Developing Redundant Binary Representations for Genetic Search

Carlos M. Fonseca
CSI – Centro de Sistemas Inteligentes
Faculdade de Ciências e Tecnologia
Universidade do Algarve
8005-139 Faro, Portugal
cmfonsec@ualg.pt

Marisol B. Correia
CSI – Centro de Sistemas Inteligentes &
Escola Superior de Gestão, Hotelaria e Turismo
Universidade do Algarve
8005-139 Faro, Portugal
mcorreia@ualg.pt

Abstract- This paper considers the development of redundant representations for evolutionary computation. Two new families of redundant binary representations are proposed in the context of a simple mutation-selection evolutionary model. The first is a family of linear encodings in which the connectivity of the search space may be designed directly via a decoding matrix. The second is a family of representations exhibiting various degrees of neutrality, and is constructed using mathematical tools from error-control coding theory. The study of these representations provides additional insight into the properties of redundant encodings, such as synonymity, locality, and connectivity, and into their interrelationships.

1 Introduction

The choice of chromosome representation is known to be a fundamental design issue in genetic search. In particular, the role of redundancy, where there is more than one representation in genotypic space for the same solution in phenotypic space, has been the subject of debate in the last years. Radcliffe [1], for example, noted how redundancy could be detrimental in neural network topology optimisation by genetic algorithms, due to the use of crossover, and developed non-redundant representations for that task. On the other hand, the use of redundant representations has attracted increasing attention in recent years [2–5] for its potential to create alternative paths for evolution and, in this way, improve the quality of the search. Unfortunately, the influence of redundant representations on the performance of evolutionary algorithms is not yet well understood, and practical evidence [4] of improved search performance afforded by redundant representations has not passed unquestioned [6].

Although the main motivation for the development of redundant representations in evolutionary computation has been the desire to obtain increased performance, the redundant representations proposed so far tend to favour large amounts of redundancy [4, 7–10], and to be based on encoding mechanisms which do not facilitate their analysis. As a result, experimental comparisons of the resulting algorithms with an algorithm based on a non-redundant representation may be affected by large differences in the size of the search spaces and the corresponding interaction with evolutionary algorithm settings such as mutation and recombination rates.

Rothlauf and Goldberg [5] identify a number of properties of redundant representations, which are known to influence their quality:

Uniformity A representation is uniformly redundant if all phenotypes are represented by the same number of genotypes;

Synonymity A representation is synonymously redundant if the genotypes which represent the same phenotype are similar to each other, i.e., they are close to each other in genotypic space;

Locality A representation has high locality if neighbouring genotypes correspond to neighbouring phenotypes;

Connectivity A representation has high connectivity if the number of phenotypes which are accessible from a given phenotype by single bit mutations is high.

However, it is fair to say that the importance and the interrelationships between these properties of redundant representations is not yet well understood. For example, Rothlauf and Goldberg [5] assert that, when using synonymously redundant representations, the connectivity between the phenotypes is not increased, whereas in this paper we provide practical evidence to the contrary.

In this study, the development of families of redundant representations with varying degrees of synonymity, locality, and connectivity is approached incrementally, from the non-redundant representation upwards. Using simple, but powerful, mathematical tools, the non-redundant representation is progressively modified so as to meet specific requirements, stated in advance. All developments are carried out in the context of a simple mutation-selection evolutionary model (the quasi-species model [11]). This allows the effect of longer chromosomes in the performance of the algorithm to be taken into account, since the interaction between chromosome length, mutation and selection is rather well understood in this model.

In the next section, the evolutionary model adopted is described. Then, a family of linear redundant encodings is introduced, which allows redundancy to be explicitly harnessed to bias the effects of *independent* mutation towards certain phenotypic traits, and/or to define the set of phenotypes reachable from each individual phenotype. Finally, a novel family of redundant, neutral representations is developed using mathematical tools from error-control coding theory. These representations are shown to be capable of achieving large levels of connectivity even with relatively

small redundancy, as well as displaying considerably high synonymity and varying degrees of locality. The paper concludes with a discussion of the results and some directions for further work.

2 Evolutionary Model

Consider a simple evolutionary model involving a population of N individuals, a constant selective pressure σ (as imposed by tournament and rank-based selection, for example), independent bit mutation, and generational replacement. Individuals in the population are represented as binary vectors $\mathbf{v}$ of length ℓ (the genotypes), and may exhibit any number of k phenotypic characteristics, or traits. Representing each trait as a binary variable, phenotypes may also be seen as binary vectors, $\mathbf{u}$, but of length k.

