# Adaptive Enlargement of State Spaces in Evolutionary Designing

JOHN S. GERO AND VLADIMIR KAZAKOV
*Key Centre of Design Computing and Cognition*
*Department of Architectural and Design Science*
*University of Sydney NSW 2006 Australia*
*{john, kaz}@arch.usyd.edu.au*

## Abstract

In designing a state space of possible designs is implied by the representation used and the computational processes that operate on that representation. GAs are a means of effectively searching that state space which is defined by the length of the genotype's bit string. Of particular interest in design computing are processes that enlarge that state space to change the set of possible designs. This paper presents one such process based on the generalization of the genetic crossover operation. A crossover operation of genetic algorithms is re-interpreted as a random sampling of interpolating phenotypes, produced by a particular case of phenotypic interpolation. Its generalization is constructed by using a more general version of interpolation and/or by adding extrapolation to interpolation. This generalized crossover has a potential to move the current population outside of the original state space. An adaptive strategy for state space enlargement, which is based on this generalization, is designed. This strategy can be used for computational support of creative designing. An example is given.

**Keywords:** Design Computing; Adaptive Search; Generalized Crossover; State Space Enlargement; Creative Designing.

## 1. Introduction

Designing distinguishes itself from other human activities in a variety of ways. We will use *designing* to indicate the verb and *design* to indicate the noun, rather than use context to disambiguate the meaning. One of the important ways is that the resulting artifact, ie the design, is expected to be different, albeit if only slightly, from previous artefacts. The magnitude and quality of those differences are commonly used to separate designs into two broad categories although the boundary between them is fuzzy and constantly changing. These two categories are variously labeled routine and nonroutine designs or routine and creative designs. It is assumed that the labels can be equally applied to the processes of designing that produced the designs as to the designs themselves. The interest in drawing this distinction lies in the implications it has for articulating computational processes that model or support the activity of designing. The inference is that processes for routine designing are likely to be different from those for nonroutine or creative designing. This has been the focus of considerable research (eg Brown and Chandrasekaran, 1984; Cagan and Agogino, 1987; McLaughlin and Gero, 1987; Gero, 1990; Gero and Maher, 1993; Wolverton and Hayes-Roth, 1995; Gero and Maher, 1998) and has resulted in a widely-held view that a supportable claim can be made for distinguishing processes on this basis. As a consequence there is increasing interest in exploring processes that can claim to be associated with creative designing. Creative designing is generally of interest only during the conceptual stage of designing where all the design variables are not yet known. Thus, this approach could be of interest as the foundation of a possible tool that has as its goal the exploring of possibilities.

The primary difference between these two classes of designing processes lies in the way they manipulate structure. Design *structure* is defined as the basic components and their relationships in a design. Structure is chosen before routine designing commences and remains static and fixed during the designing process. If we define a design state space as all designs that have this structure then we can model routine designing as a (parametric) search in this fixed and static state space. If an evolutionary algorithm is used here then it will shift the population within this space toward areas of high fitness. If a genetic algorithm (GA) is used as the search engine then both the genetic representation and the genotype-phenotype mapping are assumed to be static and fixed. Numerous evolutionary systems that support this model of routine designing have been developed (Bentley, 1999).

Within this framework, a nonroutine designing process is one where the state space of possible designs changes during the process of designing. These changes of state space may include the change of its dimensionality (introduction of new structure variables) as exemplified in Figure 1where another dimension is introduced. As a consequence the state space of possible designs in enlarged. This is a common necessary condition for a designing process to considered creative. We shall adopt this concept in this paper and focus our attention on how we can construct a computational process, using evolutionary systems as our basis, that has such a capability.

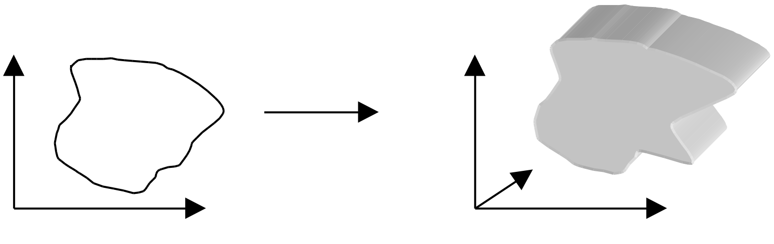

Figure 1. Design state space enlargement.

This change of state space can be carried out not only using structure reasoning but in an inductive fashion, by generalizing on the basis of the set of “good” existing examples. This stage of designing can be modeled as an inductive enlargement of the problem's state space based on the current population. In an evolutionary system, this behavior can be interpreted and modeled as construction of a hyper-surface in a larger state space that is supported by the currently evolved population of designs. Once this hyper-surface is constructed the next generation which belongs to it is evolved. If a GA is used as the computational engine of this process then during this stage a new genetic representation needs to be constructed, jointly with a new genotype-phenotype mapping and, possibly, new crossover and mutation operators. After this state space expansion process, the standard GA machinery can be used to find designs in this enlarged state space defined by this new representation and operators. One can interpret this new stage as a period when new evolutionary operators push the population towards and beyond the boundaries of the original state space. Thus, there is considerable interest in exploring the development of operators that have the capacity to expand the state space of possible designs. There are five classes of operators that appear to have this capacity:

- combination
- mutation
- analogy
- first principles
- emergence.

