements,  $\mathbf{p}$  (design structures), are called phenotypes. Each genotype corresponds to a unique phenotype. We assume that the mapping  $\mathbf{M}: \mathbf{G} \rightarrow \mathbf{P}$  is given a priori and is fixed. The fitness function  $F(\mathbf{p})$ , which measures the quality of the phenotype  $\mathbf{p}$ , is also fixed. GAs employ three operations: selection (which carries random sampling of the points from the current

population biased towards high fitness points), the crossover operator (which generates new genotypes from the sample produced by selection) and mutation which randomly changes the results of the crossover. These three operations are applied repeatedly to the current population until the next generation is created, Figure 2.

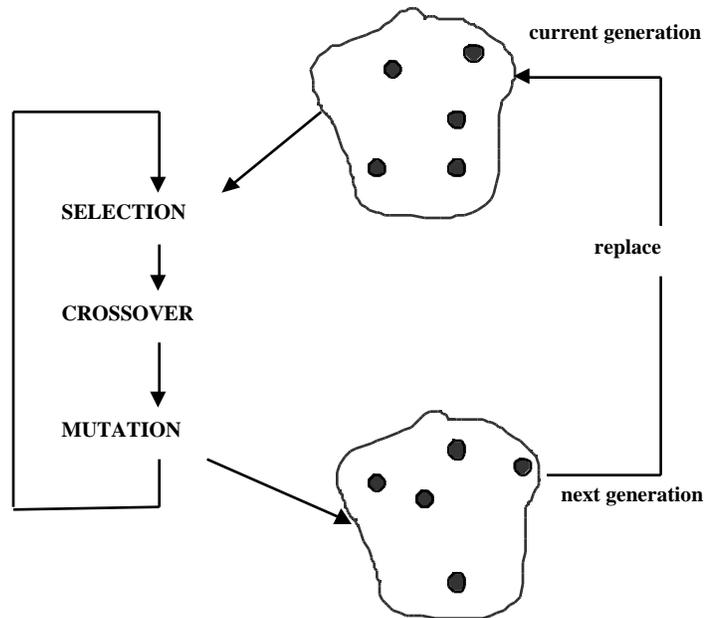


Figure 2. The architecture of the standard genetic algorithm

Our plan is to re-interpret the crossover as randomization of a particular case of phenotype-phenotype interpolation and then replace it with randomization of more general phenotype-phenotype interpolation and/or extrapolation. Except for this replacement and the possible replacement of the genetic mutation with the phenotype mutation we intend to keep the general structure of GA as shown in Figure 2.

### 1.1 GA'S CROSSOVER AS RANDOM SAMPLING OF INTERPOLATING POINTS

The most important computational operation which drives GA search is crossover,  $C$ . In the simplest case it is a binary operation,  $C(\mathbf{g}_1, \mathbf{g}_2) = \mathbf{g}_e$ , which takes two genotypes  $\mathbf{g}_1$  and  $\mathbf{g}_2$  as parameters and produces a new genotype  $\mathbf{g}_e$ . It does this by cutting each genotype into two pieces and

then concatenating the front piece from one genotype with the rear piece from the other. The cut points (crossover points) are the same in both genotypes and are chosen randomly.

Formally, any genotype,  $\mathbf{g}_c$ , produced by a crossover operator from genotypes  $\mathbf{g}_1$  and  $\mathbf{g}_2$  can be written as an interpolation:

$$\mathbf{g}_c(t) = \mathbf{f}(t)\mathbf{g}_1 + (\mathbf{I}-\mathbf{f}(t))\mathbf{g}_2, t = 0, 1, \dots, n$$

where  $\mathbf{I}$  is a unit  $n$ -dimensional matrix with all diagonal elements equal to 1 and all other elements equal to 0,  $\mathbf{f}(t)$  is the  $n$ -dimensional matrix obtained from the unit matrix by setting all diagonal elements from the  $t$ -th to the  $n$ -th to zero,  $\mathbf{f}(0)=\mathbf{I}$  and  $\mathbf{f}(1)=\mathbf{O}$ , where  $\mathbf{O}$  is the  $n$ -dimensional zero matrix.