Under such a model, the probability of individuals not undergoing mutation should be such that at least one exact copy of the best individual in the population can expect to survive beyond its parent's death. This setting, suggested by Eigen and Schuster's work (see [12]), maximises exploration of the search space by the population while guaranteeing the exploitation of the best individual. In fact, higher mutation rates would quickly make the search degenerate into a random walk, because selection would no longer be able to recover from the genetic errors introduced by mutation. This abrupt change in the qualitative behaviour of the evolutionary process as the mutation rate increases is known as *error catastrophe*. Mutation rates lower than the corresponding *error threshold* are more likely to preserve the best individual in each generation, but also make the search less explorative.

For a sufficiently large population, this balance between selection and mutation is achieved by setting the probability of the best individual surviving mutation to the inverse of the selective pressure, i.e.:

$$(1-m)^{\ell} = 1/\sigma$$

where m represents the bit mutation rate. Solving for m,

$$\begin{aligned} m &= 1-\sigma^{-1/\ell} && (1)\\ &\simeq \frac{\log\sigma}{\ell} && (2)\end{aligned}$$

where (2) is the expression derived by Eigen and Schuster in the context of molecular evolution. For smaller populations, where the effect of genetic drift cannot be neglected, this critical mutation rate is reduced, but the general principle is the same [13].

3 Developing Redundant Representations

In the evolutionary model described in the previous section, both genotypes and phenotypes consist of binary vectors. Therefore, the simplest, non-redundant, mapping from genotypes to phenotypes is the identity map. Considering $\mathbf{u}$ (respectively, $\mathbf{v}$) as an element of the vector space of all k-tuples (respectively, ℓ-tuples) over the field $\mathrm{GF}(2)$, this can be written in matrix form as:

$$\mathbf{u} = \mathbf{I}_k \cdot \mathbf{v}$$

where $\mathbf{u}$ and $\mathbf{v}$ are column vectors, $\mathbf{I}_k$ is the identity matrix of order k, and $\ell = k$. $\mathrm{GF}(2)$ is the Galois Field $(\{0,1\},+,\cdot)$, where addition is modulo-2 (logic XOR) and multiplication is the logical AND operation [14].

In evolutionary computation, the use of more complex representations is only justified when the fitness landscape is sufficiently complex for evolution based on a direct genotype-phenotype mapping to be unsuccessful. Taking the identity map as a reference, increasingly elaborate redundant representations will now be proposed, with a view to highlighting the impact of each type of redundancy in the overall evolutionary process.

3.1 Non-coding Genes

The simplest form of genotypic redundancy consists of adding non-coding, or inactive, genes to the chromosome. In matrix form, one may write:

$$\mathbf{u} = [\mathbf{I}_k \, \mathbf{0}_{k\times(\ell-k)}] \cdot \mathbf{v}$$

where $\mathbf{u} \in \{0,1\}^k$, $\mathbf{v} \in \{0,1\}^\ell$, and $\ell > k$. Matrix $\mathbf{0}_{k\times(\ell-k)}$ is the $k \times (\ell-k)$ zero matrix.

Non-coding genes will be subjected to mutation just like coding, or active, genes. Such mutations do not change the phenotype in any way, and are often called "neutral mutations" for that reason. However, unless the number of redundant bits $\ell - k$ is explicitly taken into account when setting the mutation rate, redundancy actually leads to the active part of the chromosome being mutated less often, and to slower rates of evolution.

Non-coding genes would seem to have no practical relevance in the context of the evolutionary model adopted. The question of whether a redundant representation may be constructed which preserves evolutionary behaviour under this model will be answered next.

3.2 Implementing Polygeny

In a redundant genotype-phenotype mapping without non-coding genes, at least one phenotypic trait must be determined by the interaction of two or more genes, an effect which is known as polygeny [15]. For simplicity, consider the mapping where each phenotypic trait is determined by modulo-2 addition of two bits. One may write:

$$\mathbf{u} = [\mathbf{I}_k \, \mathbf{I}_k] \cdot \mathbf{v}$$

with $\ell = 2k$. Then, for a large population, the prescribed bit mutation rate will be:

$$\begin{aligned} m &= 1-\sigma^{-1/(2k)} \\ &\simeq \frac{\log\sigma}{2k}\end{aligned}$$

i.e., half of that for the non-redundant case.

