Theoretical Computer Science 641 (2016) 53–74

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

New insights on neutral binary representations for evolutionary optimization

Marisol B. Correia [a,b,*]

[a] ESGHT, University of the Algarve, Faro, Portugal
[b] CEG-IST, Universidade de Lisboa, Lisbon, Portugal

ARTICLE INFO

Article history:
Received 21 May 2015
Received in revised form 7 March 2016
Accepted 21 May 2016
Available online 26 May 2016
Communicated by I. Petre

Keywords:
Redundant binary representations
Neutrality
Error control codes
Properties of representations
Evolutionary algorithms
NK fitness landscapes

ABSTRACT

This paper studies a family of redundant binary representations $NN_g(\ell, k)$, which are based on the mathematical formulation of error control codes, in particular, on linear block codes, which are used to add redundancy and neutrality to the representations. The analysis of the properties of uniformity, connectivity, synonymity, locality and topology of the $NN_g(\ell, k)$ representations is presented, as well as the way an *(1+1)-ES* can be modeled using Markov chains and applied to *NK* fitness landscapes with adjacent neighborhood.

The results show that it is possible to design synonymously redundant representations that allow an increase of the connectivity between phenotypes. For easy problems, synonymously $NN_g(\ell, k)$ representations, with high locality, and where it is not necessary to present high values of connectivity are the most suitable for an efficient evolutionary search. On the contrary, for difficult problems, $NN_g(\ell, k)$ representations with low locality, which present connectivity between intermediate to high and with intermediate values of synonymity are the best ones. These results allow to conclude that $NN_g(\ell, k)$ representations with better performance in *NK* fitness landscapes with adjacent neighborhood do not exhibit extreme values of any of the properties commonly considered in the literature of evolutionary computation. This conclusion is contrary to what one would expect when taking into account the literature recommendations. This may help understand the current difficulty to formulate redundant representations, which are proven to be successful in evolutionary computation.

1. Introduction

The influence of redundancy and neutrality on the behavior of the evolutionary search has been discussed in the evolutionary computation (EC) field. Some studies consider that the capacity that random mutations can contribute to the improvement of the evolutionary search can be achieved using redundant representations [1,2], while others alert that the addition of random redundancy does not appear to improve that capacity [3].

The neutral theory of molecular evolution proposed by the Japanese scientist Kimura [4] and the existence of redundancy and neutrality in the genetic code which appears in several ways are considered to be responsible for the evolutionary capacity of living beings. Kimura observed that in nature the number of different genotypes which store the genetic material

* Correspondence to: ESGHT, University of the Algarve, Faro, Portugal.
E-mail address: mcorreia@ualg.pt.

http://dx.doi.org/10.1016/j.tcs.2016.05.033
0304-3975/

of an individual greatly exceeds the number of different phenotypes, thus, the representation which describes how the genotypes are assigned to the phenotypes must be redundant, and neutral mutations become possible.

A mutation is neutral if its application to a genotype does not result in a change of the corresponding phenotype. As large parts of the genotype have no actual effect on the phenotype, evolution can use them as a store for genetic information that was necessary for survival in the past, and as a playground for developing new properties in the individual that could become advantageous in the future [5]. This has motivated the proposals of redundant representations in evolutionary computation [6–8].

The dynamics of evolutionary change in biology and of search in evolutionary optimization can be visualized as a population climbing the peaks of a fitness landscape [9]. Genetic changes such as mutations can be visualized as movements on a dimensional geographical map scaled up to a dimensionality corresponding to the number of genes. Some fitness landscapes may contain 'ridges' or level connected pathways of the same fitness, which are connected via single potential mutations and that are considered neutral networks [10].

There is a great interest in how redundant representations and neutral search spaces influence the behavior and the evolvability of evolutionary algorithms (EAs). Diverse redundant representations were used and, with some of these, the performance of the EAs improved and the evolvability, which is defined as the ability of random variations to sometimes produce improvements and as the genomes ability to produce adaptive variants when acted upon by the genetic system [11], is increased [2,1,5]. With others, as explained in [12,3], adding redundancy appeared to reduce performance with most of the problems applied. The loss of diversity in the population, the larger size of the genotypic search space and the fact that different genotypes that represent the same phenotype compete against each other, are some of the reasons stated for the deterioration of EAs performance. On the contrary, the ability of a population to adapt after changes (redundant and neutral representations allow a population to change the genotype without changing the phenotype) is one of the reasons given for the increased performance of EAs.

The research explained in this paper is an improvement of the research presented in [13,14], where the effect of redundancy and neutrality in the behavior of a simple evolutionary algorithm using two different families of redundant binary representations were studied. These families of representations proposed in [15–17], consist of neutral binary representations, denoted as $NN_g(\ell, k)$, based on the mathematical formulation of error control codes and of non-neutral binary representations, denoted as $NonNN_Z(\ell', k)$, based on linear transformations and which development was performed incrementally, starting with a non-coding redundant representation, and incorporating a different property in each phase, first polygeny, followed by pleiotropy.

This research gives new insights into the way the enumeration of $NN_g(\ell, k)$ representations can be done through the closure of an initial canonical representation under a neighborhood relationship, as well as presents a study of the properties of representations, such as, uniformity, connectivity, synonymity, locality and topology, which are commonly analyzed in the literature of EC field, and which are also studied for the family of $NN_g(\ell, k)$ representations. The performance of these representations in *NK* fitness landscapes with adjacent neighborhood are analyzed using a simple *(1+1)-ES* (Evolutionary Strategy) [18,19], which is modeled using Markov chains. This can be done because *(1+1)-ES* corresponds to a stochastic hill climbing [20], where an individual parent generates an individual child and the best of both becomes the parent in the next iteration.

The outline of this paper is as follows. Section 2 explains how redundancy and neutrality are manifested in biology, how they are used in EC field and the ways as redundancy and neutrality can be incorporated in redundant binary representations. In section 3 some properties of this type of representations are exposed. The error control codes used to define the neutral binary representations are presented in section 4. The family of neutral binary representations $NN_g(\ell, k)$ are presented in section 5, including the algorithms of coding and decoding and the definition of canonical neutral representation. Section 6 explains the definitions and algorithms used to enumerate the different neutral representations that can be generated. The properties explained in section 3 are extended to the $NN_g(\ell, k)$ representations and the relationships between these properties are exhibited in section 7. A brief explanation about the *NK* fitness landscapes is given in section 8. The ways as $NN_g(\ell, k)$ representations can change the ruggedness of *NK* fitness landscapes are analyzed in section 9. Section 10 presents how Markov chains can be used to model an *(1+1)-ES*, presenting the transition matrix for each type of representation and the probability of reaching the global optimum of the *NK* fitness landscapes. Section 11 presents how the performance of an *(1+1)-ES* is considered as the expected value to achieve the best solution and shows an analysis of the performance of $NN_g(\ell, k)$ representations versus the properties of these representations. Finally, section 12 discusses the conclusions and presents some guidelines for future work.

2. Redundant binary representations

Charles Darwin's theory of natural selection [21] assumes that natural selection is the driving force of evolution and that randomly advantageous mutations are fixed due to natural selection and can be propagated from generation to generation. On the other hand, the neutral theory of molecular evolution assumes that the driving force of molecular evolution is the random fixation of neutral mutation rather than the fixation of advantageous mutations by natural selection [5].

In biology, the genotype corresponds to the information stored in chromosomes, allowing to describe individuals at the level of genes, while the phenotype describes the appearance of the individual. In the genetic code, there are manifestations of redundancy and neutrality at several levels. For example, the genetic code is considered redundant because there are more combinations of nucleotides than amino acids formed, this is what is called the degeneracy of the genetic code [22].

Also, single secondary structure is generally represented by many different sequences, indicating a highly neutral redundant genotype–phenotype mapping because mutations in different positions in the sequences do not affect the secondary structure [23,24].

The mapping genotype–phenotype or representation uses the genotypic information to build the phenotype. Representations are considered redundant if the number of genotypes exceeds the number of phenotypes [5]. The notion of neutrality and neutral networks have attracted the attention of several researchers, due to the potential that neutral networks have in establishing alternative ways for the evolution of the population, allowing the improvement of the quality search.

There is not a single definition of neutrality and it is implicitly defined as the presence of neutral networks in the search space. A neutral network can be defined as a path through genotype space via mutations, which leave fitness unchanged [10], or as points in the search space that are connected through neutral point-mutations, where the fitness is the same for all the points in such network [7], or as a set of genotypes connected by single point mutations that map to the same phenotype [2]. The latter perspective is the one used in this research.

When using a binary representation, where genotypes and phenotypes are coded as strings with length ℓ, a genotype–phenotype mapping f_g of a non-redundant representation is defined as follows:

$$f_g : \{0,1\}^\ell \rightarrow \{0,1\}^\ell$$

where:

$$\forall (x_0, \ldots, x_{\ell-1}) \in \{0,1\}^\ell, \forall (y_0, \ldots, y_{\ell-1}) \in \{0,1\}^\ell,$$
$$f_g(x_0, \ldots, x_{\ell-1}) = f_g(y_0, \ldots, y_{\ell-1}) \Rightarrow (x_0, \ldots, x_{\ell-1}) = (y_0, \ldots, y_{\ell-1})$$

A metric has to be defined on the search space Φ when using search algorithms. Based on the metric, the distance $d(x_a, x_b)$ between two individuals $x_a \in \Phi$ and $x_b \in \Phi$ describes how similar the two individuals are. The larger the distance, the more different two individuals are [5]. In general, different metrics can be defined for the same search space, where different metrics result in different distances and different measurements of the similarity of solutions. In this way, two individuals are neighbors if the distance between two individuals is minimal. For example, when using the Hamming metric [25] for binary strings, the minimal distance between two individuals is $d = 1$. Therefore, two individuals x_a and x_b are neighbors if the distance $d(x_a, x_b) = 1$.

A way to produce a redundant code is to increase the number of genotypes, allowing for each phenotype to be represented by several genotypes. To use a redundant representation for solving an optimization problem, it is necessary to define a genotype–phenotype mapping f_g and a phenotype–fitness mapping f_p. Denoting Φ_g as the genotypic space and Φ_p as the phenotypic space, a genotype–phenotype mapping f_g determines which phenotypes result from the decoding of each genotypes and can be defined as follows:

$$f_g(x_g) : \Phi_g \rightarrow \Phi_p$$
$$x_g \mapsto x_p = f_g(x_g)$$

where $x_g \in \Phi_g$ denotes a genotype and $x_p \in \Phi_p$ denotes the corresponding phenotype. The genotype–phenotype mapping f_g only allows to determine which genotypes represent a particular phenotype, but does not give any information about the similarity between the solutions. The phenotype–fitness mapping assigns a fitness value $f_p(x_p)$ to each phenotype $x_p \in \Phi_p$:

$$f_p(x_p) : \Phi_p \rightarrow \mathbb{R}$$

The metrics used may be different in the genotypic search space and in the phenotypic search space. The metric of the phenotypic search space is determined by the problem to be solved, i.e., depends on the nature of the decision variables. For example, if the problem to be optimized is the *NK* fitness landscape, the Hamming metric can be used in the phenotypic search space. On the contrary, the genotypic search space metric depends not only on the space, but also on the search operator that is used on the genotypes. The search operator and the metric used in the genotypic search space determine each other and cannot be chosen independently of one another.

3. Properties of redundant binary representations

In order to clarify some of the questions in the literature about under which circumstances which types of redundant representations can be beneficial for EA, Rothlauf [5] identifies the following properties:

Uniformity A representation is uniformly redundant if all phenotypes are represented by the same number of genotypes;
Connectivity A representation has high connectivity if the number of phenotypes which are accessible from a given phenotype by single bit mutations is high;
Synonymity A representation is synonymously redundant if the genotypes that are assigned to the same phenotype are similar to each other;
Locality A representation has high locality if neighboring genotypes correspond to neighboring phenotypes.

