Neutrality and Self-Adaptation

Christian Igel
(christian.igel@neuroinformatik.ruhr-uni-bochum.de) and
Marc Toussaint
(marc.toussaint@neuroinformatik.ruhr-uni-bochum.de)
Lehrstuhl für theoretische Biologie, Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum

Abstract. Neutral genotype-phenotype mappings can be observed in natural evolution and are often used in evolutionary computation. In this article, important aspects of such encodings are analyzed.

First, it is shown that in the absence of external control neutrality allows a variation of the search distribution independent of phenotypic changes. In particular, neutrality is necessary for self-adaptation, which is used in a variety of algorithms from all main paradigms of evolutionary computation to increase efficiency.

Second, the average number of fitness evaluations needed to find a desirable (e.g., optimally adapted) genotype depending on the number of desirable genotypes and the cardinality of the genotype space is derived. It turns out that this number increases only marginally when neutrality is added to an encoding presuming that the fraction of desirable genotypes stays constant and that the number of these genotypes is not too small.

Keywords: evolutionary computation, genotype-phenotype mapping, neutrality, No-Free-Lunch theorem, redundancy, self-adaptation

Abbreviations: EA – evolutionary algorithm; GP – genetic programming; HI – human immunodeficiency; NFL – No-Free-Lunch; tRNA – transfer ribo-nucleic acid

1. Introduction

The potential beneficial effects of neutral, i.e., non-injective, genotype-phenotype mappings in natural evolution have extensively been discussed (Kimura, 1968; Conrad, 1990; Huynen, 1996; van Nimwegen et al., 1999; Wilke, 2001; Schuster, 2002). Hence, there is a growing interest in investigating how these results from biology carry over to evolutionary algorithms (Barnett, 1998; Newman and Engelhardt, 1998; Shipman, 1999; Shackleton et al., 2000; Weicker and Weicker, 2000; Ebner et al., 2001; Toussaint and Igel, 2002; Toussaint, 2003).

In this article, we point out that in the absence of external control, neutrality provides a way to vary the search distribution independent of phenotypic variation. We propose to define this use of neutrality for adapting the search strategy as *generalized self-adaptation*, which contains the classical self-adaptation methods as special cases since

final version in Natural Computing 2(2), 117–132, 2003

strategy parameters can be regarded as neutral traits. Thus, neutrality is a necessity for (classical) self-adaptation, which is a key concept in evolutionary computation and which is widely accepted as a means to improve the performance of evolutionary algorithms (EAs) for problem classes relevant in practice (Bäck, 1998).

In contrast to this potential positive aspect of neutrality, a bijective genotype-phenotype mapping is often regarded as a general design principle for efficient evolutionary computation. This is based on the consideration that the smaller the search space the faster the search (Radcliffe, 1991). Utilizing the sharpened no-free-lunch (NFL) theorem (Schumacher et al., 2001; Igel and Toussaint, 2003), we determine the average number of fitness evaluations needed to find a desirable (e.g., optimal) solution depending on the size of the genotype space and the number of genotypes mapped to desirable solutions. This enables us to investigate the influence of redundancy on the average performance of search algorithms in general.

We introduce our formalism in the following section. Section 3 extends the results by Toussaint and Igel (2002): The relation between self-adaptation and neutrality is described and some examples from natural and simulated evolution are given. Thereafter, we derive the average number of fitness evaluations needed to find an optimum in an NFL scenario and discuss its dependency on neutrality.

2. Evolutionary Exploration

In this section, we outline that the individuals in the population, the genotype-phenotype mapping, and the variation operators including their parameters imply a parameterization of the space of exploration distributions on the phenotype space.

2.1. General Global Random Search and Evolutionary Algorithms

Evolutionary algorithms can be considered as a special class of global random search algorithms. Let the search problem under consideration be described by a quality function $f : \mathcal{G} \to \mathcal{F}$ to be optimized, where $\mathcal{G}$ denotes the search space and $\mathcal{F}$ the space of cost values. The general global random search scheme can be described as follows (Zhigljavsky, 1991; Igel and Kreutz, 2003):

1. Choose a joint probability distribution $P^{(t)}_{\mathcal{G}^\lambda}$ on $\mathcal{G}^\lambda$. Set $t \leftarrow 1$.