From this characterisation the crossover operation can be viewed as a random sampling of interpolating genotypes between two basic points  $\mathbf{g}_1$  and  $\mathbf{g}_2$ . Note, that this linear matrix interpolation, which corresponds to the standard one-point crossover, is only one of many possible methods of interpolation between two genotypes in genotypic space of the following form:

$$\mathbf{g}_i(t) = \mathbf{c}_1(t)\mathbf{g}_1 + \mathbf{c}_2(n-t)\mathbf{g}_2,$$

where operators  $\mathbf{c}_1(t)$  and  $\mathbf{c}_2(n-t)$  obey the condition  $\mathbf{c}_1(0)=\mathbf{I}$  and  $\mathbf{c}_1(n)=\mathbf{O}$  and  $\mathbf{c}_2(0)=\mathbf{I}$  and  $\mathbf{c}_2(n)=\mathbf{O}$ . The crossover induced interpolation  $\mathbf{g}_c(t)$  is singled out from many other possible interpolations  $\mathbf{g}_i(t)$  by the condition that the sum of the Hamming distances from  $\mathbf{g}_c(t)$  to  $\mathbf{g}_1$  and to  $\mathbf{g}_2$  plus a penalty function (any kind of standard optimization penalty function will do) is to be optimized for two sequential coordinates in  $\mathbf{g}_c(t)$  one of which coincides with the component of  $\mathbf{g}_1$  and the other which the component of  $\mathbf{g}_2$ . Different versions of crossover can be constructed by choosing different conditions imposed on the interpolation points.

Since each genotype corresponds to a unique phenotype, the crossover-induced interpolation operation between two genotypes maps onto an interpolation operation between two corresponding phenotypes  $\mathbf{p}_1=\mathbf{M}(\mathbf{g}_1)$  and  $\mathbf{p}_2=\mathbf{M}(\mathbf{g}_2)$ . If  $\mathbf{p}_c(t)=\mathbf{M}(\mathbf{g}_c(t))$  for  $t=0, 1, \dots, n$  and assuming that  $\mathbf{P}$  is a linear space we can fit a path between  $\mathbf{p}_1$  and  $\mathbf{p}_2$  and  $\mathbf{p}_c(t)$ , using the following formula:

$$\mathbf{p}_c(t) = \mathbf{f}^c(t)\mathbf{p}_1 + \mathbf{q}^c(n-t)\mathbf{p}_2, t=0, 1, \dots, n$$

where  $\mathbf{f}^c(t)$  and  $\mathbf{q}^c(t)$  are operators which depend continuously on  $t$ . Since  $\mathbf{p}_c(0)=\mathbf{p}_2$  and  $\mathbf{p}_c(1)=\mathbf{p}_1$ , the weakest conditions these operators must satisfy are  $\mathbf{f}^c(0)=\mathbf{I}$ ,  $\mathbf{f}^c(1)=\mathbf{O}$  and  $\mathbf{q}^c(0)=\mathbf{I}$ ,  $\mathbf{q}^c(1)=\mathbf{O}$  (where  $\mathbf{I}$  is the unit operator whose application to any phenotypes gives the same phenotype and  $\mathbf{O}$  is the zero operator whose application to any phenotype gives an empty phenotype). If we use any operators  $\mathbf{f}(t)$  and  $\mathbf{q}(t)$  which differ from  $\mathbf{f}^c(t)$  and  $\mathbf{q}^c(t)$  but still obey these conditions then this formula will

produce interpolation points which are different from the crossover-induced ones.

As an example, in this paper we consider phenotypes (design structures) which are 2-d or 3-d shapes. These designs are represented using an F-representation ([Pashko, et al, 1995](#)), as real value functions  $F(\mathbf{x})$  such that  $F(\mathbf{x}) > 0$  is inside the object,  $F(\mathbf{x}) = 0$  is on its boundary and  $F(\mathbf{x}) < 0$  is outside of the object. Here  $\mathbf{x}$  is a 2-d or 3-d vector with a defined feasible bounded region  $D: \mathbf{x} \in D$ . Let us consider two phenotypes  $F_1(\mathbf{x})$  and  $F_2(\mathbf{x})$  and the transformation:

$$F_c(t, \mathbf{x}') = t \cdot v(\mathbf{x}') F_1(\mathbf{x}') + (1-t) \cdot w(\mathbf{x}') F_2(\mathbf{x}'), \quad \mathbf{x}'(t, \mathbf{x}) \in D \rightarrow D,$$