To understand the properties presented, it is important to clarify the notion of phenotypic neighborhood. For a non-redundant binary representation, wherein the phenotype is of length k and in that the reachable phenotypes from any phenotype are all different, the connectivity is k. While in redundant representations, the phenotypic neighborhood of a phenotype corresponds to the phenotypes that are reachable from a particular phenotype by single bit mutations of the genotypes that represent it, connectivity corresponds to the number of different phenotypes, which constitute the phenotypic neighborhood of a phenotype. For example, for an uniform redundant representation, where genotypes have length ℓ and phenotypes have length k, if phenotype $x_p = f_g(x_g)$ is represented by the following m genotypes, $\{x_{g_1}, \ldots, x_{g_m}\}$ and as each genotype has ℓ neighbors, then the set of neighbors of the m genotypes is $\{x_{g_{11}}, \ldots, x_{g_{1\ell}}, \ldots, x_{g_{m1}}, \ldots, x_{g_{m\ell}}\}$. As each genotype represent a phenotype, the set of phenotypes of this set is:

$$f_g(x_{g_{11}}), \ldots, f_g(x_{g_{1\ell}}), \ldots, f_g(x_{g_{m1}}), \ldots, f_g(x_{g_{m\ell}})$$

The number of different elements of this set corresponds to the connectivity.

According to [5], a representation is synonymously redundant if the genotypic distances between all $x_g \in \Phi_g^{x_p}$ are small for all different x_p. In other words, if for all phenotypes the sum over the distances between all genotypes that represent the same phenotype is reasonably small, i.e.:

$$\sum_{x_p \in \Phi_p} \frac{1}{2} \Big(\sum_{x_{g_1} \in \Phi_g^{x_p}} \sum_{x_{g_2} \in \Phi_g^{x_p}} d(x_{g_1}, x_{g_2}) \Big) \tag{1}$$

where $x_{g_1} \neq x_{g_2}$, $d(x_{g_1}, x_{g_2})$ represents the distance between the genotype x_{g_1} and genotype x_{g_2} and $\Phi_g^{x_p}$ denotes the set of genotypes that represent the phenotype x_p. The $\frac{1}{2}$ fraction arises because the Hamming distance is symmetric.

Furthermore, the locality of a representation describes how well the genotypic neighborhood corresponds to the phenotypic neighborhood. The locality of a representation is a property that characterizes both redundant and non-redundant representations. The locality depends on the representation used f_g, on the metric defined in Φ_g and on the metric defined in Φ_p. Rothlauf [5] proposes the following expression for the locality d_m:

$$d_m = \sum_{d(x_{g_1}, x_{g_2}) = d^g{}_{min}} \Big| d(x_{p_1}, x_{p_2}) - d^p{}_{min} \Big| \tag{2}$$

where $d(x_{p_1}, x_{p_2})$ denotes the distance between the x_{p_1} and x_{p_2} phenotypes, $d(x_{g_1}, x_{g_2})$ describes the distance between the corresponding genotypes x_{g_1} and x_{g_2}, $d^g{}_{min}$ is the minimum distance between two neighboring genotypes and $d^p{}_{min}$ indicates the minimum distance between two neighboring phenotypes. When the Hamming metric is used in phenotypic and genotypic spaces, $d^g{}_{min} = 1$ and $d^p{}_{min} = 1$, respectively. Furthermore, $d_m = 0$ indicates that for all genotypes that are neighbors, their phenotypes are also neighbors and the representation is considered perfect according to the locality. When d_m is low, the representation has high locality and this means that small changes in the genotypic space correspond to small changes in the phenotypic space and vice-versa.

It is fair to say that the importance and the relationships between these properties are not yet fully understood. For example, Rothlauf [5] stated that using neutral search spaces, where the connectivity between the phenotypes is strongly increased by the use of redundant representation, allows many different phenotypes to be reached in one single search step, however, using non-synonymously redundant representations results in random search which decreases the efficiency of EAs. The author claimed that when using synonymously redundant representations, the connectivity between the phenotypes is not increased. This conclusion is not consistent with the results obtained with the neutral redundant representations presented in [14], as can be verified in section 7.6 of this paper, where synonymously redundant representations allow increasing the connectivity between phenotypes, when compared with the corresponding non-redundant representation.

Next, the error control codes [26,27] are explained as well as the way as these codes are used to add redundancy and neutrality to a binary representation.

4. Error control codes used in binary representations

In the binary case, the division of the genotypic space into subspaces in which the minimum Hamming distance between genotypes in the same space is, at least, two, would guarantee that single bit mutations produce genotypes in subspaces other than that of the original genotype. This is an essential characteristic of error control codes design, and results from this area would allow to obtain neutral binary representations with the following characteristics:

1. Split the redundant genotypic space of cardinality 2^ℓ into $2^{\ell-k}$ subspaces, each with cardinality 2^k, in such a way that a single bit change in the genotype causes a transition from one subspace to another;
2. Define bijective mappings between each genotypic subspace and the phenotypic space, in such a way that all genotypes which represent the same phenotype form a connected neutral network.

Table 1
Generation of the code polynomials using the $g(x) = x^3 + x + 1 = 1011$ generator polynomial.

U	$U(x)$	$x^3U(x)$	$C(x) = x^3U(x)\ mod\ g(x)$	V
0000	0	0	0	0000000
0001	1	x^3	$x+1$	0001011
0010	x	x^4	x^2+x	0010110
0011	$x+1$	x^4+x^3	x^2+1	0011101
0100	x^2	x^5	x^2+x+1	0100111
0101	x^2+1	x^5+x^3	x^2	0101100
0110	x^2+x	x^5+x^4	1	0110001
0111	x^2+x+1	$x^5+x^4+x^3$	x	0111010
1000	x^3	x^6	x^2+1	1000101
1001	x^3+1	x^6+x^3	x^2+x	1001110
1010	x^3+x	x^6+x^4	$x+1$	1010011
1011	x^3+x+1	$x^6+x^4+x^3$	0	1011000
1100	x^3+x^2	x^6+x^5	x	1100010
1101	x^3+x^2+1	$x^6+x^5+x^3$	1	1101001
1110	x^3+x^2+x	$x^6+x^5+x^4$	x^2	1110100
1111	x^3+x^2+x+1	$x^6+x^5+x^4+x^3$	x^2+x+1	1111111

The idea of using error control codes is to add redundancy to the message to be transmitted in a way that allows the receiver to detect and, if possible, to correct errors that may have occurred during the transmission. Among the many error control codes available, $C(\ell, k)$ block codes are defined on a binary alphabet which consists of a set of 2^k codewords with fixed length ℓ, that result from the encoding of the k bits of the data to be transmitted, after the addition of $\ell - k$ check bits and where the codewords form a vector subspace of dimension k. In particular, linear block codes may be defined as follows [26]:

Definition 1 *(Linear block code).* A block code of length ℓ and 2^k codewords is called a linear (ℓ, k) code if and only if its 2^k codewords form a k-dimensional subspace of the vector space of all the $\ell - tuples$ over the field $GF(2)$.

Cyclic codes [27] are an important subclass of linear block codes and are generally defined using the polynomial notation. In this case, a cyclic code $C(\ell, k)$ can be set as follows:

$$\begin{aligned} V(x) &= u_{k-1}x^{\ell-1} \ldots + u_0x^{\ell-k} + c_{\ell-k-1}x^{\ell-k-1} + \ldots + c_0 \\ &= x^{\ell-k}U(x) + C(x) \end{aligned} \tag{3}$$

where $V(x)$ corresponds to the code polynomial, $U(x)$ is the message polynomial consisting of the k bits of the message to be transmitted and $C(x)$ corresponds to the parity-check polynomial. The message polynomial $U(x)$ has degree up to $k-1$ and because it is a binary code, its coefficients are 0 or 1. The code polynomial $V(x)$ has degree up to $\ell - 1$.

A cyclic code can be defined using the generator polynomial $g(x)$ which have to be a factor of $x^\ell + 1$ and the degree of $g(x)$ is equal to $\ell - k$, the number of parity-check bits. As a consequence of these properties, any code polynomial is multiple of the generator polynomial and the generator polynomial is also a code polynomial. On the other hand, any polynomial factor of $x^\ell + 1$ with degree $\ell - k$ could be used as a generator polynomial, however, not all polynomials in these conditions seem to produce codes with the desirable features. To define a non-systematic cyclic code, the code polynomials $V(x)$ are obtained multiplying the message polynomials $U(x)$ by the generator polynomial $g(x)$, i.e.:

$$V(x) = U(x)g(x) \tag{4}$$

A more popular approach consists in generating the code polynomials by multiplying the message polynomials by $x^{\ell-k}$ and dividing the results by $g(x)$, which allows the code to be generated in *systematic* form (when the message polynomial can be obtained by discarding given $\ell - k$ components from the corresponding code polynomial):

$$V(x) = x^{\ell-k}U(x) + x^{\ell-k}U(x)\ mod\ g(x) \tag{5}$$

An example of the generation of the code polynomials using the generator polynomial $g(x) = x^3 + x + 1 = 1011$ is given in Table 1.

Using equation (4), it is possible to verify that only the code polynomials are divisible by the generator polynomial. When the received polynomial, which corresponds to the word received after transmission, is divided by the generator polynomial, if the remainder is null, the received polynomial corresponds to a code polynomial, otherwise, that polynomial does not correspond to a code polynomial and an error occurred during transmission.

In cyclic codes, the syndrome polynomial can be defined as:

$$S(x) = Z(x)\ mod\ g(x) \tag{6}$$

Table 2
Table of syndrome polynomials and error polynomials of the cyclic code defined in Table 1.

e	$e(x)$	s	$S(x)$
1000000	x^6	101	x^2+1
0100000	x^5	111	x^2+x+1
0010000	x^4	110	x^2+x
0001000	x^3	011	$x+1$
0000100	x^2	100	x^2
0000010	x	010	x
0000001	1	001	1

where $Z(x)$ corresponds to the received polynomial, which might not be equal to any of the code polynomials $V(x)$ if some errors occurred during transmission. In this case:

$$Z(x) = V(x) + e(x) \tag{7}$$

where $e(x)$ is the error polynomial. The syndrome polynomial can be written as:

$$S(x) = [V(x) + e(x)] \; mod \; g(x) \tag{8}$$

When $Z(x)$ corresponds to a code polynomial, the error polynomial is null, i.e., $e(x) = 0$. As $V(x) \; mod \; g(x) = 0$, the syndrome polynomial is equal to the remainder of the division of the error polynomial $e(x)$ by $g(x)$:

$$\begin{aligned} S(x) &= [V(x) + e(x)] \; mod \; g(x) \\ &= V(x) \; mod \; g(x) + e(x) \; mod \; g(x) \\ &= e(x) \; mod \; g(x) \end{aligned} \tag{9}$$

The syndrome of cyclic codes depends on the error pattern and does not depend on the received polynomial $Z(x)$. As $g(x)$ has degree of $\ell - k$, the degree of the syndrome is, at most, $\ell - k - 1$, and can be defined as:

$$S(x) = s_{\ell-k-1}x^{\ell-k-1} + s_{\ell-k-2}x^{\ell-k-2} \ldots + s_1 x + s_0 \tag{10}$$

It is also possible to use a table of syndrome polynomials and error polynomials for decoding. Table 2 shows this table for the cyclic code presented in Table 1. Using this table, the syndrome polynomial has to be calculated using equation (6), the error polynomial $e(x)$ is calculated from the syndrome polynomial and using equation (7) the code polynomial can be determined.

Provided that $g(x)$ is a primitive (or even merely irreducible) polynomial, it is easy to show that the code polynomial $V(x)$ cannot be of type x^i (otherwise $g(x)$ 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.

Thus, 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 follows:

$$V(x) = U(x)g(x) + z_i(x) \tag{11}$$

for every non-zero polynomial $z_i(x)$ of degree up to $\ell - k - 1$. Clearly, these classes are no longer linear because the sum of two polynomials in class C_i is not in C_i, 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. Moreover, changing a single coefficient in a polynomial $V(x) \in C$ will have the effect of producing a member of one of the cosets C_i. This is the property which is explored in error control codes and provides a suitable mechanism for making the sets of reachable phenotypes vary for different genotypic representations of the same phenotype.