The research described in this paper is concerned with the first of these, ie, with the generalization of the crossover operation, as a form of combination, in genetic algorithms that is capable of enlarging the state space of the problem in an automatic or semi-automatic fashion. It should done adaptively, using only the information that is contained in (or can be extracted from) the current evolution path of the population.

We assume that some general representation is used to represent phenotypes that is simple (ie, no decoding using external knowledge is necessary to represent the designs). Thus, this representation can be, and as a rule is, redundant since the genotypic and phenotypic representations are homomorphic. We will derive operators in phenotypic space that are equivalent to the standard evolutionary operators if applied to the point from the original state space. Finally, we replace these equivalent operators with more general operators. We can apply these new generalized operators to the population of points from the original state space. This could push the population outside of the original state space. Being outside of the original state space can be interpreted as a relaxation of some of the constraints that determine the original state states. Thus we can interpret this scenario as an evolutionary process of constraint relaxation. However, since which constraints to relax is not known a priori and is only able to be determined post priori, we are not able to directly substitute constraint relaxation for the proposed operators.

Information about the current state space and genetic space is stored in three elements: the genetic representation, the genotype-phenotype mapping and in the evolutionary operators. Once we replace the first two elements with a more general genotype that is homomorphic to its phenotype, we are left with only one element of this triplet that still carries the “memory” of the original state space - the operators.

## 2. GENETIC ALGORITHMS AND CROSSOVER

### 2.1 Standard GA

In a standard GA (Holland, 1975) a search problem is represented using two spaces. The first space, *G*, is the space of ordered character strings, $g = \{e_1, e_2, \ldots, e_n\}$ (which are called genotypes). The components of *G* are drawn from some fixed finite alphabet, *A*. For simplicity, we assume that all genotypes have the same length, *n*. The second space, *P*, is the state space of the solutions (designs). Its elements, *p* (design structures), are phenotypes. Each genotype corresponds to a unique phenotype but not necessarily vice-versa. We assume that the mapping between genotype and phenotype, $M: M \rightarrow P$, is given a priori and is fixed. The fitness function, $F(p)$, which measures the quality of the phenotype, *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.

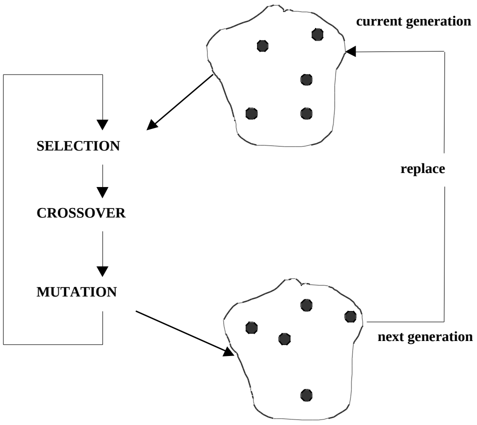

Figure 2. The architecture of the standard genetic algorithm

Our plan is to re-interpret the crossover operator as a random sampling of a particular case of phenotype-phenotype interpolation and then replace it with random sampling of a more general phenotype-phenotype interpolation and/or extrapolation (Gero and Kazakov, 1999). 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.

### 2.2 GA Crossover as random sampling of interpolating points in genotypic space

For this work the most important computational operation that drives GA search is crossover, *C*. In the simplest case it is a binary randomized operation, $C(g_1, g_2) \rightarrow g_c$, which takes two genotypes $g_1$ and $g_2$ as parameters and produces a new genotype $g_c$. It can be viewed as a two-step operation. During the first step 2*($n+1$) new genotypes $g_i(t,g_1,g_2)$ are created by cutting each genotype $g_1$ and $g_2$ into two pieces at point t, and then concatenating the front piece from one genotype with the rear piece from the other. All the possible cut points (the same points in both genotypes) are used here. This yields a fixed set of possible new genotypes $\{g_i(t)\}$, t=0,2*n+1. During the second step of crossover one of these 2*$n+2$ points is chosen randomly

$$g_c(g_1,g_2)=random_{\, t\in[0,2*n+1]}\, g_i(t,g_1,g_2).$$

It can be shown formally that any genotype $g_i$ produced during the first step of the crossover can be written as a result of linear interpolation in *G* space:

$$g_i(t,g_1\, g_2) = f(t)g_1 + (I - f(t))g_2, \qquad t = 0, 1,\ldots.,n;$$
$$g_i(t,g_1,g_2) = f(t\text{-}n\text{-}1)g_2 + (I - f(t\text{-}n\text{-}1))g_1 \quad t = n+1, 1,\ldots.,2*n+1$$

where *I* is a unit n-dimensional matrix with all diagonal elements equal to 1 and all other elements equal to 0, $f(t)$ 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, $f(0) = I$ and $f(1) = O$, where *O* is the n-dimensional zero matrix. In this form, crossover can be interpreted as an operation which first generates a discrete set of interpolating points $g_i(t,g_1,g_2)$ between two basic points $g_1$ and $g_2$ and then randomly samples these points.