What effect can such a representation have on the evolutionary process? Note that each phenotypic trait will only

change as a consequence of the mutation of one of the genes which determine it. The probability of a given trait changing as a consequence of mutation is thus:

$$\begin{aligned} m^* &= 2\,(1-\sigma^{-1/(2k)})\,\sigma^{-1/(2k)} \\ &\simeq \frac{\log\sigma}{k}\left(1-\frac{\log\sigma}{2k}\right) \\ &\simeq \frac{\log\sigma}{k} \end{aligned}$$

where the approximations are valid for sufficiently large k. When compared to the non-redundant case, such a representation will decrease the probability of mutation of each trait, but only asymptotically as k increases. This is because two bit mutations will leave the corresponding trait unchanged, and because such events are very unlikely for large k. For $\sigma = 2$ and $k = 40$, for example, the effect of the extra bits on the probability of each trait changing due to mutation will be less than 1%.

This representation is more of theoretical than of practical value. In fact, by demonstrating how redundancy may be added to a basic representation without significantly affecting evolutionary behaviour, light is shed onto how redundancy may be used to change that behaviour in advantageous ways.

3.3 Implementing Pleiotropy

Pleiotropy is the effect that a single gene may simultaneously affect several phenotypic traits [15]. The representation developed in the previous section can be extended very easily to implement pleiotropic effects without significantly affecting the probability of individuals surviving mutation. In fact, any $k \times \ell$ binary matrix $\mathbf{G}$ with non-zero columns and $\ell > k$ defines a redundant, non-neutral genotype-phenotype map, such that

$$\mathbf{u} = \mathbf{G} \cdot \mathbf{v} \tag{3}$$

Such a linear transformation is not unlike the affine transformations used by Liepins and Vose [16] to transform deceptive functions into easy ones. However, the issue of redundancy was not addressed in that work.

Despite its simplicity, the transformation in (3) has the virtue of offering considerable control over a representation to be designed. In fact, the columns of $\mathbf{G}$ consist of, and thus explicitly determine, the phenotypes which are reachable from the all-zero phenotype through single gene mutations. To understand this, consider $\mathbf{G} = [\mathbf{g}_{\ell-1}, \ldots, \mathbf{g}_0]$, where each $\mathbf{g}_i$ denotes a column of $\mathbf{G}$ and $0 \leq i < \ell$. The result of a single-bit mutation at position i of $\mathbf{v}$ may be written as $\mathbf{v} + \mathbf{e}_i$, where $\mathbf{e}_i$ is a vector of length ℓ with a single non-zero bit at position i. It follows that:

$$\mathbf{G} \cdot (\mathbf{v} + \mathbf{e}_i) = \mathbf{G} \cdot \mathbf{v} + \mathbf{G} \cdot \mathbf{e}_i = \mathbf{u} + \mathbf{g}_i$$

When $\mathbf{u}$ is zero, $\mathbf{g}_i$ is the phenotype obtained through mutation of gene i. Furthermore, the effect of mutation on an arbitrary phenotype $\mathbf{u}$ does not depend on the original genotype $\mathbf{v}$, but only on the bit mutated (and the corresponding column of $\mathbf{G}$).

In addition, the rows of $\mathbf{G}$ determine how likely each trait is to be changed through a single gene mutation in comparison to the other traits. The more ones a given row of $\mathbf{G}$ contains, the more likely is the corresponding bit of the phenotype $\mathbf{u}$ to change due to a single-bit mutation.

By selecting ℓ and $\mathbf{G}$ appropriately, it is possible to define both the connectivity and the locality of the search space. In particular, connectivity may be made to grow linearly with the chromosome length, ℓ, simply by making all columns of $\mathbf{G}$ distinct.

4 Neutrality

Although the linear encodings proposed in the previous section make it possible to specify the set of phenotypes reachable from a given phenotype through single-bit mutations, all genotypic representations of the same phenotype are equivalent as far as the search is concerned, because they all reach exactly the same set of phenotypes. On the other hand, Kimura's neutral theory of evolution [17] considers that a large fraction of mutations is neutral, i.e., leads to the same phenotype, and that the accumulation of neutral mutations eventually provides new paths for evolution.

Consider the problem of formulating redundant representations with the following characteristics:

1. All phenotypes admit the same number of genotypic representations (uniformity);
2. Individual traits may be determined by the interaction of multiple genes (polygeny);
3. Individual genes may determine any number of phenotypic traits (pleiotropy);
4. Different genotypes representing the same phenotype should reach different sets of neighbouring phenotypes through single gene mutations;
5. Different representations for the same phenotype must be connected by a path composed exclusively of neutral single-gene mutations (neutrality).