2. Obtain λ points $\boldsymbol{g}_1^{(t)}, \dots, \boldsymbol{g}_\lambda^{(t)}$ by sampling from the distribution $P_{\mathcal{G}^\lambda}^{(t)}$. Evaluate these points using f (perhaps with random noise).
3. According to a fixed (algorithm dependent) rule construct a new probability distribution $P_{\mathcal{G}^\lambda}^{(t+1)}$ on $\mathcal{G}^\lambda$.
4. Check for some appropriate stopping condition; if the algorithm has not terminated, substitute $t \leftarrow t+1$ and return to step 2.

We call the probability distribution $P_{\mathcal{G}}^{(t)}(\boldsymbol{g})$ that $\boldsymbol{g} \in \mathcal{G}$ is sampled in iteration t the *search distribution on* $\mathcal{G}$. It can be computed from $P_{\mathcal{G}^\lambda}^{(t)}(\boldsymbol{g}_1, \dots, \boldsymbol{g}_\lambda)$, the *joint search distribution on* $\mathcal{G}$. Random search algorithms can differ fundamentally in the way they represent and alter the joint search distribution. Typically, it is represented by a semi-parametric model. The choice of the model determines the distributions the algorithm can realize, which are typically only a small subset of all possible probability distributions on $\mathcal{G}$. For example, let $\mathcal{G} = I\!R$ and $\lambda = 1$. Then the class of representable distributions may be given by the class of normal densities $\mathcal{N}(m, \sigma^2)$, where m is the expectation and σ^2 the variance. This defines a *parameterization* of the space of search distributions by mapping the parameters $(m, \sigma) \in I\!R^2$ into the set of probability distributions on $\mathcal{G}$. A global random search algorithm alters its search distribution on $\mathcal{G}$ by changing such parameters (step 3).

In evolutionary computation $\mathcal{G}$ corresponds to the genotype space. How is the search distribution on $\mathcal{G}$ realized in EAs? A number of genotypes $\tilde{\boldsymbol{g}}_1^{(t)}, \dots, \tilde{\boldsymbol{g}}_\mu^{(t)}$, the parents, are stored in a population. The superscript indicates the iteration of the algorithm, i.e., the generation. In each generation, λ offspring $\boldsymbol{g}_1^{(t)}, \dots, \boldsymbol{g}_\lambda^{(t)}$ are generated by applying variation operators. Let the probability that parents $\tilde{\boldsymbol{g}}_1^{(t)}, \dots, \tilde{\boldsymbol{g}}_\mu^{(t)} \in \mathcal{G}$ generate the offspring $\boldsymbol{g}_1^{(t)}, \dots, \boldsymbol{g}_\lambda^{(t)} \in \mathcal{G}$ be described by the conditional probability distribution $P_{\mathcal{G}^\lambda}\left(\boldsymbol{g}_1, \dots, \boldsymbol{g}_\lambda \,|\, \tilde{\boldsymbol{g}}_1^{(t)}, \dots, \tilde{\boldsymbol{g}}_\mu^{(t)}; \boldsymbol{\theta}^{(t)}\right)$. This distribution is additionally parameterized by some *external strategy parameters* $\boldsymbol{\theta}^{(t)} \in \Theta$, which may vary over time. Examples of such externally controlled parameters include the probabilities that certain variation operators are applied and parameters that determine the mutation strength (cf. Smith and Fogarty, 1997; Eiben et al., 1999; Igel and Kreutz, 2003). Thus, we have

$$P_{\mathcal{G}^\lambda}^{(t)}(\boldsymbol{g}_1, \dots, \boldsymbol{g}_\lambda) = P_{\mathcal{G}^\lambda}\left(\boldsymbol{g}_1, \dots, \boldsymbol{g}_\lambda \,|\, \tilde{\boldsymbol{g}}_1^{(t)}, \dots, \tilde{\boldsymbol{g}}_\mu^{(t)}; \boldsymbol{\theta}^{(t)}\right) \quad . \tag{1}$$