5. Neutral binary representation $NN_g(\ell, k)$

If a genotype is seen as the code polynomial $V(x)$, of degree $\ell - 1$, of a linear code $C(\ell, k)$, generated by the generator polynomial $g(x)$, and the phenotype as the message polynomial $U(x)$, of degree $k - 1$, expression (5) allows to define the genotype based on the phenotype. As genotype $V(x)$ belongs to the 0 coset defined by the generator polynomial $g(x)$, the representation of $U(x)$ in each of the $2^{\ell-k}$ cosets can be defined by selecting $2^{\ell-k}$ coset leaders $z_i(x)$:

$$V_i(x) = V(x) + z_i(x) \tag{12}$$

To allow that the genotypes $V_i(x)$, which represent a $U(x)$, form a neutral connected network, it is enough that the coset leaders $z_i(x)$ define themselves a connected graph by links resulting from one bit change. When $U(x)$ corresponds to the 0 phenotype, $V(x)$ is also equal to zero and the $2^{\ell-k}$ polynomials $V_i(x)$ correspond to the $2^{\ell-k}$ coset leaders $z_i(x)$. In this context, and for this reason, the coset leaders will be designated by the *zeros* of the representation and the neutral network defined by the coset leaders forms a neutral representation of the $NN_g(\ell, k)$ (Neutral Network) family:

Definition 2 *(Neutral representation).* A neutral representation in the $NN_g(\ell, k)$ family consists of a mapping from a set of binary strings of length ℓ (the genotypic space) to the set of binary strings of length k (the phenotypic space), with $k < \ell$, which is characterized by the generator polynomial, $g(x)$, of degree $\ell - k$, and by the set of $2^{\ell-k}$ genotypes:

$$R = \{z_0, z_1, \ldots, z_{2^{\ell-k}-1}\}, z_i \in C_i, i = 0, \ldots, 2^{\ell-k} - 1$$

where each C_i denotes the subset of all genotypes with syndrome i, and each z_i is considered the coset leader of C_i and is called *zero*. For each pair (z_i, z_j), either $d(z_i, z_j) = 1$, or there is a set of genotypes $\{w_1, \ldots, w_p\} \subset R$, $p = d(z_i, z_j) - 1$, such that:

$$\begin{aligned} d(z_i, w_1) &= 1 \\ d(w_p, z_j) &= 1 \\ d(w_t, w_{t+1}) &= 1, \quad \forall t, 1 \leq t < p \end{aligned}$$

The coding of phenotype $U(x)$ to the genotypes $V_i(x)$, where $0 \leq i \leq 2^{\ell-k} - 1$, given the generator polynomial $g(x)$ and the neutral representation $\{z_0, \ldots, z_{2^{\ell-k}-1}\}$ of a family $NN_g(\ell, k)$, is explained in Algorithm 1:

Algorithm 1 Coding.

1: Input: $U(x)$, $g(x)$, $\{z_0, \ldots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$
2: Output: All $V_i(x)$
3: Calculate $V_0(x) = x^{\ell-k}U(x) + x^{\ell-k}U(x)$ *mod* $g(x)$ which corresponds to the code polynomial ('coset' 0 with syndrome 0).
4: **for** $i = 0$ to $2^{\ell-k} - 1$ **do**
5: Calculate $V_i(x)$ which results from the addition of *zero* $z_i(x)$ with polynomial $V_0(x)$
6: **end for**

The process of decoding from genotype $V(x)$ to phenotype $U(x)$, given the generator polynomial $g(x)$ and the neutral representation $\{z_0, \ldots, z_{2^{\ell-k}-1}\}$ of $NN_g(\ell, k)$, is performed by Algorithm 2:

Algorithm 2 Decoding.

1: Input: $V(x)$, $g(x)$, $\{z_0, \ldots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$
2: Output: $U(x)$
3: Determine the syndrome $S(x) = V(x)$ *mod* $g(x)$
4: Add $V(x)$ with the *zero* $z_i(x)$ of the neutral representation which has syndrome $S(x)$
5: Calculate the quotient of the division of $V(x) + z_i(x)$ by $x^{\ell-k}$ to obtain the phenotype $U(x)$ and dismiss the remainder

As an example, it is presented the decoding of genotype $V(x) = x^5 + x^4 + x^2$ or $v = 0110100 = 52$, using the neutral representation $\{0, 1, 9, 77, 4, 14, 6, 73\}$ of the family $NN_g(7, 4)$, generated by the generator polynomial $g(x) = x^3 + x + 1$. The syndrome is $S(x) = V(x)$ *mod* $g(x) = x^2 + 1$, and the *zero* with the same syndrome is $z_i(x) = x^3 + x^2 + x$. The polynomial addition of $V(x) + z_i(x) = (x^5 + x^4 + x^2) + (x^3 + x^2 + x) = x^5 + x^4 + x^3 + x$ and the quotient of the division of $V(x) + z_i(x)$ by $x^{\ell-k}$ allows to obtain the phenotype $U(x) = x^2 + x + 1$.

5.1. Equivalent representations

Having in mind that the values of the genotypes are not evaluated directly, it is assumed that the value of each gene (0 or 1) is not inherently important. In this sense, it is possible to establish an equivalence relationship between the representation defined by $R_1 = \{z_0, z_1, \ldots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$ and $R_2 = \{z_0 + c, z_1 + c, \ldots, z_{2^{\ell-k}-1} + c\} \in NN_g(\ell, k)$, for any constant c. In fact, for each genotype x_{g_1} that represents $x_p = f_{g_1}(x_{g_1})$, there is a corresponding x_{g_2} genotype, such that $x_p = f_{g_2}(x_{g_2})$, and any mutation of x_{g_1} corresponds to the same phenotype that the same mutation applied to x_{g_2}, which means that the phenotypic neighborhood of the two representations are equal.

To represent the set of representations whose *zeros* differ only by a constant c, it was decided to choose a canonical representation.

Definition 3 *(Canonical neutral representation).* The representation defined by the set of *zeros* $R = \{z_0, z_1, \ldots, z_{2^{\ell-k}-1}\}$, where the index i corresponds to the syndrome of each z_i, is a canonical representation of $NN_g(\ell, k)$ if and only if the vector $(z_0, z_1, \ldots, z_{2^{\ell-k}-1})$ is the minimum lexicographic of the set of vectors which correspond to all equivalent representations $\{z_0 + c, z_1 + c, \ldots, z_{2^{\ell-k}-1} + c\}$, $c \in \{0, 1\}^{\ell}$. The syndrome of $z_i + c$ can be different from the syndrome of z_i and the components of the vectors are ordered by the syndrome of $z_i + c$.

Given a representation $R = \{z_0, z_1, \ldots, z_{2^{\ell-k}-1}\}$, the canonical equivalent representation can be easily determined among the $2^{\ell-k}$ representations constructed as $R_i = \{z_0 + z_i, z_1 + z_i, \ldots, z_{2^{\ell-k}-1} + z_i\}$, for each $z_i \in R$. For example, Table 3 presents the representation equivalents to $\{0, 1, 9, 77, 4, 14, 6, 73\}$ and Table 4 shows the equivalents representations with *zeros* ordered by syndrome, that allows to verify that the given representation is already a canonical representation.

Table 3
Equivalent representations to $\{0, 1, 9, 77, 4, 14, 6, 73\} \in NN_g(7, 4)$.

	{0,1,9,77,4,14,6,73}	z_i
0	{0, 1, 9, 77, 4, 14, 6, 73}	z_0
1	{1, 0, 8, 76, 5, 15, 7, 72}	z_1
9	{9, 8, 0, 68, 13, 7, 15, 64}	z_2
77	{77, 76, 68, 0, 73, 67, 75, 4}	z_3
4	{4, 5, 13, 73, 0, 10, 2, 77}	z_4
14	{14, 15, 7, 67, 10, 0, 8, 71}	z_5
6	{6, 7, 15, 75, 2, 8, 0, 79}	z_6
73	{73, 72, 64, 4, 77, 71, 79, 0}	z_7

Table 4
Equivalent representations to $\{0, 1, 9, 77, 4, 14, 6, 73\} \in NN_g(7, 4)$ with *zeros* ordered by syndrome.

R_{eq_1}	{0, 1, 9, 77, 4, 14, 6, 73}
R_{eq_2}	{0, 1, 76, 8, 15, 5, 72, 7}
R_{eq_3}	{0, 68, 9, 8, 15, 64, 13, 7}
R_{eq_4}	{0, 68, 76, 77, 4, 75, 67, 73}
R_{eq_5}	{0, 10, 2, 77, 4, 5, 13, 73}
R_{eq_6}	{0, 10, 71, 8, 15, 14, 67, 7}
R_{eq_7}	{0, 79, 2, 8, 15, 75, 6, 7}
R_{eq_8}	{0, 79, 71, 77, 4, 64, 72, 73}

6. Enumeration of neutral representations

Although each representation of $NN_g(\ell, k)$ has been defined by the choice of their set of *zeros*, it is important not to forget that the elements in this set have restrictions of syndrome and connectivity. For this reason, it is important to consider how representations of such a family can be generated.

The proposed approach is to define a *neighborhood* relationship between representations of the same family and to calculate a set of canonical representations as the closure [28] of an initial canonical representation under that relationship. The closure of an initial canonical representation of a family of representations under the *neighborhood* relationship is obtained by replacing one *zero* to the representation, until no more representations can be created for that family.

Definition 4 *(Neighboring neutral representations).* Two neutral representations of a $NN_g(\ell, k)$ family defined by the sets of *zeros* $R_1 = \{z_{1,0}, z_{1,1}, \ldots, z_{1,2^{\ell-k}-1}\}$ and $R_2 = \{z_{2,0}, z_{2,1}, \ldots, z_{2,2^{\ell-k}-1}\}$ are considered neighboring if and only if the sets R_1 and R_2 differ by one element, i.e., $\exists i \in \{0, \ldots, 2^{\ell-k} - 1\}$: $\forall j \in \{0, \ldots, 2^{\ell-k} - 1\}$, $j \neq i \Leftrightarrow z_{1,j} = z_{2,j}$.

It is important to note that neighboring representations of a canonical representation will not necessarily have to be canonical representations. However, it is possible to adopt a definition of alternative neighborhood, applied to the canonical representations set. It should be noted that the addition of the same constant c to the *zeros* of two neighboring representations results into two other representations, each one neighboring of each other.

Definition 5 *(Neighboring canonical neutral representations).* Two canonical neutral representations of the $NN_g(\ell, k)$ family defined by the sets of *zeros* $R_1 = \{z_{1,0}, z_{1,1}, \ldots, z_{1,2^{\ell-k}-1}\}$ and $R_2 = \{z_{2,0}, z_{2,1}, \ldots, z_{2,2^{\ell-k}-1}\}$ are considered neighbors if and only if there is a third representation R_3, not necessarily canonical, that is neighbors of R_1 according to Definition 4, which is equivalent to R_2.

Algorithm 3 calculates the neighboring representations of a canonical neutral representation. One neutral representation of $NN_g(\ell, k)$ will have a maximum of $2^{\ell-k}\ell$ neighboring representations, where some of them can be equal to the representation in question by symmetry issues. It is important to note that a neighboring representation is valid when the neutral network is connected and it is only analyzed after its canonical equivalent representation has been found.

The neighboring representations of $\{0, 1, 2, 3\} \in NN_g(4, 2)$ and the equivalent canonical representations are indicated in Table 5.

6.1. Closure of $NN_g(\ell, k)$ under Neighborhood relationship

The determination of the canonical representations of the $NN_g(\ell, k)$ family can be performed on the basis of the calculation of the closure of an initial canonical representation under the *Neighborhood* relationship between canonical representation (Definition 5). Assuming that the canonical representations set is closed under the neighborhood relationship, this calculation will enumerate all representations. In the absence of a demonstration of the validity of this assumption, it is possible to consider that, at least, a part of the desired representations will be found.