Note, that this linear matrix interpolation, which corresponds to the standard one-point crossover, is only one of many possible methods of linear interpolation $g_i^*(t, g_1, g_2)$ between two genotypes in genotypic space:

$$g_i^*(t, g_1, g_2) = c_1(t)g_1 + c_2(n - t)g_2,$$

where operators $c_1(t)$ and $c_2(n - t)$ obey the condition $c_1(0) = I$ and $c_1(n) = O$ and $c_2(0) = I$ and $c_2(n) = O$. The crossover-induced interpolation $g_i(t,g_1,g_2)$ is singled out from many other possible linear interpolations $g_i^*(t, g_1, g_2)$ by the condition that the generalized Hamming distance from the interpolating points $g_i(t,g_1,g_2)$ to the basic points $g_1$ and $g_2$ is minimal. The generalized Hamming distance is defined as the sum of the standard Hamming distances from

$g_i(t,g_1,g_2)$ to the basic points $g_1$ and $g_2$ plus a penalty function (any kind of standard optimization penalty function will do) for each switch from genetic material of one parent to the genetic material of another in the genotype of the offspring. Different versions of crossover can be constructed by choosing different conditions imposed on the interpolation points.

This interpretation of crossover can be illustrated in the space of genotypes *G* as the generation of a discrete set of interpolating points that lie along the "line segment" between the two basic points $g_1$ and $g_2$ and then randomly sampling these points, Figure 3.

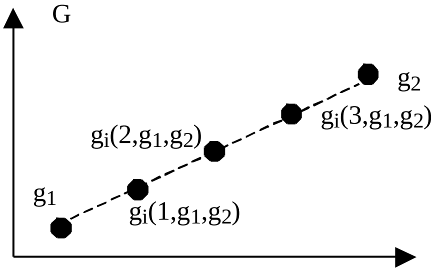

Figure 3. Schematic geometric illustration of crossover as a random sampling of a finite set of discrete points $g_i(t,g_1,g_2)$ that are located along a straight line segment, which connects the basic points $g_1$ and $g_2$ in genetic space G.

## 2.3 Genetic-induced phenotypic crossover

We will restrict ourselves to the case where each genotype isomorphically corresponds to a unique phenotype, the crossover-induced interpolation operation between two genotypes maps onto a interpolation operation between two corresponding phenotypes $p_1=M(g_1)$ and $p_2=M(g_2)$: $p_c(t)=f_i^c(t,p_1,p_2)$, where $f_i^c(0,p_1,p_2)=p_1$ and $f_i^c(1,p_1,p_2)=p_2$. In the general interpolation case $f_i^c$ is a non-linear function (functional) of $p_1$ and $p_2$. This is a discrete interpolation operation since the interpolation parameter *t* takes a discrete set of values 0,1,2,..,2*n+1. Geometrically this means that a non-linear transformation *M* maps a straight line segment in space *G* from Figure 3 onto some non-linear trajectory in space *P*, shown in bold in Figure 4.

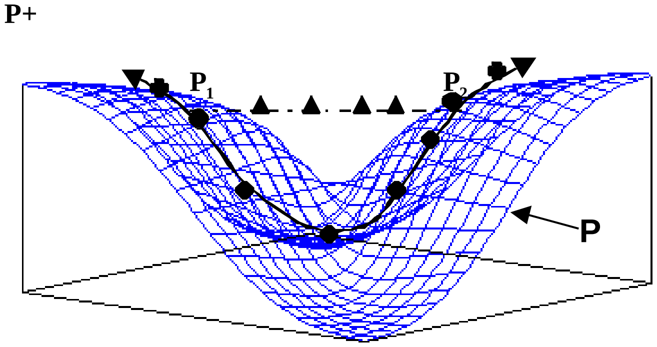

Figure 4. The illustration of the crossover-induced interpolation in *P* and direct interpolation in enlarged space $P^+$. The enlarged space $P^+$ represents the complete 3-d space *P* and the set represents the surface in it. The solid line represents an interpolation between points $p_1$ and $p_2$ in *P*, whilst the dotted line represents an interpolation in $P^+$ between these points. Two external sections of the solid line with arrows represent crossover-induced extrapolation. The black circles represent the possible results of applying crossover to phenotypes $p_1$ and $p_2$ , while the black triangles represent possible results using a more general form of discrete interpolation

## 2.4 Generalization of phenotypic crossover-induced interpolation

Thus, we have suggested that genetic crossover is equivalent to the application of a particular form of discrete phenotype-phenotype interpolation $p_c(t)=f_i^c(t,p_1,p_2)$ that does not require any reference to the genotype. This operation can be generalized in one of the following ways.