In some EAs, the joint distribution $P_{\mathcal{G}^\lambda}^{(t)}$ can be factorized as

$$P_{\mathcal{G}^\lambda}^{(t)}(\boldsymbol{g}_1, \dots, \boldsymbol{g}_\lambda) = P_{\mathcal{G}}^{(t)}(\boldsymbol{g}_1) \cdot \ldots \cdot P_{\mathcal{G}}^{(t)}(\boldsymbol{g}_\lambda) \quad . \tag{2}$$

This means, the offspring are created independently of each other based on the same distribution. In this case, the above scheme corresponds to global random search as described by Zhigljavsky (1991). In general though, $P^{(t)}_{\mathcal{G}^\lambda}$ can not be factorized as in (2), e.g., when crossover producing multiple offspring is employed or each parent generates exactly one offspring as in evolutionary programming (Fogel, 1995).[1] Nevertheless, the search distribution $P^{(t)}_{\mathcal{G}}$ can always be computed from $P^{(t)}_{\mathcal{G}^\lambda}$.

2.2. Genotypes and Phenotypes

In evolutionary systems, there is a phenotype associated with each individual (Mahner and Kary, 1997). The phenotype of an individual is the set of *all* its traits (including its fitness and its genotype). However, in most cases only some of the traits are of interest. A subset of traits of an individual is called its partial phenotype. In the following, we only deal with partial phenotypes and therefore omit the attribute “partial”.

Let $\mathcal{P}$ denote the set of all phenotypes under consideration. The function $\Gamma : \mathcal{G} \to \mathcal{P}$ that maps a genotype to its phenotype is called genotype-phenotype mapping. If Γ is not injective, i.e., multiple genotypes map to the same phenotype, we speak of a *neutral* encoding.

As the phenotype reflects the important properties of an individual, we are interested in the *search distribution on the phenotype space* $P^{(t)}_{\mathcal{P}}$, which describes the probability that a phenotype $p \in \mathcal{P}$ is sampled in generation t, rather than in the search distribution on $\mathcal{G}$. We call $P^{(t)}_{\mathcal{P}}$ the *exploration distribution*.

The genotype-phenotype mapping Γ *lifts* $P^{(t)}_{\mathcal{G}}$ from the genotype space onto the phenotype space:

$$\forall p \in \mathcal{P} : \; P^{(t)}_{\mathcal{P}}(p) = \sum_{\boldsymbol{g}' \in \Gamma^{-1}(p)} P^{(t)}_{\mathcal{G}}(\boldsymbol{g}') \,, \tag{3}$$

where $\Gamma^{-1}(p) = \{\boldsymbol{g}' \in \mathcal{G} \,|\, \Gamma(\boldsymbol{g}') = p\}$ is called the *neutral set* or *phenotypic class* of $p \in \mathcal{G}$ (Schuster, 1996; Mahner and Kary, 1997). Thus, the genotype population together with the variation operators, external parameters $\boldsymbol{\theta}$, and the genotype-phenotype mapping Γ can be regarded as parameters of the exploration distribution, thus inducing a parameterization of the space of distributions on $\mathcal{P}$.

We propose the entropy of $P^{(t)}_{\mathcal{P}}$ as a measure for the degree of exploration, i.e., for finite genotype spaces

$$H\left(P^{(t)}_{\mathcal{P}}\right) = -\sum_{p \in \mathcal{P}} P^{(t)}_{\mathcal{P}}(p) \log P^{(t)}_{\mathcal{P}}(p) \;\; . \tag{4}$$

Increasing exploration corresponds to an increase of $H\left(P_{\mathcal{P}}^{(t)}\right)$. This measure has the advantages that it incorporates all factors that control exploration (the population, the variation operators, …), that it is independent of an "extrinsic" distance measure on $\mathcal{P}$ (e.g., a canonical metric like the Hamming distance or the Euclidean distance on $\{0,1\}^n$ or $\mathbb{R}^n$, respectively), and that it allows comparison of the degrees of exploration across different algorithms.