Clearly, goals 1.–3. are met by the linear encodings discussed earlier. In contrast, goals 4. and 5. suggest that a radically different approach to encoding formulation is required. A suitable strategy may be summarised as follows:

1. Split the redundant genotypic space into $2^{\ell-k}$ classes of equal cardinality 2^k each, such that single gene mutations allow moving from one class to another;
2. Map the vectors in each class to phenotypic space in such a way that different representations of the same phenotype may reach different sets of phenotypes, each including at least another representation of the same phenotype.

This idea is illustrated in Figures 1 and 2. In Figure 1, an 8×8 grid is divided into 4 different interspersed subspaces of 16 elements each, represented by the symbols ○, ●, □ and ■. Note that moving either vertically or horizontally

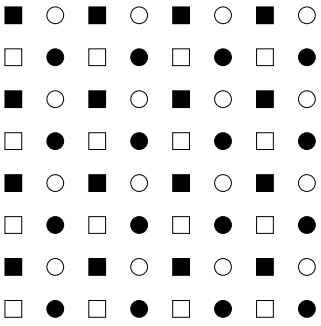

Figure 1: Redundant genotypic space divided into 4 interspersed subspaces.

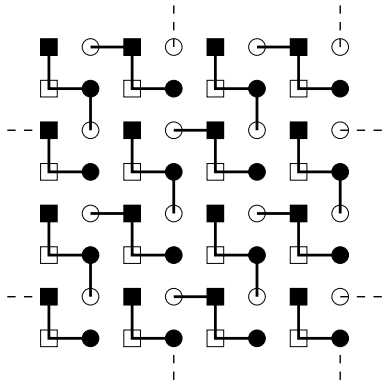

Figure 2: Genotypes mapping to the same phenotypes.

on the grid always leads to a subspace other than the original one, and that the grid is assumed to wrap around the edges. In Figure 2, phenotypic values are assigned to each subspace in such a way that each genotype which encodes for a given phenotype is within reach of another such genotype. A solid line is used to highlight the neutral networks thus formed, with dashed lines indicating continuation of the network at the edges. It can be clearly seen that each phenotype may now reach 6 other phenotypes instead of 4, as would be the case on a non-redundant 4×4 grid, and that different genotypes in the same neutral network have different neighbouring phenotypes, as desired.

Going back to the binary case, by defining classes in such a way that the minimum Hamming distance between genotypes in the same class is at least 2, it can be guaranteed that single gene mutations produce genotypes in classes other than that of the original genotype. This is a common problem in error-control coding design, and results from this area [14] are readily applicable here.

4.1 Linear Block Error-control Codes

Linear block codes may be defined as follows [14]:
Definition 1 *A binary block code of length ℓ and with 2^k code words is called a linear (ℓ,k) code if its 2^k code vectors form a k-dimensional subspace of the vector space of all the ℓ-tuples over the field $GF(2)$.*

Linear block codes define a mapping between a certain class of binary code vectors of length ℓ and binary message vectors of length k. Although linear codes can also be described based on the matrix notation used in the previous section, an alternative, polynomial representation is especially useful in describing, for example, the important subclass of cyclic linear codes. Unlike the matrix notation, this polynomial representation is unrelated to that used by Liepins and Vose [16].

4.2 Polynomial Notation

Code vectors $\mathbf{v}$ can also be represented by polynomials where each component of a given vector is treated as a coefficient, as follows:

$$\mathbf{v}(X) = v_{\ell-1}X^{\ell-1} + v_{\ell-2}X^{\ell-2} + \cdots + v_1X + v_0.$$

Cyclic codes, as well as some other, more general code subclasses, can be constructed by selecting a suitable generator polynomial $\mathbf{g}(X)$ of degree $\ell - k$, and computing its multiples for every possible value of the message polynomial $\mathbf{u}(X)$, i.e.:

$$\mathbf{v}(X) = \mathbf{g}(X)\mathbf{u}(X) \tag{4}$$

Provided that $\mathbf{g}(X)$ is a primitive (or even merely irreducible) polynomial,[1] it is easy to show that $\mathbf{v}(X)$ cannot be of type X^i (otherwise $\mathbf{g}(X)$ itself would have to be of the same type and would not be primitive or even irreducible). Due to the linearity of the code, neither can the difference between two code polynomials have that form, which implies that the minimum distance between different code polynomials must be at least two.