(1) By replacing discrete range of the interpolation variable *t* [0,1,..,2*n+1] with a continuous one *t* [0,..,2*n+1]. Geometrically this means that instead of random sampling of a finite discrete set of point $p_c(t)$ (that are shown as black circles in Figure 4) we randomly sample all the points from the continuous interpolating trajectory that connects points $p_1$ and $p_2$ (shown as a bold line in Figure 4).

(2) By replacing the particular case of interpolation $f_i^c(t,p_1,p_2)$ with a more general one $f_i^{c*}(t,p_1,p_2)$, ie, $f_i^{c*}(0,p_1,p_2)=p_1$ and $f_i^{c*}(t,p_1,p_2)=p_2$). Geometrically this means using a discrete set of points that lie along some trajectory that connects points $p_1$ and $p_2$ rather than the discrete set of points that lie along the previous trajectory in Figure 4 with the bold line. For example, this could be the straight line segment shown in Figure 4 with a dotted line. The new discrete interpolating points here are shown as black triangles.

(3) By extending the range of the interpolating variable *t* from [0,…,2*n+1] to [k,k+1,…,0,1,…,n,n+1,…,n+m]. Geometrically this means that we can use additional discrete points that lie along the continuation of the trajectory shown as black crosses on the solid line beyond points $p_1$ and $p_2$ in Figure 4. We can also use their combinations; for example, by replacing crossover-induced interpolation with a more general interpolation and replacing the discrete range of interpolating variables with continuous one, etc.

### 2.4 Limiting enlargement of design state space

Let our phenotypic space (space of designs), *P*, be represented using some representation $R$. We assume that this representation is more general than is required to represent the designs that belong to *P*. That is, a much larger super-space of designs $P^+ : P$ í $P^+$ can be represented using the same representation. This means that the representation *R* implicitly determines the limiting enlargement of the design state space *P*. We shall call such representations *generalizable*. The existence of the generalizable representation is a necessary condition for the algorithm that is proposed in this paper to be productive (that is, for it to have an ability to lead to an enlargement of the design state space). If the representation used by the computational system is generalizable then the system already contains all the knowledge that is necessary to enlarge the state space.

Let us give a simple geometric illustration of what has been described above. The genotypic space *G* maps onto the phenotypic space *P*. The representation which supports *P* is generalizable, therefore another space $P^+$ (which is a superspace with respect to $P$ í $P^+$) 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 *G* space. It maps these segments onto the interpolation trajectories in *P* space, denoted with the bold line in Figure 4, and then samples points from this trajectory. The generalized crossover consists of interpolating trial points directly in $P^+$ using trial points from *P* as the end points of interpolation. They are shown in Figure 4 with the dotted line.

The expectation is that since these end points belong to the established search space *P*, the exploration due to interpolation in the enlarged $P^+$ will not distort the consistency and viability of the space $P$ too much. The issue here is that the results of interpolation in $P^+$ do not lie in *P*, as seen in Figure 4,. 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.

### 2.5 Computational methods for construction of generalized phenotypic interpolation

Let us briefly discuss how to construct this generalized interpolation. Whilst there are methods that are specific for a particular class of problems, in the general case, a "brute force" method has to be used. Here, one a priori defines the class of interpolations, Ω, that can be used (the class of functions it can include, etc). Then, each time a generalized crossover is applied an auxiliary search among different interpolations from this pre-defined class of interpolations is run. Different strategies can be employed here. For example, this can be the maximizing of the fitness of the offspring, then this search is the maximization with respect to $\vartheta \quad \Omega$, and a standard crossover operation $p_c(t)=\text{random}_{t \quad [0,1]}\ f_i^c(t,p_1,p_2)$ is replaced with a generalized crossover

$$C(p_1, p_2) = \max_{\vartheta \quad \Omega} \text{random}_{t \quad [0,1]} F(f_i^* (\vartheta,t, p_1, p_2));$$

or it can be the minimal violation of the set of constraints *s(p)* in some norm ||…||, and it becomes

$$C(p_1, p_2) = \min_{\vartheta \quad \Omega} \text{random}_{t \quad [0,1]} \| s(f_i^* (\vartheta,t, p_1, p_2)) \|.$$

Here $f_i^* (\vartheta,t, p_1, p_2)$ *is the interpolation* $\vartheta$. Again GA can be used as a search engine of this embedded auxiliary search among different interpolations interpolation $\vartheta \quad \Omega$.