We believe that the questions of how to parameterize and how to alter exploration distributions are among the most fundamental in evolutionary computation.

3. Self-Adaptation

We restate the common definition of self-adaptation in EAs and discuss its relation to neutrality. We propose an extension of this notion and give some instructive examples of what we call *classical* and *generalized* self-adaptation.

3.1. Classical Definition

The ability of an evolutionary algorithm to adapt its search strategy during the optimization process is a key concept in evolutionary computation (cf. Smith and Fogarty, 1997; Eiben et al., 1999; Igel and Kreutz, 2003). An online adaptation is important, because the best setting of an EA is in general not known *a priori* for a given task and a constant search strategy is usually not optimal during the evolutionary process. One way to adapt the search strategy online is self-adaptation (cf. Bäck, 1998), which is now used in all main paradigms of evolutionary computation, genetic algorithms (Schaffer and Morishima, 1987; Bäck, 1992), evolutionary programming (Fogel et al., 1995), evolution strategies (Schwefel, 1977; Rechenberg, 1994; Schwefel, 1995), and genetic programming (Angeline, 1996). This method can be described as follows (Eiben et al., 1999):

> "The idea of the evolution of evolution can be used to implement the self-adaptation of parameters. Here the parameters to be adapted are encoded into the chromosomes and undergo mutation and recombination. The better values of these encoded parameters lead to better individuals, which in turn are more likely to survive and produce offspring and hence propagate these better parameter values."

In other words, each individual does not only represent a candidate solution for the problem at hand, but also certain strategy parameters that are subject to the same selection process—they hitchhike with the object parameters. More formally:

DEFINITION 1 (self-adaptation). *The use of* strategy parameters *in order to adapt the exploration distribution* $P_{\mathcal{P}}^{(t)}$ *is called* (classical) self-adaptation. *Strategy parameters are parts of the genotype of individuals. Changes of these parts do not alter the phenotype of the corresponding individual but the search distribution* $P_{\mathcal{G}}^{(t)}$.

The attribute "classical" is used in order to distinguish between this definition and the notion of *generalized self-adaptation* defined in section 3.3.

The main property of self-adaptation compared to other strategy adaptation mechanisms is that in the case of self-adaptation the search strategy is subject to the evolutionary process itself and is not controlled externally.

3.2. Neutrality and Self-Adaptation

The adaptation of the search distribution on $\mathcal{P}$ is a fundamental concept in evolutionary computation. In principle, this can only be realized by varying

1. the genotype population $(\tilde{\boldsymbol{g}}_1^{(t)}, \dots, \tilde{\boldsymbol{g}}_\mu^{(t)})$, or
2. the external strategy parameters $\boldsymbol{\theta}^{(t)}$, or
3. the genotype-phenotype mapping Γ.

Apart from the population, which of these mechanisms to adapt $P_{\mathcal{P}}^{(t)}$ can be subject to the evolutionary process itself and allows for self-adaptation? The external parameters cannot, they are external by definition. Typically, neither can the genotype-phenotype mapping: From a biological point of view, one might associate environmental conditions (like the temperature, etc.) either with external parameters that vary the mutation and recombination probabilities or with the genotype-phenotype mapping. In most cases one would also not consider such conditions as subject to evolution.[2] Sometimes mechanisms that adapt the exploration distribution by genotype variations are regarded as examples of adaptive genotype-phenotype mappings. To our minds, this is a misleading point of view. For example, tRNA determines a *part* of the genotype-phenotype mapping and it is itself encoded in the genotype. However, the genotype-phenotype mapping should be understood as a *whole*—mapping all the genotype (including