In an evolutionary algorithm context, expression (4) provides a method of mapping arbitrary phenotypes onto a linear class C of code polynomials of degree up to $\ell - 1$. Furthermore, it suggests that another $2^{\ell-k} - 1$ classes of code polynomials may be defined as:

$$\mathbf{v}(X) = \mathbf{g}(X)\mathbf{u}(X) + \mathbf{r}_j(X)$$

for every non-zero polynomial $\mathbf{r}_j(X)$ of degree up to $\ell - k - 1$. Clearly, these classes are no longer linear (the sum of two polynomials in a class C_j is not in C_j), but since they can be obtained from the linear class C through the addition of a constant polynomial, they are called the *cosets* of C. Furthermore, changing a single coefficient in a polynomial $\mathbf{v}(X) \in C$ will have the effect of producing a member of one of the cosets C_j. This is exactly the property which is explored in error-control applications, but here it will provide a suitable mechanism for making the sets of reachable phenotypes vary for different genotypic representations of the same phenotype.

Equation (4) shows how code polynomials can be obtained by multiplying the corresponding message polynomials by $\mathbf{g}(X)$. A more popular approach consists of generating code polynomials by multiplying message polynomials by $X^{\ell-k}$ and dividing the results by $\mathbf{g}(X)$, which allows the code to be generated in *systematic* form. A code

[1] A polynomial is irreducible if it cannot be factored. A polynomial of order n is primitive if it is irreducible and the smallest degree e of a polynomial $X^e + 1$ which it divides is $e = 2^n - 1$.

is in systematic form if message vectors can be obtained by discarding given $\ell - k$ components from the corresponding code vectors.

Formally,

$$\begin{aligned} X^{\ell-k}\mathbf{u}(X) &= \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X) \qquad (5) \\ &\Updownarrow \\ X^{\ell-k}\mathbf{u}(X) + \mathbf{b}(X) &= \mathbf{a}(X)\mathbf{g}(X) = \mathbf{v}(X), \end{aligned}$$

where $\mathbf{b}(X)$ is a polynomial of degree $\ell - k - 1$ or less. Thus, $\mathbf{v}(X)$ is in systematic form, since the message can be recovered by discarding the terms of degree lower than $\ell - k$. As before, cosets C_j may be defined by adding to each polynomial in C a constant polynomial not in C.

4.3 Neutral Representations

A neutral genotype-phenotype encoding with the desired properties may now be defined as follows:

1. If $\mathbf{v}(X)$ is in $C_0 = C$, then $\mathbf{u} = [\mathbf{I}_k \, \mathbf{0}_{k\times(\ell-k)}] \cdot \mathbf{v}$
2. If $\mathbf{v}(X)$ is in C_j, for some $0 < j < 2^{\ell-k}$, then $\mathbf{u} = [\mathbf{I}_k \, \mathbf{0}_{k\times(\ell-k)}] \cdot (\mathbf{v} + \mathbf{z}_j)$

where $\mathbf{z}_j(X) \in C_j$. Vectors $\mathbf{z}_j$ are no more than the genotypic representations of the all-zero phenotype in each coset C_j, whereas the corresponding representation in C is the all-zero genotype (denote it $\mathbf{z}_0$). Determining the class C_j to which each $\mathbf{v}(X)$ belongs may be accomplished by determining the remainder $\mathbf{r}(X)$ of its polynomial division by $\mathbf{g}(X)$ and, for example, taking j as the integer whose binary representation is $\mathbf{r}$. Since polynomial division can be implemented efficiently, so can genotype decoding.

Since, under this encoding, single gene mutations cause genotypes to move to a different class, and since the mapping between each class C_j and phenotypic space may be changed through the corresponding $\mathbf{z}_j$, different representations $\mathbf{v}$ of the same phenotype $\mathbf{u}$ may now reach different sets of phenotypes through single gene mutations. Finally, by selecting the various $\mathbf{z}_j$, $0 \le j < 2^{\ell-k}$, so that they are reachable from each other, a neutral network may be defined for the all-zero phenotype. Similar neutral networks will arise for every other phenotype.

4.4 Enumerating All Neutral Encodings

The encoding proposed in the previous subsection is actually a family of encodings. In fact, for each pair (ℓ, k) and primitive generator polynomial $\mathbf{g}(X)$ of degree $\ell - k$, many different neutral representations may be constructed by choosing appropriate $\mathbf{z}_j$. In particular, a redundant representation involving $\ell - k$ non-coding genes, similar to those described in section 3.1, may be defined as a special case, by setting each $\mathbf{z}_j$ to the binary representation of j.