## 2.6 Interpolation/extrapolation functions for shapes

As an example, 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 valued functions F(**x**) such that F(**x**) >0 is inside the object, F(**x**)=0 is on its boundary and F(**x**) <0 is outside the object. Here **x** is a 2-d or 3-d vector with a defined feasible bounded region D: **x** ⊂ D. This case is especially well suited to illustrate this approach because here phenotypic interpolation becomes standard function interpolation. Let us consider two phenotypes $F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ and the transformation:

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

$$t\,(1-t)[z_1(\mathbf{x}')\, F_1(\mathbf{x}') + z_2(\mathbf{x}'')\, F_2(\mathbf{x}'') + z_3(\mathbf{x}',\mathbf{x}'')\, F_1(\mathbf{x}')\, F_2(\mathbf{x}'')+\ldots]$$

$$\mathbf{x}'(t,\mathbf{x}): D \rightarrow D,$$

where t is a scalar which changes from 0 to 1, v(**x**) and w(**x**) are non-negative scalar functions (called modulating functions) of **x**, $z_1(\mathbf{x}')$, $z_2(\mathbf{x}'')$, $z_3(\mathbf{x}',\mathbf{x}'')$,… are arbitrary continuous functions, **x**'(t,**x**) is a coordinate transformation which produces a homomorphism from (D, t) to D for any t   [0,1] and depends continuously on t, and t (1-t) z(**x**',**x**'') is a penalty function. Usually the mapping **x**'(t,**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 **x**'(t,**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 $F_c(1,\mathbf{x}'(\mathbf{1},\mathbf{x}))=F_2(\mathbf{x})$) is the positivity of functions v(**x**) and w(**x**) for all **x**   D.

Hence, here the space Ω is determined by the form of transformation $F_c(t,\mathbf{x}')$, the individual points in this space are defined by their coordinates $\vartheta=\{v(\mathbf{x}'),w(\mathbf{x}'), z_1(\mathbf{x}'),z_2(\mathbf{x}''),.., \mathbf{x}'(t,\mathbf{x})\}$ and the goal is to find such positive functions $v^c(\mathbf{x})$ and $w^c(\mathbf{x})$ (we shall call them "crossover modulating functions") and an arbitrary continuous scalar functions $z_1^c(\mathbf{x}')$, $z_2^c(\mathbf{x}'')$, $z_3^c(\mathbf{x}',\mathbf{x}'')$,… jointly with the transformation $\mathbf{x}^{c\prime}(t,\mathbf{x})$. 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 $v^c(\mathbf{x})$, $w^c(\mathbf{x})$ and $z^c(\mathbf{x}',\mathbf{x}'')$). Whether or not this can be done (that is, whether or not $v^c(\mathbf{x})$, $w^c(\mathbf{x})$ and $z^c(\mathbf{x}',\mathbf{x}'')$ exist), or another form of interpolation should to be used, needs to be investigated for each particular case.

Now, that we have re-interpreted GA crossover first as an interpolation operation in genotypic space and then in phenotypic space, we are in a position to produce a generalization of GA crossover which yields an enlargement of the phenotypic space. This enlarged space is subsequently to be 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. Example of Adaptive Enlargement of a Design State Space

As a vehicle to demonstrate the ideas, we consider the problem of designing the cross-section of a beam. This is not normally a conceptual design problem, although as Jorn Utzon, the architect and Ove Arups, the engineers, showed in the design of the support beams for the concourse/forecourt of the *Sydney Opera House* there is no reason why it could not be one (Anon, 1973; Smith, 1984). They designed a beam with a cross-section that varied sinusoidally along the length of the beam. The structure space consists of cross-sections of the pre-defined shape with a fixed area, Figure 5. This shape is initially 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 is set to 500. Scaling the shape guarantees that this area constraint is satisfied. The problem has a two-component fitness function **F** which consists of the moment of inertia **I** (the second moment of area of a geometric shape of the cross-section) and the section modulus **Z** (the moment of inertia divided by the distance from the neutral axis to the extreme edge of the cross-section) (Cowan, 1988). The moment of inertia is the geometric property involved in deformation calculations and the section modulus is the geometric property involved in strength calculations.

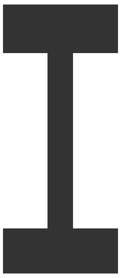

*Figure 5*. The cross-section template which defines the original structure space.

The Pareto-set is used in multicriteria optimization to replace the notion of a single optimal design when there is only one criterion to design for. The definition of membership of the Pareto-set is " a solution for which no other solution exists that is capable of providing a better performance in one criterion and no worse performance in all other criteria" (Radford and Gero, 1988). Whilst a number of GA-based methods have been developed for the generation of the Pareto-set, we use the method implemented by Jun (Jun, 1993).