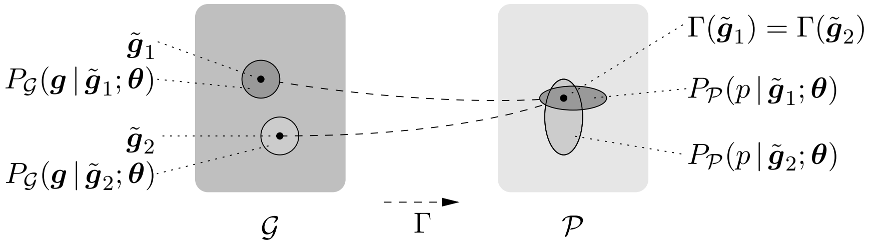
$\tilde{\boldsymbol{g}}_1$
$P_{\mathcal{G}}(\boldsymbol{g}\,|\,\tilde{\boldsymbol{g}}_1;\boldsymbol{\theta})$
$\tilde{\boldsymbol{g}}_2$
$P_{\mathcal{G}}(\boldsymbol{g}\,|\,\tilde{\boldsymbol{g}}_2;\boldsymbol{\theta})$
$\Gamma(\tilde{\boldsymbol{g}}_1) = \Gamma(\tilde{\boldsymbol{g}}_2)$
$P_{\mathcal{P}}(p\,|\,\tilde{\boldsymbol{g}}_1;\boldsymbol{\theta})$
$P_{\mathcal{P}}(p\,|\,\tilde{\boldsymbol{g}}_2;\boldsymbol{\theta})$
$\mathcal{G}$
Γ
$\mathcal{P}$

Figure 1. Two different points $\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2$ in $\mathcal{G}$ are mapped onto the same point in $\mathcal{P}$. The two elliptic ranges around the phenotype $\Gamma(\tilde{\boldsymbol{g}}_1) = \Gamma(\tilde{\boldsymbol{g}}_2)$ illustrate the corresponding exploration distributions by suggesting the range of probable mutants. Thus, the two points $\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2$ belong to one neutral set but represent two different exploration strategies.

the parts that code for the tRNA) to the phenotype, such that it becomes inconsistent to speak of genotypic information parameterizing the genotype-phenotype mapping. The same arguments also apply to evolutionary computation. In particular, self-adaptation as defined above does not alter the genotype-phenotype map.

Thus we consider only option 1 as a possibility to vary the exploration distribution in a way that is itself subject to evolution. However, if the genotype-phenotype mapping is injective, every variation of genotypes alters phenotypes and therefore bears the risk of a fitness loss. Hence, we conclude:

Given a constant genotype-phenotype mapping and in the absence of external control, only neutral genetic variations can allow an adaptation of the exploration distribution without changing the phenotypes in the population. Thereby, search distributions can be explored independently of phenotypic adaptation. For example, $P^{(t)}_{\mathcal{P}}$ *can change by neutral drift without the risk of loosing phenotypes with high fitness.*

Of course, the adaptation of exploration distributions is driven by selection, please see Toussaint (2003) for a thorough theoretical analysis of this process.

A *neutral genetic variation* means that parent and offspring have the same phenotype. For instance, consider two genotypes $\tilde{\boldsymbol{g}}_1$, $\tilde{\boldsymbol{g}}_2$ in a neutral set. Neglect crossover and assume that the probability for an offspring $\boldsymbol{g}$ of a single parent $\tilde{\boldsymbol{g}}_i$ is given by $P_{\mathcal{G}}(\boldsymbol{g}\,|\,\tilde{\boldsymbol{g}}_i;\boldsymbol{\theta})$. The two genotypes may induce two arbitrarily different exploration strategies around the same phenotype $p = \Gamma(\tilde{\boldsymbol{g}}_1) = \Gamma(\tilde{\boldsymbol{g}}_2)$, see figure 2. Transitions between these genotypes allow for switching the exploration strategy. In general, the variety of exploration distributions that can be explored in a neutral set $\Gamma^{-1}(p)$ is given by $\big\{P_{\mathcal{P}}(p\,|\,\tilde{\boldsymbol{g}}_i;\boldsymbol{\theta}) \mid \tilde{\boldsymbol{g}}_i \in \Gamma^{-1}(p)\big\}$.