Algorithm 3 Calculate neighboring representations *NeighRep*.

```
Input: Neutral representation {0, z_1, ..., z_{2^{ℓ-k}-1}} ∈ NN_g(ℓ, k)
Output: Neighboring representations of neutral representation NeighRep
R = {z_0, z_1, ..., z_{2^{ℓ-k}-1}}
Neighboring representations set is empty NeighRep = ∅
while there are zeros z_i ∈ R do
    Calculate neighboring of zero z_i, such that Neigh(z_i) = {neigh_1(z_i), neigh_2(z_i), ..., neigh_ℓ(z_i)}
    while there are neigh_m ∈ Neigh(z_i) do
        Calculate syndrome s(neigh_m)
        Search the zero with the same syndrome s(z_j) = s(neigh_m)
        Replace the zero found z_j with neigh_m
        if representation R is connected then
            Determine the canonical representation equivalent to R
            Add to NeighRep
        end if
    end while
end while
```

Table 5
Neighboring representations of $\{0, 1, 2, 3\} \in NN_g(4, 2)$ and the equivalent canonical representations.

R_1	$\{0, 1, 2, 4\} \Leftrightarrow \{0, 1, 2, 4\}$
R_2	$\{0, 8, 2, 3\} \Leftrightarrow \{0, 1, 2, 10\}$
R_3	$\{0, 1, 5, 3\} \Leftrightarrow \{0, 1, 2, 4\}$
R_4	$\{9, 1, 2, 3\} \Leftrightarrow \{0, 1, 2, 10\}$
R_5	$\{0, 6, 2, 3\} \Leftrightarrow \{0, 1, 2, 4\}$
R_6	$\{0, 1, 2, 10\} \Leftrightarrow \{0, 1, 2, 10\}$
R_7	$\{7, 1, 2, 3\} \Leftrightarrow \{0, 1, 2, 4\}$
R_8	$\{0, 1, 11, 3\} \Leftrightarrow \{0, 1, 2, 10\}$

The algorithm starts with the $\{0, 1, \ldots, 2^{\ell-k} - 1\}$ representation. This is a canonical representation of $NN_g(\ell, k)$, whatever ℓ and k are considered and corresponds to the representations with non-coding bits. Thus, the starting set is:

$$A = \{\{0, z_1, \ldots, z_{2^{\ell-k}-1}\}\} \tag{13}$$

The set D of neighboring representations of each representation is calculated by Algorithm 3. Algorithm 4 performs the calculation of the closure A under the *Neighborhood* relationship:

Algorithm 4 Calculate the closure A under *Neighborhood* relationship.

```
Input: A = {0, 1, ..., 2^{ℓ-k} - 1} ∈ NN_g(ℓ, k)
Output: ClosureA
ClosureA = A
while there are representations to analyze in ClosureA do
    Calculate D, the set of neighboring canonical representations of the representation analyzed
    for each neighboring representation in D do
        if neighboring representation does not belong to ClosureA then
            Add to ClosureA
        end if
    end for
end while
```

Table 6 presents the number of canonical neutral representations of $NN_g(\ell, k)$ found for $0 < k \leq 11$, when the number of redundant bits is in $0 < \ell - k \leq 4$ and the generator polynomial $g(x)$ has degree $0 < \ell - k \leq 4$.

As observed, the number of neutral representations grows rapidly with k and very rapidly with the number of redundant bits $\ell - k$. This huge number creates difficulties not only on execution time but also on the storage capacity of the generated representations. For example, for a $|\Phi_p| = 2^2 = 4$ phenotypic space, if 3 redundant bits are added, 266 neutral representations are obtained, while 1 845 628 are reached when 4 redundant bits are joined.

As the number of neutral representations obtained for some $NN_g(\ell, k)$ is very high, it was decided to use a freeware library, Oracle Berkeley DB that provides a high performance bases from embedded data and can be used with the C programming language, C++, Java, Perl, Python, among others. In this case, the C language was used. This database allows you to manipulate data in the order of terabytes in various systems including UNIX and Microsoft Windows.

7. Properties of $NN_g(\ell, k)$

To analyze the relationships between the properties of the representations and the performance of $NN_g(\ell, k)$ applied to *NK* fitness landscapes, the properties of $NN_g(\ell, k)$ are presented and discussed [29].

Table 6
Number of neutral representations of $NN_g(\ell, k)$.

	$x+1$	x^2+x+1	x^3+x+1	x^4+x+1	$g(x)$
1	2	4	19	1489	
2	3	9	266	1 845 628	
3	4	18	1828		
4	5	32	8128		
5	6	50	25 100		
6	7	75	66 750		
7	8	108	158 784		
8	9	147	342 701		
9	10	196	690 748		
10	11	256	1 307 528		
11	12	324	2 350 336		
k					

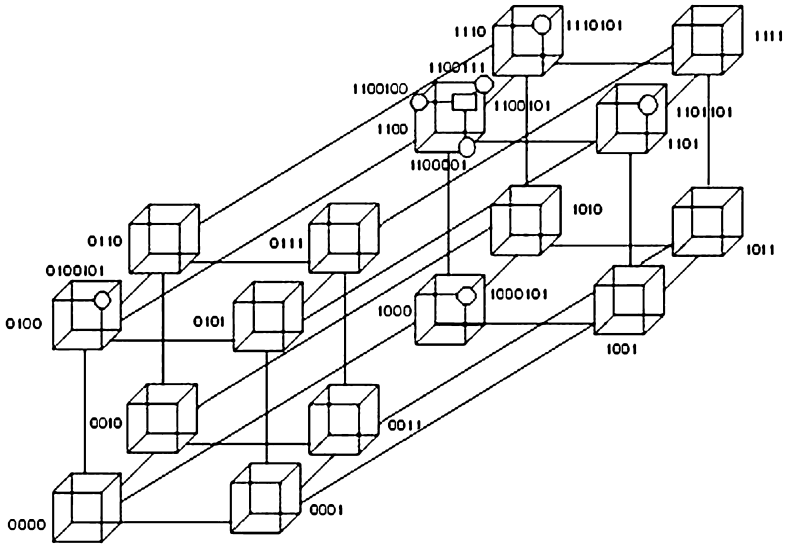

Fig. 1. Hypergraph representing a genotypic space with $|\Phi_g| = 2^\ell = 2^7$.

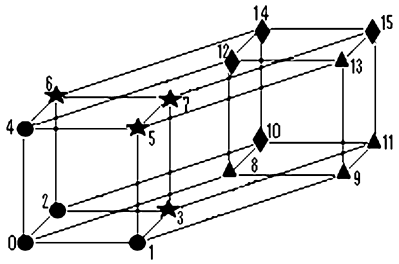

Fig. 2. Hypergraph representing the genotypic space with $\ell = 4$ and the phenotypes of the neutral representation $\{0, 1, 2, 4\} \in NN_g(4, 2)$.

7.1. Uniformity

The $NN_g(\ell, k)$ family is considered uniform because all phenotypes are represented by the same number of genotypes. In each representation of $NN_g(\ell, k)$, each phenotype is represented by $2^{\ell-k}$ genotypes.

7.2. Phenotypic neighborhood and connectivity

Using hypergraphs is a way of visualizing the phenotypic neighborhood of a representation. Fig. 1 displays a genotypic space with cardinality $|\Phi_g| = 2^\ell = 2^7$, where for the sake of simplicity, only the 1100101 genotype (□) and its 7 neighbors (○) are presented, while for all other genotypes, only the first 4 bits are indicated.

For example, Fig. 2 presents a hypergraph of the genotypic space of the neutral representation $\{0, 1, 2, 4\} \in NN_g(4, 2)$. In this case, phenotype 0 is represented by a ●, phenotype 1 by a ★, phenotype 2 by a ▲ and, finally, phenotype 3 by a ♦.

Table 7
Genotypes and corresponding phenotypes [] of neutral representation $\{0, 1, 2, 4\} \in NN_g(4, 2)$.

syndrome 0	syndrome 1	syndrome 2	syndrome 3
0000 [00]	0001 [00]	0010 [00]	0100 [00]
0111 [01]	0110 [01]	0101 [01]	0011 [01]
1001 [10]	1000 [10]	1011 [10]	1101 [10]
1110 [11]	1111 [11]	1100 [11]	1010 [11]

Table 8
Neighboring genotypes and corresponding phenotypes of the neutral network that represents phenotype 01 of $\{0, 1, 2, 4\} \in NN_g(4, 2)$.

Neigh. gen.	1	2	3	4
0111	0110 [01]	0101 [01]	0011 [01]	1111 [11]
0110	0111 [01]	0100 [00]	0010 [00]	1110 [11]
0101	0100 [00]	0111 [01]	0001 [00]	1101 [10]
0011	0010 [00]	0001 [00]	0111 [01]	1011 [10]

Table 9
Maximum connectivity for each $NN_g(\ell, k)$.

	$x+1$	x^2+x+1	x^3+x+1	x^4+x+1	x^5+x^2+1	$g(x)$
1	1	1	1	1	1	
2	2	3	3	3	(3)	
3	3	5	7	(7)	(7)	
4	4	7	14	(15)	(15)	
5	5	9	18	(30)	(31)	
6	6	11	22	(43)	(59)	
7	7	13	26	(55)	(98)	
8	8	15	30	(63)	(125)	
9	9	17	34	(–)		
10	10	19	38	(–)		
11	11	21	42	(91)		
k						

Table 7 presents the genotypes and the corresponding phenotypes [] obtained using the neutral representation presented in Fig. 2. As the representation is uniform, each phenotype is represented by $2^{\ell-k} = 2^2 = 4$ genotypes, which corresponds to the number of *zeros* of the neutral representation. Furthermore, as each phenotype is represented by a neutral network with $2^{\ell-k}$ genotypes, each of which has ℓ neighboring genotypes, including, at least, a genotype which decodes to the same phenotype, the phenotypic neighborhood of each phenotype has, at most, $2^{\ell-k}(\ell-1)$ different phenotypes. The actual number of different phenotypes in this neighborhood determines the connectivity of the representation, which is the same for all phenotypes defined by the representation, although the reached phenotypes can be different. As can be confirmed with Table 8, the connectivity of $\{0, 1, 2, 4\} \in NN_g(4, 2)$ is 3, because it is possible to reach 3 different phenotypes (excluding itself). When comparing this representation with a non-redundant representation with phenotype length of 2, in which each phenotype can only reach 2 different phenotypes, the neutral representation $\{0, 1, 2, 4\} \in NN_g(4, 2)$ can reach a larger number of different phenotypes, in this case 3.

In conclusion, the way as the neutral networks of these representations are structured allowed that:

1. All neutral representations $NN_g(\ell, k)$ have a connectivity higher or equal to the connectivity of the corresponding non-redundant representation, i.e., higher than or equal to k;
2. The connectivity of any representation of $NN_g(\ell, k)$ does not vary with the phenotype and the number of different reachable phenotypes is equal for all phenotypes of the same neutral representation;
3. Different genotypes of a neutral network can reach different phenotypes.

Table 9 presents the maximum connectivity for each $NN_g(\ell, k)$, where for the neutral representations whose numeration was not possible due to the limitations explained in section 5, a lower limit for the maximum connectivity was obtained by generating neutral representations from the neutral representation with the greatest connectivity found until the moment, and leaving the algorithm running until no more increasing in connectivity was detected for a given period of time. These values appear inside () in Table 9. Note that the maximum connectivity appears to increase exponentially with the number of redundant bits $\ell - k$.