The standard GA gives the Pareto-set of designs for the problem and is shown in Figure 6. Then the extended GA was initiated, where the crossover is replaced with the minimization of the 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 7. They were generated by a "linear" interpolation procedure with the space-independent modulating functions $v(\mathbf{x})=1$ and $w(\mathbf{x})=1 \quad \mathbf{x}$:

$$F_c(t,\mathbf{x})= t\, v(\mathbf{x})\, F_1(\mathbf{x}) + (1-t)w(\mathbf{x})\, F_2(\mathbf{x})$$

and the two parental shapes were positioned such that the center of inertia of each of them was located at the coordinate origin. The corresponding interpolating sequence is shown in Figure 7.

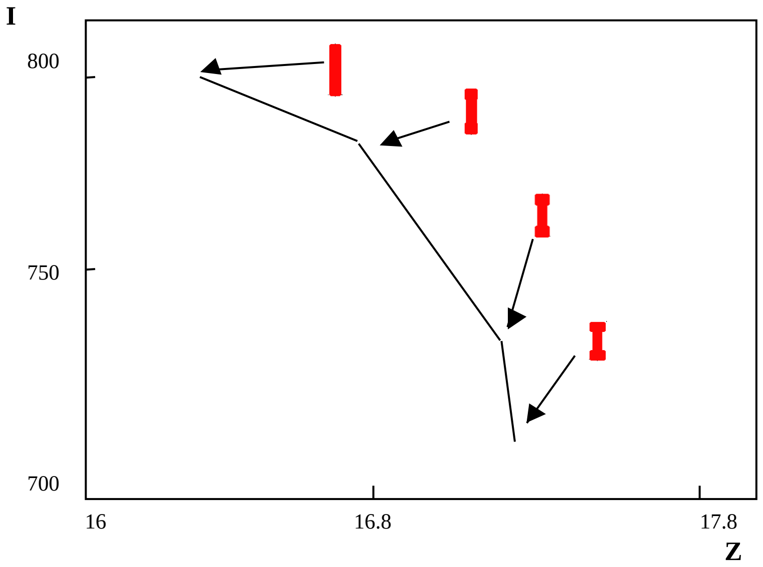

*Figure 6*. The Pareto-set for the initial structure space with the corresponding shapes.

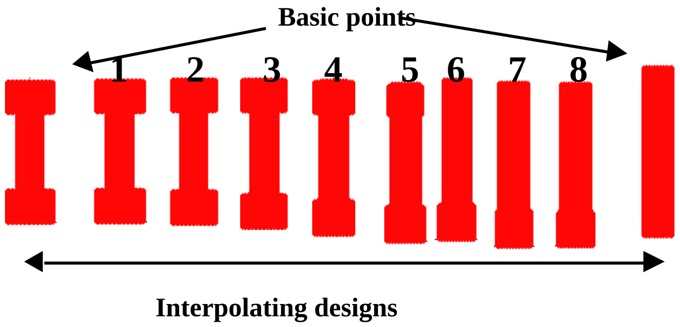

*Figure 7*. The “linear” phenotype-phenotype interpolation

The solutions obtained clearly do not belong to the original structure space. New structure variables have been introduced that are used to change the possible shape of the resulting cross-section. These variables were not known at the outset and are only implied in the interpolating function. The interpolating sequence generated by the “nonlinear” interpolation (with space-variable modulating functions $v(\mathbf{x})=[\sin(\pi|x|/x_{max})]^2+0.01$, $w(\mathbf{x})=[\cos(\pi|x|/x_{max})]^2+0.01$, where $x_{max}$ is the maximal distance from the centre of inertia to the shape boundary averaged over two parental shapes, |..| denotes Euclidian distance) is shown in Figure 8. As one would expect it produces greater deviations from the original structure space and a greater variety of designs. A more “exotic” (seemingly more different from the original space) design generated by this “nonlinear” process, where extrapolation is used instead of interpolation, is shown in Figure 9. This design in Figure 9 clearly demonstrates how the initial, implicit constraints have been relaxed through the use of the interpolating function.

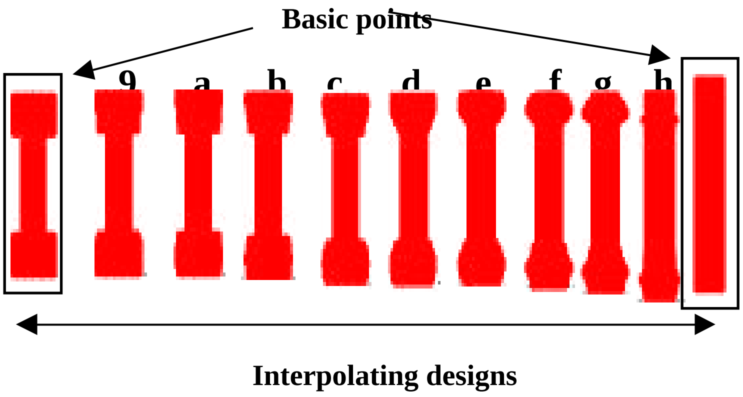

*Figure 8.* The “non-linear” (with non-constant modulating functions) phenotype-phenotype interpolation.