3.3. Generalized Self-Adaptation

It is widely accepted that changing the genotypes without significantly changing the phenotypes in the population is a key search strategy in natural evolution (Huynen, 1996; Kimura, 1968; Schuster, 2002). We propose the following definition:

DEFINITION 2 (generalized self-adaptation). *Adaptation of the exploration distribution $P_{\mathcal{P}}^{(t)}$ by exploiting neutrality—i.e., independent of changing phenotypes in the population, of external control, and of changing the genotype-phenotype mapping—is called* generalized self-adaptation*.*

The classical concept of self-adaptation, see definition 1, where one distinguishes strategy parameters that typically control the mutation operators and object parameters describing fitness relevant traits, can be embedded in this definition.

Self-adapting the strategy parameters and keeping the object parameters constant corresponds to a move within a neutral set. But neutral sets exist even if there are no special, always neutral strategy parameters. Two or more loci on a chromosome may be phenotypically non-neutral but a specific simultaneous mutation of them might be neutral and may induce a change of the exploration distribution. Thus, movement within neutral sets is a general concept for self-adaptation, which is not bound to the idea of single loci being responsible for the search strategy. A good example of generalized self-adaptation without explicit strategy parameters is that topological properties of the search space (e.g., the set of most probable variations of a phenotype) may vary along such a neutral set if the genotype-phenotype mapping is chosen similar to grammars (Toussaint, 2001), which seems hard to realize by strategy parameters.

We believe that allowing for a self-adaptive exploration strategy is the main benefit of neutral encodings as stated by Shackleton et al.: “For redundancy to be of use it must allow for mutations that do not change the current phenotype, thus maintaining fitness, and which allow for moves to areas where new phenotypes are accessible”. There might be at least one additional case where neutrality can be useful, namely when it leads—by chance—to a bias in the search space, such that desirable phenotypes are represented more often than other ones (i.e., redundancy is not introduced in a “fair” manner, see section 4). An example where redundancy induces a negative bias is given by Igel and Stagge (2002).

Changing the exploration distribution corresponds to the ability of reaching new phenotypes. In this view, neutrality is not necessarily *redundant*: Different points in a neutral set can encode different exploration strategies and thus different information; a genotype encodes not only the information on the phenotype but also information on further explorations (Toussaint, 2003). One might put it like this: Although a non-injective genotype-phenotype mapping can be called redundant, the corresponding mapping from genotype to exploration distribution is in general non-redundant.

3.4. Examples

In the following, we give three examples that illustrate the concepts of classical and generalized self-adaptation.

3.4.1. *Codon Bias in Natural Evolution*

An interesting example of the interrelationship between neutrality and self-adaptation in nature has been investigated by Stephens and Waelbroeck (1999). They empirically analyze the codon bias and its effect in RNA sequences of the HI virus. In biology, several nucleotide triplets encode the same amino acid. Codon bias means that, although there exist several codons that code for the same amino acid (which form a neutral set), genotypes exhibit a preference on which codon is used to code for a specific amino acid. Let the neutral degree be defined as the fraction of neighbors of a codon that are mapped to the same amino acid, where two codons are said to be neighbored if they have Hamming-distance one. At some places of the HI virus' sequence codons are preferred that have a high neutral degree, whereas at other places codons are biased to have a low neutral degree. It is clear that these two cases induce different exploration distributions; the prior case means low mutability, the latter means high mutability. Stephens and Waelbroeck (1999) give an explanation for these two exploration strategies: Loci with low mutability cause "more resistance to the potentially destructive effect of mutation", whereas loci with high mutability might induce a "change in a neutralization epitope which has come to be recognized by the immune system." This can be regarded as an example of generalized self-adaptation without explicit strategy parameters.