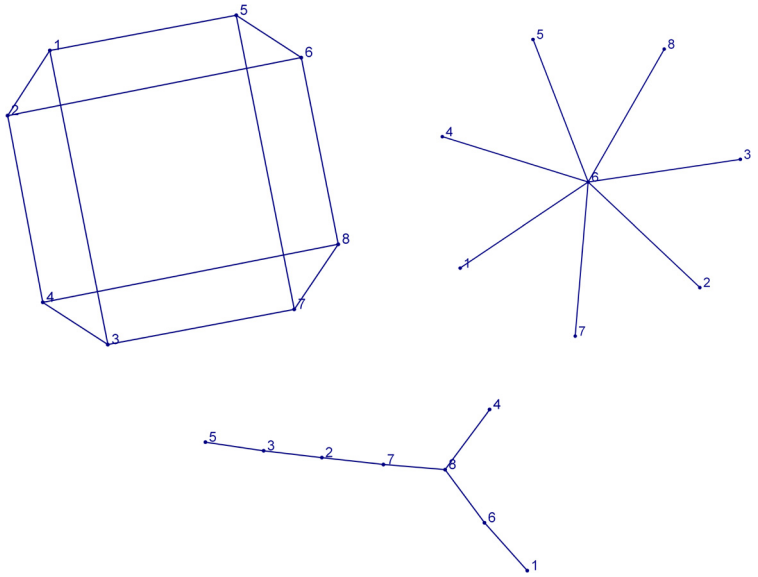

Fig. 3. MDS of some $NN_g(7, 4)$ representations.

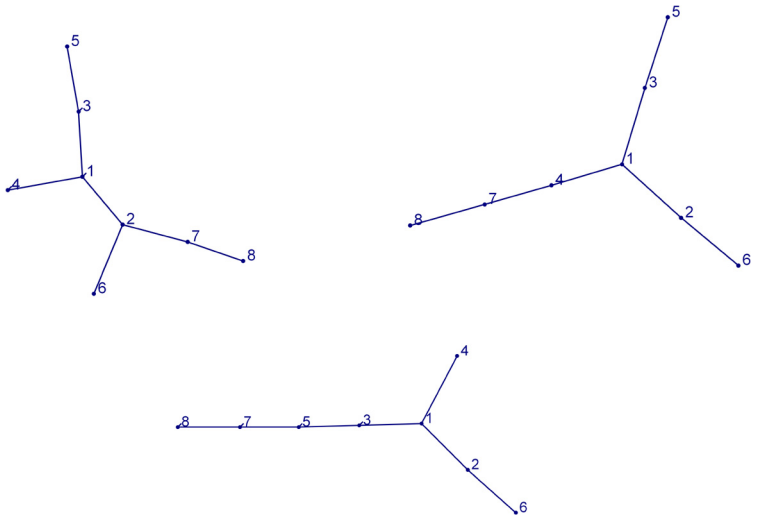

Fig. 4. MDS of some $NN_g(14, 11)$ representations.

7.3. Topology

As the $NN_g(7, 4)$ representations proposed exhibit neutral networks, the topology is an appropriated property to be analyzed. In order to visualize the topology of a neutral network, a technique called MDS (Multi-Dimensional Scaling) was used [30]. This technique consists in determining the positions of a set of points in the Euclidean space, such that the distances between them approximate a given measure of dissimilarity of the objects that are to be visualized. Neutral network topologies may be visualized in this way, by taking the Hamming distance between the genotypes of the neutral networks as the MDS dissimilarity measure. Fig. 3 shows the topologies of some neutral networks of the $NN_g(7, 4)$ family. While the non-coding redundant representations, in which the genotypic redundant bits do not affect the phenotype, display the topology of a cube, the representations with maximum connectivity (14) in the $NN_g(7, 4)$ family exhibit a Y topology.

Fig. 4 shows the topologies of some neutral networks of the $NN_g(14, 11)$ representations, which presents the greatest connectivity (42), while Fig. 5 displays the topologies of some representations of the $NN_g(6, 2)$ family.

7.4. Synonymity

The calculation of the synonymity follows the equivalence classes approach proposed by [5], in which if all genotypes $x_g \in \Phi_g^{x_p}$ belong to the same equivalence class and if the sum of the genotypic distances between the individuals that belong

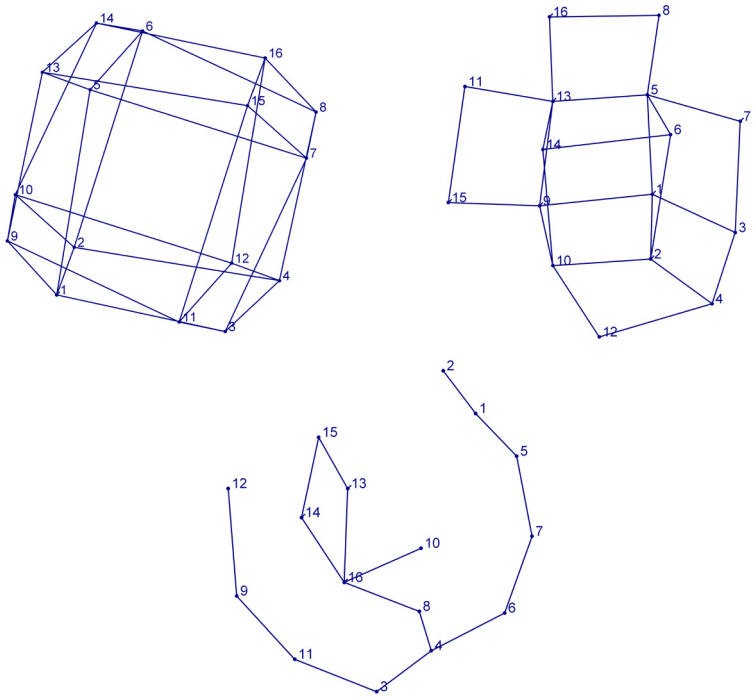

Fig. 5. MDS of some $NN_g(6,2)$ representations.

Table 10
Minimum and maximum synonymity for each $NN_g(\ell,k)$.

	$x+1$	x^2+x+1	x^3+x+1	x^4+x+1	$g(x)$
1	1	1.33–1.5	1.71–2.18	2.13–2.67	
2	1	1.33–1.67	1.71–2.71		
3	1	1.33–1.67	1.71–2.86		
4	1	1.33–1.67	1.71–2.86		
5	1	1.33–1.67	1.71–3		
6	1	1.33–1.67	1.71–3		
7	1	1.33–1.67	1.71–3		
8	1	1.33–1.67	1.71–3		
9	1	1.33–1.67	1.71–3		
10	1	1.33–1.67	1.71–3		
11	1	1.33–1.67	1.71–3		
k					

to the same equivalence class is small, then the representation is considered synonymously redundant. Thus, synonymity for $NN_g(\ell,k)$ representations can be simplified using expression (14) because the distances between all genotypes of the neutral network that represent a given phenotype are equal for all phenotypes:

$$S_{yn} = \frac{\sum_{i=0}^{2^{\ell-k}-1}\sum_{j=0}^{2^{\ell-k}-1} d(z_i,z_j)}{(2^{\ell-k})\times(2^{\ell-k}-1)} \tag{14}$$

where $d(z_i,z_j)$ corresponds to the Hamming distance between *zero* z_i and *zero* z_j. This expression gives a normalized measure of synonymity, since the total number of distances were considered. A representation is considered synonymously redundant when the value of S_{yn} is low and non-synonymously redundant when this value is high. As the synonymity is a measure of how close the genotypes that represent the same phenotype are, the synonymity is related with the neutral networks topology. Table 10 shows the minimum and maximum synonymity for $NN_g(\ell,k)$.

7.5. Locality

As the locality does not vary with different phenotypes for $NN_g(\ell,k)$, this measure can be calculated on the basis of a single phenotype, leading to a simplification of expression (2). Thus, locality for the $NN_g(\ell,k)$ family can be calculated using the average distance from a phenotype to the corresponding neighboring phenotypes, including the phenotype in question. Table 11 presents the minimum and maximum locality calculated for some $NN_g(\ell,k)$.

Table 11
Minimum and maximum locality for each $NN_g(\ell, k)$.

	$x+1$	x^2+x+1	x^3+x+1	x^4+x+1	$g(x)$
1	0.5	0.33–0.5	0.25–0.50	0.2–0.57	
2	0.67–1.0	0.5–1.0	0.4–1.0		
3	0.75–1.25	0.6–1.5	0.5–1.58		
4	0.8–1.4	0.67–1.67	0.57–2.0		
5	0.83–1.5	0.71–1.86	0.62–2.38		
6	0.86–1.57	0.75–2.0	0.67–2.78		
7	0.88–1.62	0.78–2.0	0.7–2.95		
8	0.89–1.67	0.8–2.1	0.73–3.14		
9	0.9–1.7	0.82–2.18	0.75–3.25		
10	0.91–1.73	0.83–2.17	0.77–3.27		
11	0.92–1.75	0.85–2.23	0.79–3.36		
k					

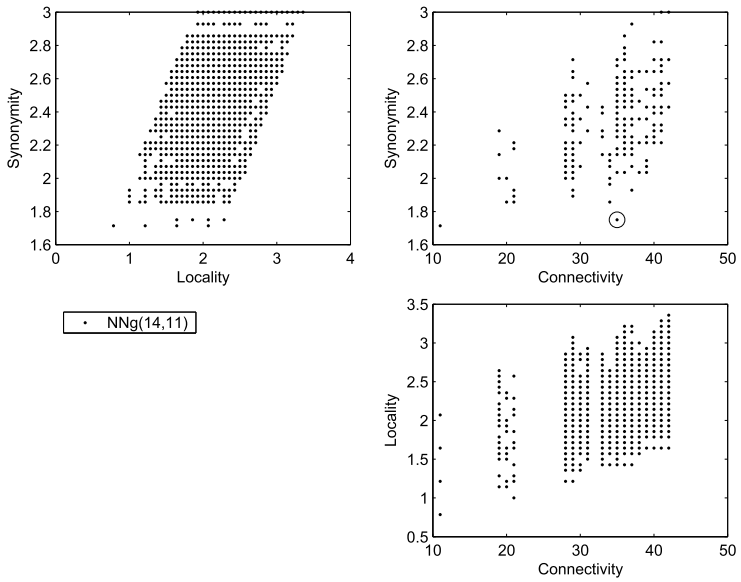

Fig. 6. Relationships between connectivity, synonymity and locality of $NN_g(14, 11)$.

There are other properties such as innovation rate [31], diameter of connected neutral networks, accessibility of neighboring neutral networks [32] and others that will be analyzed in a future work.

7.6. Relationships between properties of $NN_g(\ell, k)$

Fig. 6 presents scatter graphs showing the relationship between connectivity, locality and synonymity of the $NN_g(14, 11)$ family, which corresponds to the family of representations with the highest value of k that was completely enumerated. Although the scatter graphs only show the properties for the $NN_g(14, 11)$ family, the results are similar for the remaining families of $NN_g(\ell, k)$. These scatter graphs show that there are representations with low locality that reach high connectivity values. The chart at the right-top in Fig. 6 contradicts the idea presented in [5] that synonymously redundant representations do not allow the connectivity between the phenotypes to increase in comparison with the corresponding non-redundant representations. The circle in this chart shows that there is a set of synonymously neutral representations that have high values of connectivity. The $NN_g(\ell, k)$ family has a variety of representations with different properties that provide an important basis for the study of the relationships between them and will allow to study to what extent these properties can affect the performance of an evolutionary algorithm.

8. *NK* fitness landscapes

In order to study the effect of redundancy and neutrality on evolutionary algorithm performance, evolutionary strategies, originally proposed by [18,19] for integer and continuous spaces, are used in this research for binary spaces and with constant mutation rate. In particular, an evolutionary strategy *(1+1)-ES* is applied to *NK* fitness landscapes [33] with different degrees of ruggedness.

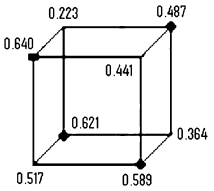

Fig. 7. Hypergraph representing an instance of a $NK(3,2)$ fitness landscape.