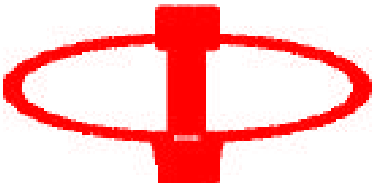

*Figure 9.* “Exotic” or increasingly unexpected design

The designs produced by this interpolation/extrapolation process were not only manifestly different from those produced by standard genetic crossover, they have the capacity to extend the state space of designs in such a way such that better performances than previously become possible. This is exemplified in Figure 10 where a number of these new designs (labelled 4, 5, b, c, d and e) have better Pareto performances than any previously produced designs. The “exotic” design has a very high level of fitness **I**=5301 and **Z**=154 but it is clear that this is an artificial result and that some constraints were broken during its generation which are vital for the viability of the design.

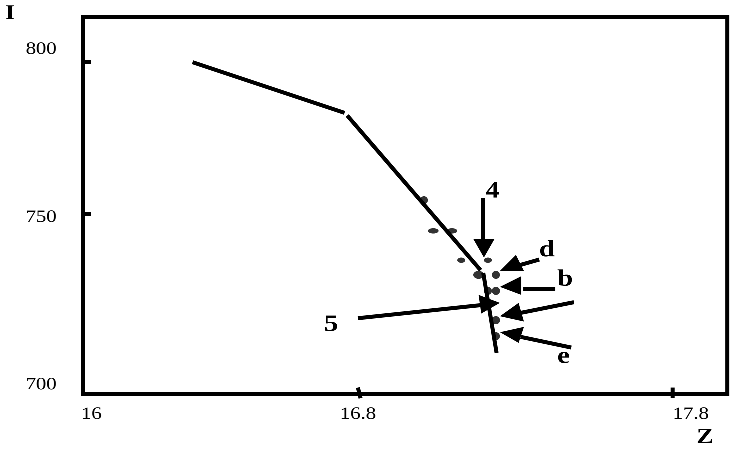

*Figure 10.* The location of the interpolated designs versus Pareto-set of the original design space.

As a further example of how designs produced by a direct phenotype-phenotype interpolation could differ  significantly from the parental designs we show the results of extrapolation using a circular shaft and the previously used I-shape, Figure 11.  The same space-dependent modulating functions have been used here.

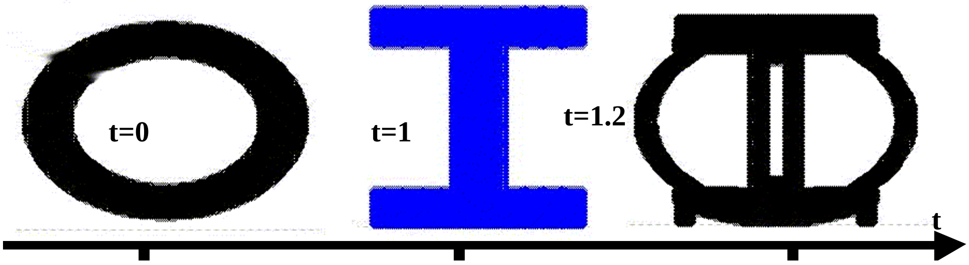

*Figure 11.* A tube and an I cross-section as phenotypes with the resulting interpolated cross-section.

We then tried two scenarios of the extended evolution that included periods where these interpolations occurred. The first scenario was simply a single period during which generalized interpolation/extrapolation was replaced by the most general type of genetic-induced crossover. The modulating functions were constructed as polynomials built using simple trigonometric functions ($\sin(\vartheta|x-\underline{x}|)$, $\cos(\vartheta|x-\underline{x}|)$). The second scenario includes a short period (one to three generations) of the generalized interpolation/ extrapolation followed the stage when phenotypic genetic-induced interpolation is used. The idea was to consider how different is an extreme exploration in the former scenario from the more moderate in the latter. It turned out that in our particular example this difference is not significant. This could be explained by the completeness of the corresponding space of interpolations.

## 4. Discussion

Evolutionary systems have been successfully applied in designing. Genetic algorithms and genetic programming have proven to be robust search methods in models of designing that present themselves as designing by optimization. The standard application of these techniques is based on the assumption that the state space of possible designs is bounded and fixed. As a consequence the ability to produce unexpected designs is limited. However, the global competitiveness of designs often requires that the design finally produced is not simply a parametric variation of the commencing designs, unless it can be demonstrated that the parametric variation has dramatically better performance than its progenitors. This brings us into the realm of nonroutine or creative designing: designing that results in designs that could not be found in the original state space. In this view of creative designing, the state space of possible designs has to change. Such behaviour is evident in creative designers, witness the cross-section of the beams in the concourse/forecourt of the *Sydney Opera House*.