A variant of the closure algorithm [18] was used to systematically generate all possible representations under these conditions. Starting from the representation described in the last paragraph, the various $\mathbf{z}_j$ were successively varied until no more representations with the desired properties could be generated.

Exhaustive enumeration was only feasible for reasonably small ℓ (up to 8) and low redundancy ($\ell - k \le 3$), but this exercise did allow the properties of the resulting representations to be inspected, and insight to be gained into the potential of such representations for genetic search.

5 Results and Discussion

Families of neutral representations were exhaustively enumerated for $0 < k \le 8$ and primitive polynomials $\mathbf{g}(X)$ of degrees $0 < \ell - k \le 4$, where practicable. Their sizes are presented in Table 1 as a function of k and of the generator polynomial used. It can be seen that the number of possible representations increases rapidly with k and very rapidly with the degree $\ell - k$.

Subsequently, for every representation produced, the set of reachable phenotypes was determined. Table 2 shows the maximum connectivity which can be achieved by representations in each family, complemented with some lower bounds for larger codes, which were obtained through a combination of the closure algorithm and hill-climbing (in brackets).

It is interesting to note that the maximum connectivity in each representation family appears to increase exponentially with the number of redundant bits. This surprising result indicates that very large connectivities may be achieved even with comparatively little redundancy, and suggests that the current emphasis on highly-redundant representations should probably be reconsidered.

In Figures 3 and 4, scatter plots of connectivity versus average and maximum distance to reachable phenotypes are presented for the (11,8) family of neutral representations. These plots show that relatively strong locality can be maintained by certain representations, despite large connectivity.

Similarly, scatter plots of connectivity versus average and maximum distance between genotypes representing the same phenotype are given in Figures 5 and 6. Strikingly, degrees of connectivity as high as 23 (in comparison to 8 for the non-redundant representation) are exhibited even by highly synonymous representations (at least as highly synonymous as the base representation including 3 non-coding bits), in clear contrast with Rothlauf and Goldberg's assertion [5] that synonymity does not increase connectivity.

Finally, the shape of the actual neutral networks induced by these representations may be visualised through a technique known as multi-dimensional scaling (MDS) [19], by approximating in Euclidean space the Hamming distances between all pairs of genotypes which represent the same phenotype. In Figure 7, MDS graphical representations of the neutral networks induced by two different $(11,8)$ representations are given: the first one corresponds to a highly neutral and highly synonymous representation which exhibits, nevertheless, a degree of connectivity of 23; the second one corresponds to a representation with degree of connectivity 30.

Table 1: Number of neutral representations

	1	2	3	4	5	6	7	8	k
$X+1$	2	3	4	5	6	7	8	9	
X^2+X+1	4	9	18	32	50	75	108	147	
X^3+X+1	19	266	1828	8128	25100	66750	158784	342701	
X^4+X+1	1489								
$\mathbf{g}(X)$									

Table 2: Maximum connectivity

	1	2	3	4	5	6	7	8	k
$X+1$	1	2	3	4	5	6	7	8	
X^2+X+1	1	3	5	7	9	11	13	15	
X^3+X+1	1	3	7	14	18	22	26	30	
X^4+X+1	1	(3)	(7)	(15)	(30)	(43)	(55)	(63)	
X^5+X^2+1	1	(3)	(7)	(15)	(31)	(59)	(98)	(125)	
$\mathbf{g}(X)$									