The *NK* fitness landscapes proposed by [33] are based on the fitness landscapes proposed by [34]. This author found that when fitness interactions exist between genes, the genetic composition of a population can evolve into multiple domains of attraction [35]. On the other hand, genes can suppress or enable the effects of other genes, and this interaction is called epistasis. Wright established a conceptual link between this microscopic property of the organisms (epistasis) and a macroscopic property of the evolutionary dynamics (multiple attractors of the population in the area of genotypes), and used the analogy with a landscape with multiple peaks to illustrate this situation. Fitness landscapes may be defined as follows [36]:

Definition 6 *(Fitness landscape).* A landscape is a triplet (S, V, f) where S is a set of potential solutions that match the search space, $V : S \to 2^S$ is a function that assigns to each $s \in S$ a set of neighbors $V(s) \subset S$, defining the neighborhood structure, and $f : S \to \mathbb{R}$ is a fitness function that assigns a value of fitness to each solution s.

The *NK* fitness landscapes allow to explore the way as epistasis can control the ruggedness of the landscape. To this purpose, fitness functions family whose ruggedness could be shaped by a single parameter was developed. *NK* fitness landscapes are defined as stochastically generated fitness functions of bit strings with length N and interactions between groups of $K+1$ bits. The fitness of each bit depends on its value and the values of the bits with which it interacts. For each instance of the landscape, fitness values associated with each bit are realizations of a random variable with uniform distribution between 0.0 and 1.0, and the fitness of each chromosome is the average of the bits fitness for all N loci.

The ruggedness of the landscape is controlled by the parameter K and reaches its maximum when $K = N-1$ and its minimum when $K = 0$. When $K = 0$, there is no epistasis, the landscape has a unique local optimum (the global optimum), and the fitness of different chromosomes is highly correlated with the Hamming distance between them. When $K = N-1$, the number of interactions between bits is maximum and the function is purely random, so the landscape has many local optima, and the fitness of chromosomes is not correlated with the Hamming distance between them.

The choice of the K bits is known to affect the computational complexity of the *NK* fitness landscape optimization problem. When these bits are the K nearest to the locus that is being assessed, the problem is polynomial in N, whereas it becomes NP-hard when $K > 1$ and such bits are randomly chosen [37].

Next the definitions of local optimum and global optimum in *NK* fitness landscapes are given as well as how the $NN_g(\ell, k)$ representations can change the ruggedness of the *NK* fitness landscapes.

9. Ruggedness of *NK* fitness landscapes

A local optimum [38] corresponds to a point in the fitness landscape that has no neighbors with higher fitness. A local optimum will be the global optimum if it is the best solution for the problem. A definition of local optimum and global optimum is given [36]:

Definition 7 *(Local optimum).* Given a fitness landscape (S, V, f), a solution $s^* \in S$ is a local optimum if and only if:

$$\forall s \in V(s^*),\ f(s) \leq f(s^*)$$

Definition 8 *(Global optimum).* Given a fitness landscape (S, V, f), a solution $s^* \in S$ is a local optimum if and only if:

$$\forall s \in S,\ f(s) \leq f(s^*) \tag{15}$$

As the *NK* fitness landscape can be gradually tuned from smooth to rugged, it is a good fitness model to study different types of representations. A way to characterize the nature of the landscape is to understand the ruggedness or smoothness of the landscape based on the number of local optima, also known as modality [38], and the distribution of those local optima. As it is known, a landscape is induced by the operator which is used to define neighborhoods [39]. In this case, as the landscape is defined over the binary space, the Hamming metric is used and the neighborhood relation can be represented by a hypergraph. Next, an example is given that shows that $NN_g(\ell, k)$ representations can change the number of local optima of the *NK* fitness landscapes, changing in this way the difficulty of the problem.

Fig. 7 represents an instance of a $NK(3,2)$ fitness landscape, where $N = 3$ and $K = 2$. Having in mind the definitions given in section 9, the global optimum in this instance corresponds to genotype 100 with fitness 0.640, while 001, 010

Table 12
Genotypes that decode for each phenotype of $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$.

Syndrome	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	17
1	11	10	9	8	15	14	13	26
2	22	23	20	21	18	19	16	7
3	29	28	31	30	25	24	27	12
4	39	38	37	36	35	34	33	54
5	44	45	46	47	40	41	42	61
6	49	48	51	50	53	52	55	32
7	58	59	56	57	62	63	60	43
Phenotype								

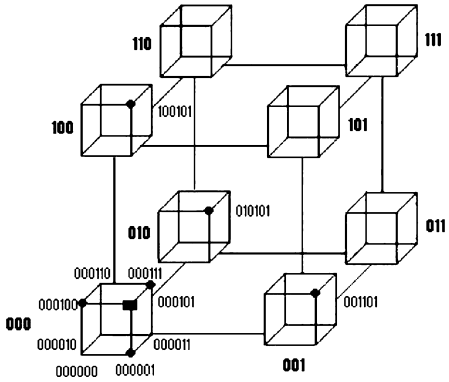

Fig. 8. Hypergraph representing a binary genotypic space with $\ell = 6$.

and 111 are local optima with fitness 0.589, 0.621 and 0.487, respectively. Graphically, the global optimum in Fig. 7 is represented as a rectangle, while the local optima are represented as diamonds.

An example is given using the neutral representation $\{0, 1, 2, 3, 4, 5, 6, 17\} \in NN_g(6, 3)$ and the fitness landscape of Fig. 7. The genotype–phenotype mapping of neutral representation is presented in Table 12, where the genotypes that decode to the same phenotype appear in the same row, ordered by the syndrome. In order to make the table more compact, all values are presented in decimal.

Fig. 8 shows a hypergraph representing the genotypic space when it is used this representation. For the sake of simplicity, only some genotypes are indicated in the figure. As $\ell = 6$ each genotype has 6 neighbors. A genotype (■) and its neighbors (●) are shown in the figure.

Assigning to each genotype of the neutral representation the corresponding phenotype fitness, the fitness landscape of Fig. 7, which presents 4 local optima, is transformed into a landscape with just one optimum (global).

Let us take as an example the phenotype 1 with fitness 0.589 that corresponds to a local optimum when it is used the non-redundant representation. As it can be verified, this phenotype is no more a local optimum because there are neighbors, in this case genotypes 7 and 18, which decode to phenotype 2, which has higher fitness 0.621. The other genotypes of phenotype 1 do not have as neighbors any genotypes with higher fitness, but may reach one of these two genotypes through neutral mutations.

Analyzing the others phenotypes that are also local optima when using the non-redundant representation, it turns out that the landscape changes from a multimodal to an unimodal landscape. Starting from any genotype is possible to access to phenotype 4 with fitness 0.640. For example, from genotype 15, jump to genotype 7, finishing at genotype 39, which has fitness 0.640.

Experiments conducted with the family of $NN_g(14, 11)$ allowed to transform $NK(11, 10)$ with hundreds of local optima into landscapes with just few local optima, despite the number of redundant bits introduced be just 3. According to [35], the expected value of the number of local optima in a fitness landscape with $K = N - 1$ is given by $2^N/(N + 1)$. For the fitness landscapes $NK(11, 10)$, this number is $2^{11}/(11 + 1) = 170.67$. In a typical instance of $NK(11, 10)$ with 170 local optima, some representations of the family $NN_g(14, 11)$ reduced the number of local optima to 31, while others raised this number to 191. Clearly, not all neutral representations transform multimodal landscapes into unimodal landscapes. The opposite can also happen. In conclusion, the number of local optima can decrease, to be equal or increase when is used a representation of $NN_g(\ell, k)$.

In this research, the *NK* fitness landscapes were used, but some authors [40] used the tunably neutral *NK* landscapes proposed by [41] and developed by [42] to investigate the effects of neutrality and ruggedness on topologies of fitness landscape in test problems. The results of their studies demonstrated that landscapes with a higher degree of neutrality

have the larger sizes of neutral networks and that in landscapes with the lowest degree of rudggedness, all networks lead to networks with the highest fitness via any networks.

Given the definition of neutral network adopted (connected network), it is necessary to review the concept of a local optimum. As all genotypes that represent a particular phenotype are connected, instead of a local optimum, it is possible to think of a *locally optimal neutral network*. Having in mind the definition of local optimum considered in this section, all genotypes of the neutral network that do not directly reach genotypes with better fitness are considered local optima. However, it is sufficient that one of the genotypes of the neutral network that represents a particular phenotype not to be a local optimum, so that it can walk through the neutral network and escape it through the genotype that can access genotypes with better fitness. This structure can be understood as a 'bridge' or a 'plateaux' in the fitness landscape.

10. Markov chain modeling of *(1+1)-ES*

This section explains how to calculate the probability of reaching the global optimum of NK fitness landscapes using a stochastic hill climber modeled as a Markov chain. It is important to know that the evolution process of a stochastic hill climbing can be modeled as a Markov Chain [43]. The variant of the *Hill Climber* heuristic used here considers that given the current individual i, a neighbor j is randomly generated and is accepted if and only if it has a fitness not lower than the fitness of i.

The modeling of an evolutionary algorithm, specifically an evolutionary strategy *(1+1)-ES*, may be done through Markov chains [43], because this strategy corresponds to a stochastic hill climbing [20], where an individual parent generates an individual child and the best of both becomes the parent in the next iteration.

Definition 9 *(Markov chain).* A Markov chain is a sequence of discrete random variables X_0, X_1, X_2,..., which satisfies the Markov property, i.e.:

$$\begin{aligned}&\text{Prob}(X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = \\ &\text{Prob}(X_{n+1} = i_{n+1} | X_n = i_n)\end{aligned} \tag{16}$$

The Markov property means that the system has no memory, i.e., the value of the next state only depends on the present state. In this paper, only finite state spaces and homogeneous Markov chains, where the transition probabilities are constant in time, are considered. The transition matrix is defined as $P = [p_{ij}]$, where:

$$p_{ij} = \text{Prob}(X_{n+1} = j | X_n = i) \tag{17}$$

and $\sum_{j=1}^{|S|} p_{ij} = 1$, where $|S|$ is the number of states [43].

The probability that a chain with $|S|$ states is in state j at a given instant $n+1$, is given by:

$$\begin{aligned}&\text{Prob}(X_{n+1} = j) = \textstyle\sum_{i=1}^{|S|} \text{Prob}(X_{n+1} = j \wedge X_n = i) \\ &= \textstyle\sum_{i=1}^{|S|} \text{Prob}(X_{n+1} = j | X_n = i)\, \text{Prob}(X_n = i)\end{aligned} \tag{18}$$

provided that the probability distribution of state X in the previous instant, i.e., $\text{Prob}(X_n = i)$, $i = 1, \ldots, |S|$, is known.

Whereas $v_{n,i} = \text{Prob}(X_n = i)$ and that the distribution of X_n is given by vector $\vec{v}_n = (v_{n,1}, \ldots, v_{n,|S|})$, the distribution of X_{n+1} can be calculated from the transition matrix P:

$$\vec{v}_{n+1} = \vec{v}_n P \tag{19}$$

If the distribution of the initial state, denoted $\vec{v}_0$, is known, the distribution of X_n is given by:

$$\vec{v}_n = \vec{v}_0 P^n \tag{20}$$

In the case of an *(1+1)-ES*, the current state corresponds to the parent in each iteration, and the state space matches the search space. In the strategy used, the child is generated by mutation of a single bit of the parent genotype, replacing it in the next iteration if its fitness is not less than the fitness in the current iteration.

The transition probabilities depend both on the mutation operator and on the fitness function. Since the neighborhood is finite and small, the transition matrix P is sparse. This is useful because it facilitates the storage of the P matrix in memory, and reduces the computation time of the state distribution according to expression (19).