There is, therefore, a need to produce computational processes that have the capacity to expand the state space of possible designs. It is always possible to expand such a state space arbitrarily, but this process fails to ensure that the resulting designs are similar to existing designs (to allow for continuity of marketing, for example) but sufficiently different to offer the possibility of changed or improved performance. It is to this need that the process presented in this paper is addressed.

What we have done is use the existing crossover of standard genetic algorithms and recast it as interpolation in the space of possible designs using the Hamming distance as the distance measure. Then we re-represented the genotype through its isomorphic phenotype and generalized the interpolation. The effect of this was to produce an interpolation trajectory that no longer necessarily lay inside the space of possible designs defined at the outset in the problem's formulation. Thus, the designs produced that lay along that trajectory would be unexpectedly different to those that lay along the standard genetic algorithm crossover trajectory. Once this generalization was made it was possible to add extrapolation to the process to produce designs that were even more different. Extrapolation in terms of standard genetic algorithm crossover as interpolation has no meaning since the set of possible interpolations is fixed and extrapolation simply maps onto one member of this existing set. Which parts of that expanded design 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 designs produced in the example, unexpected designs are produced. These designs are unexpected in the sense that new variables have been introduced into the representation of those designs, variables that were not in the two designs that formed the endpoints of the interpolation. Further, these new variables were not explicitly in the modulating functions either. This adds another approach to the introduction of new variables into the design space. There has 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.

This paper can be seen as a generalization of a number of related interpolation approaches where cross-image interpolation has been used to replace a GA's crossover operator(see for example, Ruprecht, 1994 and Graf and Banzhaf, 1995). Our results are more general because we use a more general type of phenotype-phenotype (cross-object and not cross-image) interpolation that can be reduced to cross-image interpolation in a particular case only.

## Acknowledgments

This work is supported by a grant from the Australian Research Council (Grant No. A89700249). Computing resources have been provided by the Key Center of Design Computing and Cognition.

## References

Aelion, V., Cagan, J. and Powers, G. (1992). Input variable expansion - an algorithmic design generation technique. *Research in Engineering Design* 4, 101-113.

Anonymous (1973). *Sydney Opera House 1973,* Land Printers, Sydney.

Bentley, P. (ed.) (1999). *Evolutionary Design by Computers,* Morgan Kaufmann, San Francisco.

Brown, D. and Chandrasekaran, B. (1984). Expert systems for a class of mechanical activity, *in* J.S. Gero (ed.), *Knowledge Engineering in Computer-Aided Design*, North-Holland, Amsterdam, 259-282.

Cagan, J. and Agogino, A. (1987). Innovative design of mechanical structures from first principles, *AIEDAM* 1(3), 169-189.

Cowan, H.J. (ed.) (1988) *Encyclopedia of Building Technology,* Prentice Hall, Englewood Cliffs, N.J.

Fujimura, K. and Makarov, M. (1997). Homotopic shape deformation, *International Conference on Shape Modeling and Applications*, Aizu-Wakamatsu, 215-225.

Gero, J.S. (1990). Design prototypes: A knowledge representation schema for design, *AI Magazine* 11(4), 26-36.

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.

Gero, J.S. and Maher. M.L. (eds) (1993). *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum, New Jersey.

Gero, J.S. and Maher. M.L. (eds) (1998). *Computational Models of Creative Design IV,* Key Centre of Design Computing and Cognition, University of Sydney, Sydney.

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, 303-316.

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

Jo, J.H. (1993). *A Computational Design Process Model using a Genetic Evolution Approach*, Ph.D. Thesis, Department of Architectural and Design Science, University of Sydney.

McLaughlin, S. and Gero, J.S. (1987). Requirements of a reasoning system to support innovative and creative design activity, *Knowledge-Based Systems* 2(1), 62-71.

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.

Radford, A.D. and Gero, J.S. (1988). *Design by Optimization*, Van Nostrand Reinhold, New York.

Ruprecht, D. (1994). *Geometrische Deformationen als Werkszeug in der graphischen Datenverarbeitung*, Doctoral Dissertation, University of Dortmund, Dortmund, Germany.

Smith, M. P. (1984). *Sydney Opera House*, Collins, Sydney.

Wolverton, M. and Hayes-Roth, B. (1995). Finding analogues for innovative design, *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, 59-84.

This is a copy of the paper Gero, J. S. and Kazakov, V. (2000) Adaptive enlargement of state spaces in evolutionary designing, *AIEDAM* **14**(1): 31-38.