where  $t$  is a scalar which changes from 0 to 1, and  $v(\mathbf{x})$  and  $w(\mathbf{x})$  are non-negative scalar functions (called modulating functions) of  $\mathbf{x}$  and  $\mathbf{x}'(t, \mathbf{x})$  is a coordinate transformation which produces a homomorphism from  $(D, t)$  to  $D$  for any  $t \in [0, 1]$  and depends continuously on  $t$ . Usually the mapping  $\mathbf{x}'(t, \mathbf{x})$  is chosen to provide a correspondence between the positions  $i_j(t)$  of similar features (points, line-segments, etc.) in the two shapes:  $\mathbf{x}'(t, i_j(0)) = i_j(t)$ . The mapping  $\mathbf{x}'(t, \mathbf{x})$  can be constructed using the algorithm proposed by [Fujimura and Makarov \(1997\)](#). The only condition which is necessary for this formula to define an interpolation between  $F_1(\mathbf{x})$  and  $F_2(\mathbf{x})$  (that is,  $F_c(0, \mathbf{x}) = F_1(\mathbf{x})$  and  $F_c(1, \mathbf{x}'(1, \mathbf{x})) = F_2(\mathbf{x})$ ) is the positivity of functions  $v(\mathbf{x})$  and  $w(\mathbf{x})$  for all  $\mathbf{x} \in D$ .

Hence, the goal is to find such positive functions  $v^c(\mathbf{x})$  and  $w^c(\mathbf{x})$  (we shall call them crossover modulating functions). Such functions must satisfy the requirement that the interpolation path  $F_c(t, \mathbf{x})$  fits the mapping of the genetic interpolation  $\mathbf{g}_c(t)$  induced by a crossover in genotypic space onto phenotypic space. With such functions, the GA search can be formulated completely in terms of phenotypic space without any explicit references to genotypic space (except implicit information built into functions  $v^c(\mathbf{x})$  and  $w^c(\mathbf{x})$ ). Whether or not this can be done (that is, whether or not  $v^c(\mathbf{x})$  and  $w^c(\mathbf{x})$  exist) needs to be investigated for each particular system. Note that for this problem phenotype interpolation turns out to be the problem of interpolation of real functions.

Now, that we have re-interpreted GA crossover as an interpolation operation in both genotypic and phenotypic spaces, we are in a position to produce a generalization of GA crossover which yields exploration, that is, enlargement of the phenotypic space, which is to be subsequently searched. In the next section we will do that and establish when this

generalization will be productive, that is, when it could yield this enlargement.

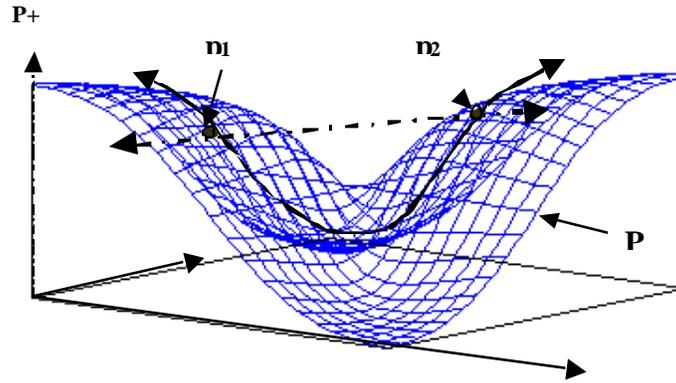
### 3. Generalized Interpolation as a Tool for Design Space Enlargement

Let us consider a space of designs,  $\mathbf{P}$ , which are represented using some representation  $\mathbf{R}$ . That is, each object is represented as a finite set of parameters  $p = \{p_1, \dots, p_m\}$  (not necessarily with numerical values) with associated feasibility constraints ( $f(p) = 0$ , for definiteness) and range constraints (which we consider to be a part of representation itself). These parameters jointly with the representation are sufficient to completely describe an object. A representation is *generalizable* if it is possible to choose parameters,  $p$ , such that they obey all the range constraints, part of the feasibility constraints and do not obey the rest of the feasibility constraints,  $f(p) = 0$ , and these parameters jointly with the same representation are still able to describe some feasible objects. In other words, a *generalizable* representation has the potential to support a broader class of objects and this class of objects can be produced by simple manipulations with the parameters of the objects in the current space. Formally, the space  $\mathbf{P}$  is generalizable with respect to the representation  $\mathbf{R}$  if a superspace  $\mathbf{P}^+$  exists ( $\mathbf{P} \subset \mathbf{P}^+$ ) whose states can be represented using  $\mathbf{R}$ .