10.1. Transition matrix for non-redundant representation

A non-redundant representation with genotypes of length k, where $f_p(x_{p_i})$ corresponds to the fitness of an individual x_{p_i} and the neighbors of x_{p_i} with fitness greater than or equal to $f_p(x_{p_i})$ are defined as [43]:

$$VM_{x_{p_i}} = \left\{ x_{p_j} | d(x_{p_i}, x_{p_j}) = 1 \wedge f_p(x_{p_j}) \geq f_p(x_{p_i}) \right\} \tag{21}$$

Each element of the transition matrix of the Markov chain P is defined as follows:

$$p_{ij} = \begin{cases} \frac{1}{k} & i \neq j \wedge x_{p_j} \in VM_{x_{p_i}} \\ 1 - \frac{|VM_{x_{p_i}}|}{k} & i = j \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

If an individual x_{p_i} only has neighbors with lower fitness, then it is a local optimum and $VM_{x_{p_i}} = \emptyset$. In this case, the chain will remain in this state, and x_{p_i} is called an absorbing state. In the transition matrix, the values in a row which corresponds to a local optimum are all 0, except for the value on the diagonal of the matrix, which is 1.

10.2. Transition matrix for $NN_g(\ell, k)$

Consider a neutral representation $\{z_0, z_1, \ldots, z_{2^{\ell-k}-1}\} \in NN_g(\ell, k)$, where ℓ corresponds to the number of bits of x_{g_i} genotypes, f_g denotes the genotype–phenotype mapping, f_p denotes the phenotype–fitness mapping, the phenotype $x_{p_i} = f_g(x_{g_i})$ corresponds to genotype x_{g_i} and the set of neighbors of x_{g_i} with fitness greater than or equal to $f_p(x_{p_i})$ is defined as follows:

$$VM_{x_{g_i}} = \left\{x_{g_j} | d(x_{g_i}, x_{g_j}) = 1 \wedge f_p(x_{p_j}) \geq f_p(x_{p_i})\right\} \tag{23}$$

Each element of the $2^\ell \times 2^\ell$ dimension transition matrix of the Markov chain P is defined as follows:

$$p_{ij} = \begin{cases} \frac{1}{\ell} & i \neq j \wedge x_{g_j} \in VM_{x_{g_i}} \\ 1 - \frac{|VM_{x_{g_i}}|}{\ell} & i = j \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

10.3. Probability of reaching the global optimum

The Algorithm 5 calculates the probability of reaching the global optimum in function of the number of iterations, having as input parameters the global optimum, the transition matrix P and the number of iterations:

Algorithm 5 Calculate the probability of reaching the global optimum.

```
1: Input: global optimum, transition matrix P, number of iterations
2: Output: Probabilities
3: y = global optimum
4: v[1, :] = 1/dim(P)
5: for t from 2 to number of iterations do
6:     v[t, :] = v[t − 1, :] × P
7:     p[t] = sum(v[t, y])
8: end for
```

At first, all states have the same probability, so all components of the vector $v[1:]$ are equal to $\frac{1}{dim(P)}$, where $dim(P)$ corresponds to the number of rows of the matrix P. For the neutral representation, the global optimum is represented by a neutral network, then the probability that the global optimum y is reached at each time t, corresponds to the sum of the probabilities of the state to be equal to each of the genotypes which represented it, i.e., $p[t] = sum(v[t, y])$.

11. Analysis of the performance of $NN_g(\ell, k)$ versus $NN_g(\ell, k)$ properties

As a first approach, only the probability of achieving the global optimum was considered, but as was reported in a research developed in [14], it was not a good indicator. In that research, the performance of the $NN_g(\ell, k)$ representations was compared with the performance of the $NonNN_Z(\ell', k)$ representations. The study allow to detect that, in the long term, sometimes it is the $NN_g(\ell, k)$ representation which has a higher probability of reaching the global optimum than the $NonNN_Z(\ell', k)$ representation which exhibits the same phenotypic neighborhood, or vice versa.

Thus, instead of using only the probability of achieving the global optimum, the evolution of the expected value of fitness of the *NK* landscape in function of time was considered as a good indicator of a representation performance. This value is calculated from the probability distributions provided by the Markov chain as follows:

1. Determine the expected value of fitness for each pair representation-instance of the *NK* fitness landscape;
2. Rank all representations based on that value, separately, for each instance of the *NK* fitness landscape, since the value of fitness is not comparable between different instances.

To guarantee that the number of local optima of the *NK* fitness landscapes does not increase with the redundant representations used, representations in which phenotypic neighborhood contains the non-redundant representation neighborhood were the only ones considered. This set of representations matches 12 819 representations of $NN_g(14, 11)$.

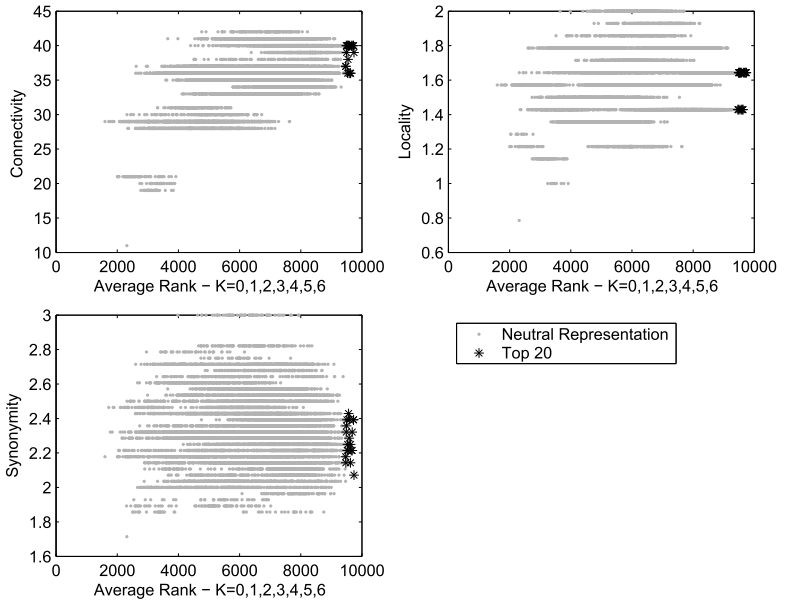

Fig. 9. Performance of the selected 12 819 representations of $NN_g(14, 11)$ *versus* neutral representations properties – $NK(11, K)$ fitness landscapes.

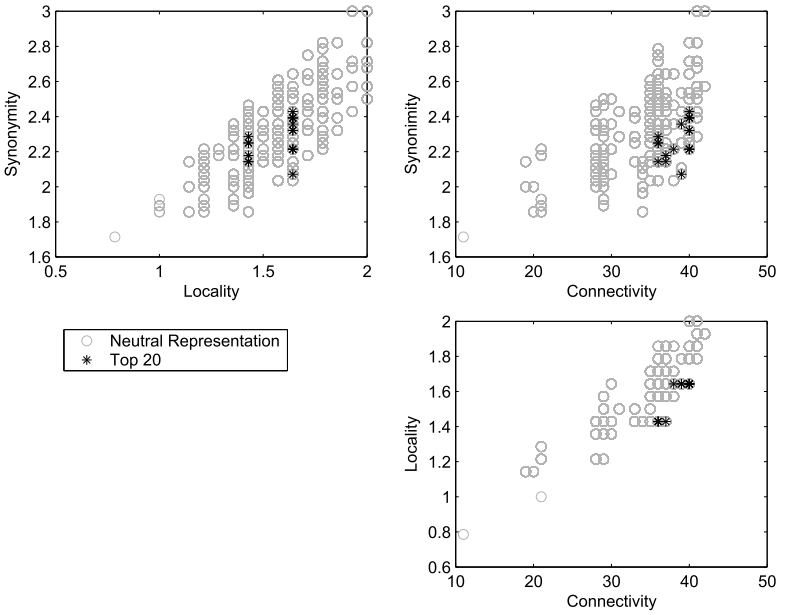

Fig. 10. Neutral representations properties *versus* neutral representations properties – $NK(11, K)$ fitness landscapes.

As can be seen in Fig. 9, the performance of the selected representations is related with the synonymity, the locality and the connectivity of the representations. That figure shows that the 20 representations with better performance obtained in $NK(11, K)$, where $K = 0, 1, 2, 3, 4, 5, 6$, present high values, but not extreme of connectivity and intermediate values of locality and synonymity.

The same can be seen in Fig. 10, which shows that the connectivity, locality, and synonymity are positively correlated considering the set of the representations analyzed.

In Figs. 11, 12 and 13, the neutral representation with the topology in cube that corresponds to the non-coding redundant representation, appears inside a circle. This representation presents a connectivity of 11, a $S_{yn} = 1,71$ and $d_m = 0,79$, i.e., it has high locality (has the lowest value of d_m), but presented one of the worst performances. However, for $K = 0$, this representation presented the best performance, corroborating another conclusion forwarded by [5] that representations with high locality are the most suitable for an efficient evolutionary search in easy problems. The performance of this representation worsens with the increase of K, as can be visualized in Fig. 11.

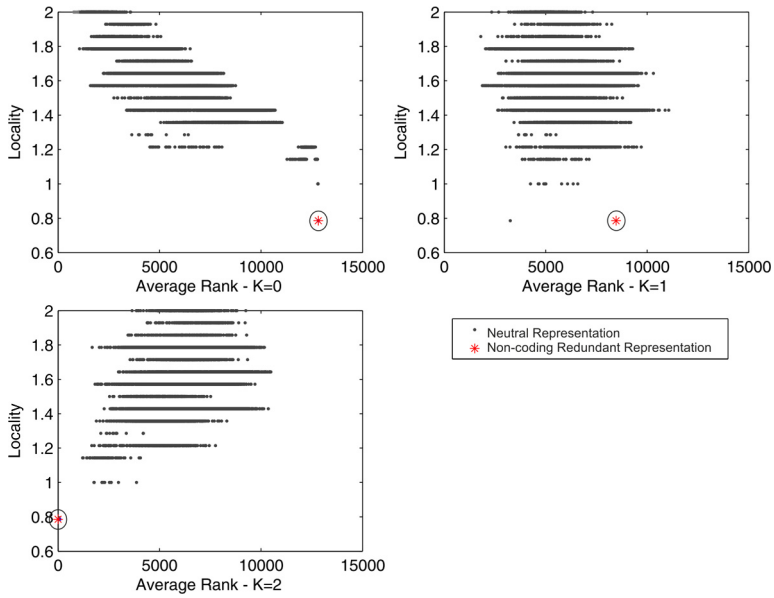

Fig. 11. Performance of the selected representations of $NN_g(14, 11)$ versus locality.

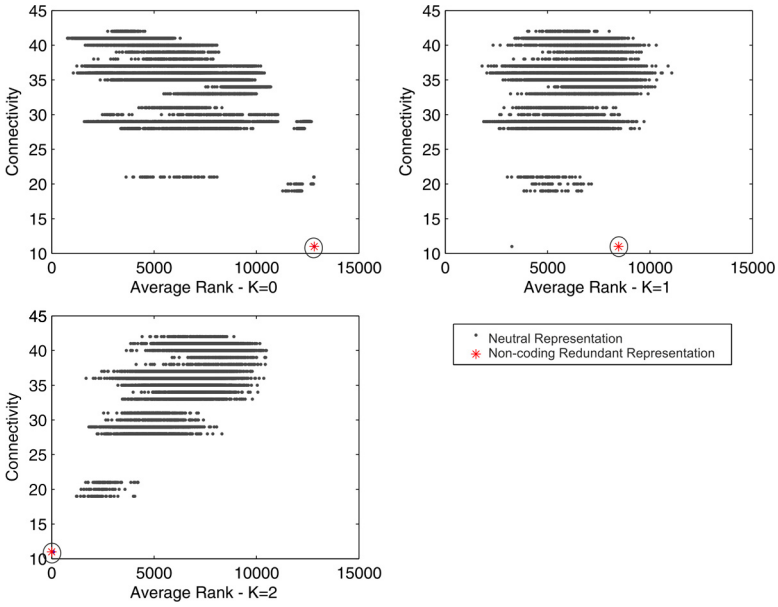

Fig. 12. Performance of the selected representations of $NN_g(14, 11)$ versus connectivity.