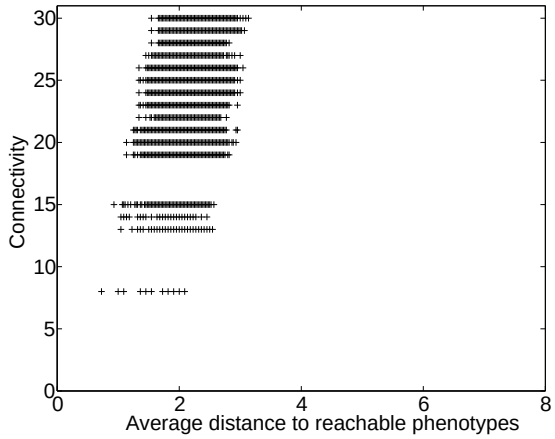

Figure 3: Connectivity versus locality of (11,8) neutral representations

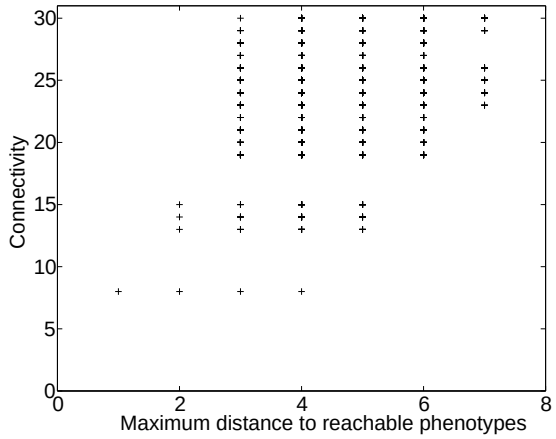

Figure 4: Connectivity versus locality of (11,8) neutral representations

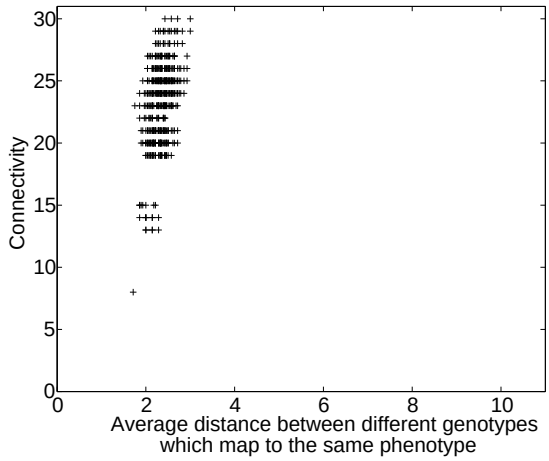

Figure 5: Connectivity versus synonymity of (11,8) neutral representations

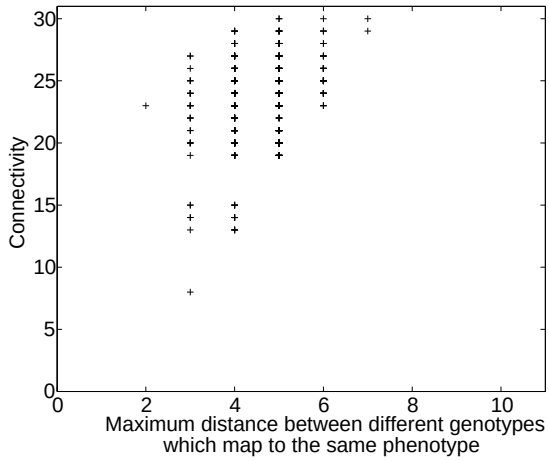

Figure 6: Connectivity versus synonymity of (11,8) neutral representations

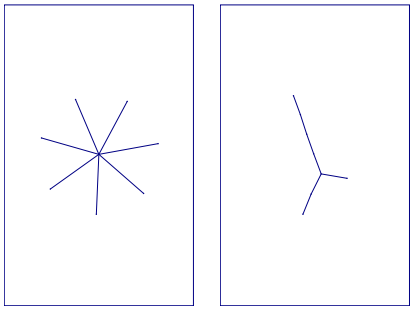

Figure 7: MDS representation of two neutral networks

6 Concluding Remarks

In this paper, an incremental approach to the development of new redundant binary representations was adopted. After specifying a suitable evolutionary model based on selection and independent mutation only, redundancy was shown not to be necessarily detrimental in that context. This was achieved by constructing a purely polygenic, redundant representation leading to a performance which is asymptotically equivalent to that of the corresponding non-redundant genotype-to-phenotype mapping. This result was then extended to implement dependencies between genes, or pleiotropy. In this setting, the connectivity of the search space can be explicitly designed and made to increase proportionally to the chromosome length, but the representation remains non-neutral.

A radically different approach based on results from error-control code theory was adopted to implement neutrality. The result was the definition of a rich family of redundant binary representations implementing various levels of neutrality, connectivity, and locality. In particular, the maximum connectivity achievable with these codes appears to increase exponentially in the number of redundant bits! Thus, even small levels of redundancy may have a large impact in search behaviour. On the other hand, this apparent gain in genotype length comes at the immediate expense of having to store (or compute) all the representations of the all-zero phenotype, the number of which also grows exponentially with the number of redundant bits. Whether at least some of these encodings may be represented more efficiently is the subject of future work.