#### 3.1 ILLUSTRATION

Let us give a simple geometric illustration of the constructions which were described above. The genotypic space  $\mathbf{G}$  maps onto the phenotypic space  $\mathbf{P}$ . The representation which supports  $\mathbf{P}$  is generalizable, therefore another space  $\mathbf{P}^+$  (which is a superspace with respect to  $\mathbf{P} \subset \mathbf{P}^+$ ) can be constructed which is supported by the same representation. The standard crossover-driven GA search consists, from this viewpoint, of drawing straight line segments (in metric space with the distance as the sum of Hamming distances plus the penalty we described earlier) between trial points in  $\mathbf{G}$  space. It maps these segments onto the interpolation trajectories in  $\mathbf{P}$  space, denoted with the bold line in Figure 3, and then samples points from this trajectory. The generalized crossover consists of interpolating trial points directly in  $\mathbf{P}^+$  using trial points from  $\mathbf{P}$  as the end points of interpolation. They are shown in Figure 3 with the dotted line. The expectation is that since these end points belong to the established search space  $\mathbf{P}$ , the exploration due to interpolation in the enlarged  $\mathbf{P}^+$  will not distort the consistency and viability of the space  $\mathbf{P}$  too much. The critical effect can be noticed in Figure 3, namely that the

interpolation in  $\mathbf{P}^+$  does not lie in  $\mathbf{P}$ . Hence, these interpolations have the capacity to produce designs outside the original state space. The interpolation process expands the state space of possible designs and is therefore an exploratory process.



*Figure 3.* The illustration of the crossover-induced interpolation in  $\mathbf{P}$  and direct interpolation in enlarged space  $\mathbf{P}^+$ . The enlarged space  $\mathbf{P}^+$  represents the complete 3-d space and the set  $\mathbf{P}$  represents the surface in it. The solid line represents an interpolation between points  $p_1$  and  $p_2$  in  $\mathbf{P}$ , whilst the dotted line represents an interpolation in  $\mathbf{P}^+$  between these points. Two external sections of the solid line with arrows represent crossover-induced extrapolation. Two external sections of the dotted line with arrow represent extrapolation based on non-crossover based interpolation

#### 4. Extrapolation as an Exploration Tool

Since we have already converted the crossover operation into interpolation, it may not be necessary to restrict the search area to the points in between these phenotypes (which we do if we use only interpolation). Intuitively extrapolation (if restricted to the states which are close to the parental points) seems to keep only marginally less information from these points but is capable of going outside the current space. Hence, we can further increase the potential exploratory power of a GA modified in this way by completely replacing the crossover operation with the random choice of one of the points given by the interpolating formula  $\mathbf{p}_c(t)$  extended over the wider range of  $t$  values than  $[0, 1, \dots, n]$ , for example over  $[-k+1, -k, \dots, -1, 0, 1, \dots, n, n+1, \dots]$  to encompass extrapolation.

Geometrically this case corresponds to the situation when the continuation of the crossover-based interpolating path between two parental points on the surface  $\mathbf{P}$  yields path which leaves this surface, see the external sections of the solid line beyond two basic points in Figure 3.