Furthermore, as stated in Fig. 12, with easy problems, the connectivity does not need to be high in order to get good results. On the other hand, when the difficulty of the problem grows, the representations that present best performances exhibit values of connectivity between intermediate to high, although this characteristic does not guarantee by itself a good performance.

Finally, observation of Fig. 13 indicates that when the problem is easy, synonymously redundant representations have a good performance. With the increased difficulty of the problem, the representations that present best results have intermediate values of synonymity. However, it must be considered the fact stated above that these three measures are related to each other in $NN_g(\ell, k)$. This becomes difficult the identification of the relationships of cause and effect between each one individually and the performance that they lead. The identification of these relationships is one of the paths for future work.

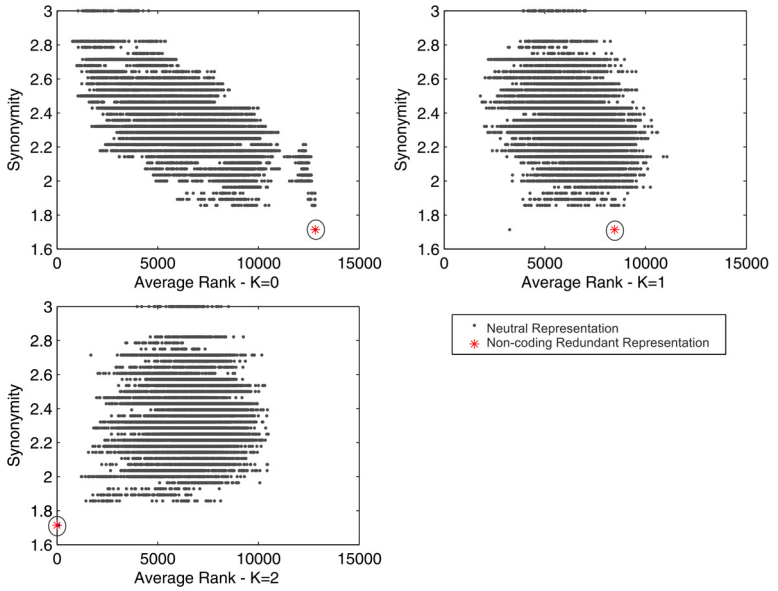

Fig. 13. Performance of the selected representations of $NN_g(14, 11)$ versus synonymity.

12. Conclusions

The existence of redundancy and neutrality in the genetic code and the neutral theory of molecular evolution proposed by Kimura [4] have motivated the use of redundant and neutral representations in EC. This inspired the definition of a family of neutral representations, denoted as $NN_g(\ell, k)$, which is based on the mathematical formulation of the linear block codes, in particular the Hamming codes and its formulation as cyclic codes. In general, the idea of using error control codes is to add redundancy to the message to be transmitted in a way that allows the receiver to detect and, if possible, to correct errors that may have occurred during the transmission.

For each family of $NN_g(\ell, k)$ representations, for $0 < k \leq 11$, where the number of redundant bits is in the range $0 < \ell - k \leq 4$, the neutral representations were generated using the closure algorithm. Each representation of this family is defined by the neutral network topology and exhibits different properties that provide an important basis for the study of the relationships between them and allowed to study to what extent these properties can affect the performance of an evolutionary algorithm. These properties are uniformity, connectivity, synonymity, locality and topology, which are also commonly studied in the literature of EC.

An *(1+1)-ES* (Evolutionary Strategy) was used and applied to *NK* fitness landscapes with adjacent neighborhood, and its behavior was modeled as Markov chains. The performance of the *(1+1)-ES* was considered as the expected value for achieving the best solution. The results showed that it is possible to design synonymously redundant representations that allow an increase of the connectivity between phenotypes and that for easy problems, $NN_g(\ell, k)$ representations with high locality, synonymously redundant and where it is not necessary high values of connectivity are the most suitable for an efficient evolutionary search. Increasing the difficulty of the problem, $NN_g(\ell, k)$ representations with low locality, which present connectivity between intermediate to high and with intermediate values of synonymity are the best ones.

In conclusion, $NN_g(\ell, k)$ representations with better performance in *NK* fitness landscapes do not present extreme values of connectivity, locality and synonymity. This conclusion is contrary to what one would expect when taking into account the literature recommendations.

The results achieved in this study open multiple paths for future work. One of them is to identify the relationships of cause and effect between each one of the analyzed properties and the performance that they lead. Another future work is to apply the $NN_g(\ell, k)$ representations to other problems as the 0–1 Quadratic Programming problem, and finally, to use *NK* fitness landscapes with random neighborhood instead of adjacent neighborhood in order to verify if the conclusions obtained are the same.

References

[1] R. Shipman, M. Shackleton, M. Ebner, R. Watson, Neutral search spaces for artificial evolution: a lesson from life, in: Proceedings of the Seventh International Conference on Artificial Life, 2000, pp. 162–169.
[2] M. Shackleton, R. Shipman, M. Ebner, An investigation of redundant genotype–phenotype mappings and their role in evolutionary search, in: Proceedings of the 2000 Congress on Evolutionary Computation (CEC00), vol. 1, 2000, pp. 493–500.
[3] J.D. Knowles, R.A. Watson, On the utility of redundant encodings in mutation-based evolutionary search, in: Proceedings of the 7th Conference on Parallel Problem Solving From Nature (PPSN VII), 2002, pp. 88–98.
[4] M. Kimura, Evolutionary rate at the molecular level, Nature 217 (1968) 624–626.

[5] F. Rothlauf, Representations for Genetic and Evolutionary Algorithms, 2nd edition, Springer, 2006.
[6] R. Poli, E. Galván-López, The effects of constant and bit-wise neutrality on problem hardness, fitness distance correlation and phenotypic mutation rates, IEEE Trans. Evol. Comput. 16 (2012) 279–299.
[7] E. Galván-López, R. Poli, A. Kattan, M. O'Neill, A. Brabazon, Neutrality in evolutionary algorithms ... what do we know?, Evolv. Syst. 2 (2011) 145–163.
[8] L.F. Simoes, D. Izzo, E. Haasdijk, A.E. Eiben, Self-adaptive genotype–phenotype maps: neural networks as a meta-representation, in: Parallel Problem Solving from Nature – PPSN XIII, in: LNCS, vol. 8672, 2014, pp. 110–119.
[9] P.F. Stadler, Towards a theory of landscapes, in: R. López-Pena, H. Waelbroeck, R. Capovilla, R. Garcia-Pelayo, F. Zertuche (Eds.), Complex Systems and Binary Networks, Springer, Berlin, 1995, pp. 78–163.
[10] I. Harvey, A. Thompson, Through the labyrinth evolution finds a way: a silicon ridge, in: Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (ICES96), in: LNCS, vol. 1259, 1996, pp. 406–422.
[11] G.P. Wagner, A. Lee, Complex adaptations and the evolution of evolvability, Evolution 50 (1996) 967–976.
[12] S. Ronald, J. Asenstorfer, M. Vincent, Representational redundancy in evolutionary algorithms, in: Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, vol. 2, 1995, pp. 631–636.
[13] M.B. Correia, The effect of redundancy and neutrality in genetic search, in: Proceedings of the Third World Congress on Nature and Biologically Inspired Computing (NaBIC 2011), 2011, pp. 250–255.
[14] M.B. Correia, A study of redundancy and neutrality in evolutionary optimization, Evol. Comput. 21 (3) (2013) 413–443.
[15] C.M. Fonseca, M.B. Correia, Developing redundant binary representations for genetic search, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC05), vol. 2, 2005, pp. 1675–1682.
[16] M.B. Correia, C.M. Fonseca, On the roles of redundancy and neutrality in evolutionary optimization: an experimental study, in: Proceedings of the 2007 Genetic and Evolutionary Computation Conference (GECCO07), vol. 2, 2007, p. 1504.
[17] M.B. Correia, C.M. Fonseca, How redundancy and neutrality may affect evolution on NK fitness landscapes, in: Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC07), 2007, pp. 2842–2849.
[18] I. Rechenberg, Evolutionsstragie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Holzboog, 1973.
[19] H.-P. Schwefel, Numerische Optimierung von Computer-Modellen mittels der Evolutionsstragie, Birkhäuser, 1977.
[20] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, 1996.
[21] C. Darwin, On the Origin of Species by Means of Natural Selection, John Murray, London, 1859.
[22] S.C. Manrubia, J.A. Cuesta, Neutral networks of genotypes: evolution behind the curtain, ARBOR Ciencia, Pensamiento y Cultura CLXXXVI 746 (noviembre–diciembre 2010) 1051–1064.
[23] P. Schuster, W. Fontana, P. Stadler, I. Hofacker, From sequences to shapes and back: a case study in RNA secondary structures, Proc. - Royal Soc. B., Biol. Sci. 255 (1994) 279–284.
[24] P. Schuster, Landscapes and molecular evolution, Physica D 107 (1997) 351–365.
[25] R. Hamming, Coding and Information Theory, Prentice-Hall, London, 1980.
[26] S. Lin, D. Costello, Error Control Coding: Fundamental and Application, Prentice-Hall Series in Computer Application in Electrical Engineering, 1983.
[27] A.B. Carlson, Communication Systems: An Introduction to Signals and Noise in Electrical Communication, 4th edition, McGraw-Hill College, 2002.
[28] H. Lewis, C. Papadimitriou, Elements of the Theory of Computation, 2nd edition, Prentice-Hall, 1998.
[29] M.B. Correia, Study of some properties of binary neutral representations, in: Proceedings of the 7th Iberian Conference on Information Systems and Technologies (CISTI 2012), 2012, pp. 673–678.
[30] J. de Leeuw, Multidimensional scaling, UCLA Statistics Preprint 274, 2000.
[31] M. Ebner, M. Shackleton, R. Shipman, How neutral networks influence evolvability, Complexity 7 (2001) 19–33.
[32] P.K. Lehre, P.C. Haddow, Accessibility between neutral networks in indirect genotype–phenotype mappings, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC05), vol. 1, 2005, pp. 419–426.
[33] S.A. Kauffman, At Home in the Universe – The Search for the Laws of Self-Organization and Complexity, Oxford University Press, New York, 1995.
[34] S. Wright, The roles of mutation, inbreeding, crossbreeding and selection in evolution, in: Proceedings of the Sixth International Congress on Genetics, vol. 1, 1932, pp. 356–366.
[35] L. Altenberg, NK fitness landscapes, in: T. Bäck, D. Fogel, Z. Michalewicz (Eds.), The Handbook of Evolutionary Computation, Oxford University Press, 1997.
[36] S. Verel, Etude et exploitation des réseaux de neutralité dans les paysages adaptatifs pour l'optimisation difficile, Ph.D. thesis, l'Université de Nice-Sophia Antipolis, 2005.
[37] A.H. Wright, R.K. Thompson, J. Zhang, The computational complexity of n–k fitness functions, IEEE Trans. Evol. Comput. 4 (2000) 373–379.
[38] J. Horn, D.E. Goldberg, Genetic algorithm difficulty and the modality of fitness landscapes, in: Proceedings of the 1995 Conference on Foundations of Genetic Algorithms (FOGA), vol. 3, 1995, pp. 243–269.
[39] C.R. Reeves, Landscapes, operators and heuristic search, Ann. Oper. Res. 86 (1999) 473–490.
[40] Y. Katada, K. Ohkura, Analysis on topologies of fitness landscapes with both neutrality and ruggedness based on neutral networks, in: Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO09), 2009, pp. 1855–1856.
[41] M. Newman, R. Engelhardt, Effects of selective neutrality on the evolution of molecular species, Proc. R. Soc. Lond. 265 (1403) (1998) 1333–1338.
[42] T. Smith, P. Husbands, P. Layzell, M. O'Shea, Fitness landscapes and evolvability, Evol. Comput. 10 (1) (2002) 1–34.
[43] P.G.H. Toonen, Markov chain analysis of evolutionary database optimization, 1994.