Comparatively speaking, the linear encodings proposed in the first part of this study have the clearly advantage of allowing search neighbourhoods to be designed directly, whereas the link between a neutral network and its neighbours is much less clear at this stage. In contrast, the neutral representations developed in the second part clearly open a new avenue for research into the relative merits of neutrality, synonymity, locality and connectivity in genetic search, especially with respect to a non-neutral, but comparable, linear encoding. The study of both encoding families in the presence of recombination is likely to raise new and interesting issues, as well.

Unfortunately, it is still not clear how to select a specific representation from the classes proposed here to obtain good results on a given optimisation problem. However, the large number and variety of neutral redundant encodings which can now be generated should be useful in improving that understanding.

Bibliography

[1] N. J. Radcliffe, “Genetic set recombination and its application to neural network topology optimisation,” *Neural Computing and Applications*, vol. 1, no. 1, pp. 67–90, 1993.

[2] M. A. Huynen, P. F. Stadler, and W. Fontana, “Smoothness within ruggedness: The role of neu-

trality in adaptation," *Proceedings of the National Academy of Sciences of the USA*, vol. 93, pp. 397–401, 1996.

[3] T. Smith, P. Husbands, and M. O'Shea, "Neutral networks and evolvability with complex genotype-phenotype mapping," in *Proceedings of the European Conference on Artificial Life (ECAL 2001)*, pp. 272–281, 2001.

[4] M. Ebner, M. Shackleton, and R. Shipman, "How neutral networks influence evolvability," *Complexity*, vol. 7, no. 2, pp. 19–33, 2001.

[5] F. Rothlauf and D. E. Goldberg, "Redundant representations in evolutionary computation," *Evolutionary Computation*, vol. 11, no. 4, pp. 381–415, 2003.

[6] J. D. Knowles and R. A. Watson, "On the utility of redundant encodings in mutation-based evolutionary search," in *Proceedings of Parallel Problem Solving From Nature – PPSN VII, Seventh International Conference* (J.-J. Merelo Guervós, P. Adamidis, H.-G. Beyer, J.-L. Fernández-Villacañas, and H.-P. Schwefel, eds.), vol. 2439 of *Lecture Notes in Computer Science*, pp. 88–98, Springer, 2002.

[7] M. Shackleton, R. Shipman, and M. Ebner, "An investigation of redundant genotype-phenotype mappings and their role in evolutionary search," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 493–500, IEEE, 2000.

[8] R. Shipman, "Genetic redundancy: Desirable or problematic for evolutionary adaptation?," in *Proceedings of the 4th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA)* (A. Dobnikar, N. C. Steele, D. W. Pearson, and R. F. Albrecht, eds.), pp. 337–344, Springer, 1999.

[9] R. Shipman, M. Shackleton, M. Ebner, and R. Watson, "Neutral search spaces for artificial evolution: A lesson from life," in *Artificial Life VII: Proceedings of the Seventh International Conference* (M. A. Bedau, J. S. McCaskill, N. H. Packard, and S. Rasmussen, eds.), pp. 162–169, MIT Press, 2000.

[10] R. Shipman, "Issues in designing a neutral genotype-phenotype mapping," in *Proceedings of the London Communications Symposium*, 2002.

[11] C. Reidys, C. V. Forst, and P. Schuster, "Replication and mutation on neutral networks," *Bulletin of Mathematical Biology*, vol. 63, no. 1, pp. 57–94, 2001.

[12] I. Harvey, "Evolutionary robotics and SAGA: The case for hill crawling and tournament selection," in *Artificial Life III* (C. G. Langton, ed.), pp. 299–326, Addison Wesley, 1994.

[13] M. Nowak and P. Schuster, "Error thresholds of replication in finite populations mutation frequencies and the onset of Muller's ratchet," *Journal of Theoretical Biology*, vol. 137, no. 4, pp. 375–395, 1989.

[14] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice Hall, 1983.

[15] D. Fogel, "Principles of evolutionary processes," in *Handbook of Evolutionary Computation* (T. Bäck, D. B. Fogel, and Z. Michalewicz, eds.), sec. C2.1, IOP Publishing and Oxford University Press, 1997.

[16] G. E. Liepins and M. D. Vose, "Representational issues in genetic optimization," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 2, pp. 101–115, Apr. 1990.

[17] M. Kimura, "Evolutionary rate at the molecular level," *Nature*, vol. 217, pp. 624–626, 1968.

[18] H. Lewis and C. Papadimitriou, *Elements of the Theory of Computation*. Prentice-Hall, 2nd ed., 1998.

[19] J. de Leeuw, "Multidimensional scaling," Statistics Preprint 274, UCLA, 2000.