## 5. Choosing the Form of Phenotypic Interpolation

When we constructed the computational operations which could form the basis of our algorithm we assumed that the non-crossover based interpolating operators  $\mathbf{f}(t)$  and  $\mathbf{g}(t)$  (or modulating functions  $v(\mathbf{x})$  and  $w(\mathbf{x})$  in our example) are somehow chosen. Their choice is critically important for the algorithm to be productive. Three strategies could be employed for finding such “good” interpolating operators. In the first approach, some general parametric forms of these operators are chosen a priori  $\mathbf{f}(e,t)$  and  $\mathbf{g}(e,t)$  (or  $v(e,\mathbf{x})$  and  $w(e,\mathbf{x})$ ) where  $e$  is the vector of parameters  $e \in E$  within the feasible range  $E$ . Then either the additional random sampling is done with respect to  $e$  simultaneously with  $t$ :

$$p^* = \text{random}_e \text{random}_t p_c(e,t),$$

or the auxiliary search on  $e$  is carried out:

$$p^* = \min_e \text{random}_t p_c(e,t).$$

The second approach to the choice of interpolating formula (or modulating functions) is based on the following strategy. First, the set of properties required of the states in an enlarged space in order to be valuable are established. Which properties should be preserved from the original space in the enlarged space need to be decided upon. Second, an interpolating operator with this set of properties is constructed. Various shape transformation algorithms which preserve different properties of the end point designs have been invented which assist in this process (eg Beier and Neely, 1992, Wolberg, 1990). The process is still largely experimental. In the example we demonstrate how these algorithms are re-interpreted as the machinery to provide exploration modelling when using GAs during creative designing.

In the third approach the same set of properties required from the states in the enlarged design space are used as a “viability” test which is applied to each new design generated by generalized crossover. Only those new points which pass it are placed into a new generation. Of course, this is a search where there is no guarantee here that such points can be found or even exist.

Since design points produced by the phenotype-phenotype interpolation in the general case are not supportable by the problem’s genotype, one has to design a new operation which coincides with the standard genetic mutation if it is valid and is defined in situations when it is not valid. It seems natural to do this by directly mutating the phenotypes.

## 6. Example

As an example, we consider the problem of designing a cross-section of a beam. The structure space consists of cross-sections of the pre-defined shape with a fixed area, Figure 4. This shape is described by 4 parameters (the width and height of the top and bottom rectangular flange with ranges [1,10] and [1,4] and the width and height of the rectangular web with ranges [1,4] and [4,10]). The area of the cross-section is fixed and equals 500. The fulfillment of this area constraint is guaranteed by scaling the shape. The problem has a two-component behaviour (fitness) function  $\mathbf{F}$  which consists of the moment of inertia  $\mathbf{I}$  and the section modulus  $\mathbf{Z}$ .



*Figure 4.* The cross-section template which defines the original structure space

The standard GA gives the Pareto-set of designs for the problem and is shown in Figure 5.

Then the extended GA was initiated, where the crossover is replaced with the minimisation of direct phenotype-phenotype interpolation, between the points found by the GA in the original problem's space. Some of the improved designs (designs 4 and 5) found by this procedure are shown in Figure 6. They were generated by a "linear" interpolation procedure with constant modulating functions. The corresponding interpolating sequence is shown in Figure 6.

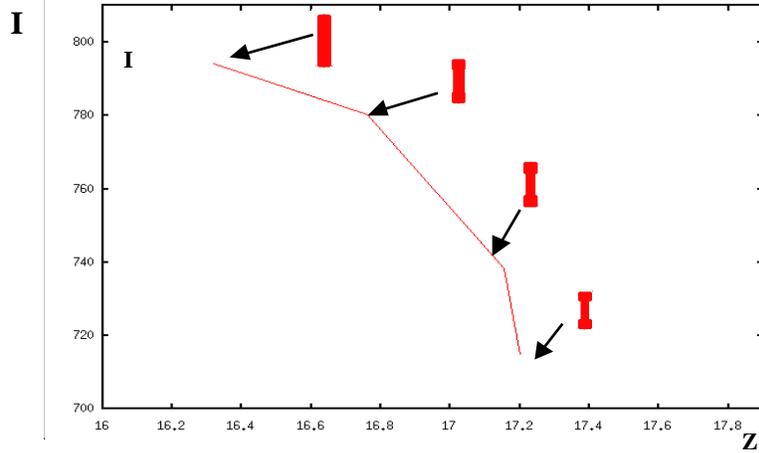


Figure 5. The Pareto set for the initial structure space with the corresponding shapes

The solution obtained clearly does not belong to the original structure space.

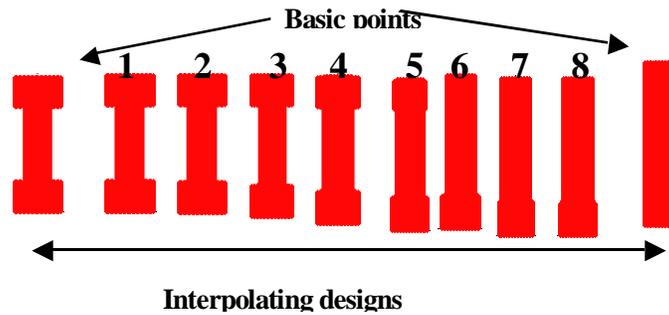


Figure 6. The “linear” phenotype-phenotype interpolation

The interpolating sequence generated by the “nonlinear” interpolation (with variable modulating functions) is shown in Figure 7.

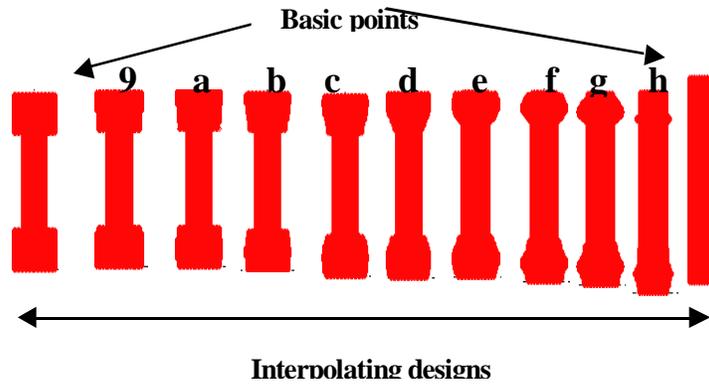


Figure 7. The “non-linear” (with non-constant modulating functions) phenotype-phenotype interpolation

As one would expect, it produces greater deviations from the original structure space and a greater variety of designs. More “exotic” (ie increasingly different from the original space) design generated by this “nonlinear” process is shown in Figure 8.

Figure 8. Example of “exotic” or increasingly unexpected design

Figure 9 shows the Pareto performance of some of the results produced from the use of the interpolating functions. It is interesting to note that some of the designs produced perform better than the original designs, ie they dominate some of the original designs.

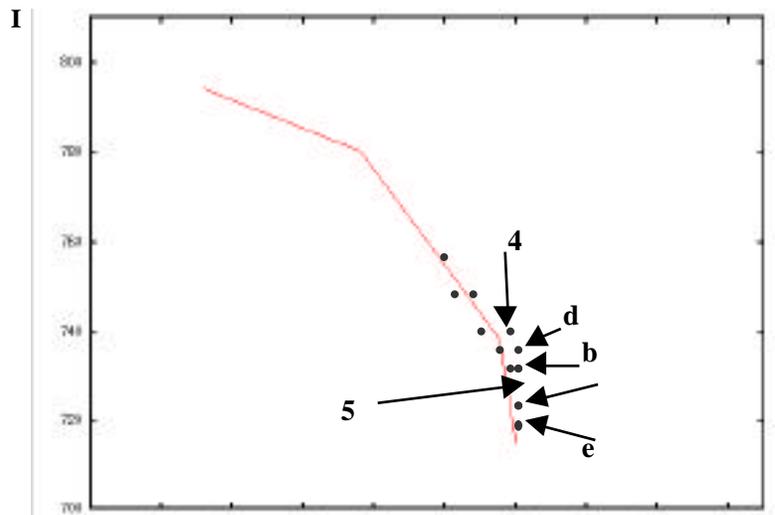


Figure 9. The location of the interpolated designs plotted on the Pareto front of the



original design space.

## 7. Discussion

This paper can be seen as a generalisation of a number of other related approaches (see for example, Graf and Banzhaf, 1995) where cross-image

interpolation has been used to replace a GA's crossover operator. Our processes are more general because we use a more general type of phenotype-phenotype (cross-object and not cross-image) interpolation which can be reduced to cross-image interpolation in a particular case only. When using GAs there is a clear difficulty in developing exploration processes at the direct genotype level. Any genotypic representation can be reduced to a bit string of length  $n$ . All possible genotypes lie within the space of  $2^n$  possible designs. Without either increasing the length of the genotype or introducing new members of the alphabet (beyond 0 and 1), it is not possible to expand the state space. The approach described in this paper solves this problem by developing a homomorphism between the phenotype and its genetic representation. It does away with the separate bit string genotype representation, replacing it with this homomorphism after the exploration process. The interpolation and extrapolation processes operate on the phenotype, changing it. As a consequence of this homomorphism a new genotypic representation is constructed each time exploration occurs.

One of the well-established notions related to creative designing processes is that an important means of characterising them is to determine whether they have the capacity to expand the state space of possible designs - exploration (Gero, 1994). Of the three state spaces used to describe designs (function space, behaviour space and structure space), only the structure space has the capacity to directly produce novel designs as it is expanded. The other spaces can also be expanded and hence produce either novel interpretations or indirectly force the expansion of the structure space (Gero and Kazakov, 1998). There are two classes of expansion or modification operators applicable here. The first class contains those operators which rely largely on external knowledge which is applied to the values of the existing space and as a consequence expand it. The second class contains those operators which make use of emergent features in the design space and use those to expand the design space. The extrapolation operator described here belongs to the first class.

One of the difficulties in deriving operations which are capable of expanding a design space is the need for the operators to have the capacity to add variables to the range of variables used to describe the space originally. There have been a number of approaches adopted previously. These include splitting a single variable into two variables (Aelion et al., 1992), importing variables from other design spaces using combination or analogy, and emerging new features and reverse engineering new variables to describe them. The approach adopted here distinguishes itself from these and other approaches in that it opens a

range of possible new variables. It does so by recasting the structure space from being restricted to the surface defined by the variables and the original processes of genetic algorithms to being a larger potential space, Figure 3. Which parts of that larger space are used is dependent on the interpolation/extrapolation functions used and the values of the variables used when the functions are applied. The range of applicable functions may well be unlimited. Each function potentially produces a different trajectory outside the original surface and each trajectory represents potentially different structure variables which are required to describe the resulting design.

As can be seen from the example the resulting designs are unpredictable in the sense that they are unexpected given only knowledge of the original designs and of the interpolation/extrapolation functions. In this sense the process matches well the meaning of exploration both in the technical sense used in this paper and in the natural language sense. The designs produced by the system demonstrate both the novelty and unexpectedness of what can be generated.

### Acknowledgements

This work is supported by a grant from the Australian Research Council. Computing resources are provided by the Key Centre of Design Computing.

### References

- Aelion, V., Cagan, J. and Powers, G.: 1992, Input variable expansion - an algorithmic design generation technique, *Research in Engineering Design* **4**:101-113.
- Beier, T. and Neely, S.: 1992, Feature-based image metamorphosis, *Computer Graphics (Proc. SIGGRAPH)*, pp.35-52.
- Fujimura, K. and Makarov, M.: 1997, Homotopic shape deformation, *International Conference on Shape Modeling and Applications*, Aizu-Wakamatsu, pp.215-225.
- Gero, J. S.: 1990, Design prototypes: a knowledge representation schema for design, *AI Magazine* **11** (4):26-36.
- Gero, J. S.: 1994, Computational models of creative design processes, in T. Dartnall (ed), *Artificial Intelligence and Creativity*, Kluwer, Dordrecht, pp.269-281.
- Gero, J. S. and Kazakov, V.: 1998, Using analogy to extend the behaviour state space in creative designing, in J. S. Gero and M. L. Maher (eds), *Computational Models of Creative Design IV*, Key Centre of Design Computing, University of Sydney, Sydney, Australia (this volume).

- Graf, J. and Banzhaf, W.: 1995, Interactive evolution in civil engineering, in J. S. Gero, M.L. Maher and F. Sudweeks (eds), *Preprints Computational Models of Creative Design*, Key Centre of Design Computing, University of Sydney, Sydney, pp.303-316.
- Pashko, A.A., Adzhiev, V.D., Sourin, A. I. and Savchenko, V. V.: 1995, Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer* **11** (8):429-446.
- Wolberg, G.: 1990, *Digital Image Warping*, IEEE Computer Society Press, Chicago.

This paper is a copy of: Gero, J. S. and Kazakov, V. (1999) Adapting evolutionary computing for exploration in creative designing, in J. S. Gero and M. L. Maher (eds), *Computational Models of Creative Design IV*, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia, pp. 175-186.