3.4.2. *Self-Adaptation in Evolution Strategies*

In evolution strategies (ES) self-adaptation is employed to adapt (components of) the covariance matrix of the mutation operator (Schwefel, 1977; Rechenberg, 1994; Schwefel, 1995). An individual $\boldsymbol{g}_i^{(t)} \in I\!R^{n+s}$ can be divided into two parts, $\boldsymbol{g}_i^{(t)} = (\boldsymbol{x}_i^{(t)}, \boldsymbol{\sigma}_i^{(t)})$, the object variables $\boldsymbol{x}_i^{(t)} \in I\!R^n$ and the strategy parameters $\boldsymbol{\sigma}_i^{(t)} \in I\!R^s$. The latter are mapped to a covariance matrix of a Gaussian mutation distribution.

The fitness function depends only on the object variables. Thus, a reasonable definition of the phenotype is $\Gamma(\boldsymbol{g}_i^{(t)}) = \boldsymbol{x}_i^{(t)}$. This genotype-phenotype mapping is highly neutral, because typically $s = n$ or even $s = n(n+1)/2$. As (classical) self-adaptation in ES is widely accepted for improving the performance on real world optimization problems, this is a strong evidence for the benefits of neutrality in evolutionary computation.

3.4.3. *Introns in Genetic Programming*

In standard genetic programming (GP) the genotypes are expressions of a formal language. Variations are realized by randomly choosing a subexpression in the genotype and replacing it by another one. Banzhaf et al. (1998) define an intron in the context of GP as a feature of the genotype that emerges from the process of the evolution of variable length structures and that does not directly affect the survivability of the individual. Consider the following S-expression in prefix notation

$$T_1 = (\text{ IF } (< \ 2 \ 1 \ ) \ T_2 \ T_3 \ ) \tag{5}$$

as an example of a GP genotype, where T_2 and T_3 are arbitrary, but different S-expressions and the functions and terminals have the canonical semantics. If we define the phenotype of an individual by the functionality of the represented expression, then individuals with genotypes T_1 and T_3 are in the same phenotypic class or neutral set. T_1 without T_3 is called an intron in GP.

Obviously, replacing a genotype T_3 by T_1 changes the exploration distribution in several ways. The probability that the phenotype of the individual in question changes through a single variation is decreased, depending on the size of T_2, because each variation affecting T_2 does not alter the phenotype. It can be said that T_2 protects the information in T_3 against destructive variations (an effect that might explain bloat in GP, cf. Banzhaf et al., 1998). The exploration distribution can be narrowed by replacing T_2 within T_1 with an expression that consists of more subexpressions than T_2. In this way, altering T_2 controls the exploration distribution. Furthermore, the phenotype $\Gamma(T_2)$ may become more likely to evolve, because on the one hand its genetic material is available for crossover and on the other hand a slight change in the first argument of the IF expression in T_1 yields $\Gamma(T_2)$.

This is an example of generalized self-adaptation, where the adaptation of the search distribution is based on neutrality but takes place without the use of special strategy parameters.

4. Expected Hitting Time of Desirable Solutions

In the previous section, we gave examples of positive aspects of neutrality, e.g., in evolution strategies. However, one might argue that neutral encodings are *in general* disadvantageous, because they enlarge the search space (Radcliffe, 1991). In this section, we want to quantify this effect. Therefore, we derive the average time to find a desirable, say globally optimal, solution in NFL scenarios depending on the cardinality of the search space. This enables us to compare general properties of search algorithms on search spaces of different sizes.

We consider a finite search space $\mathcal{G}$ and a finite set of cost values $\mathcal{F}$ (the following considerations also hold when $\mathcal{F}$ is replaced by an arbitrary, finite phenotype space). Let $\mathcal{F}^{\mathcal{G}}$ be the set of all objective functions $f : \mathcal{G} \to \mathcal{F}$. Let the performance of an algorithm be the (average) number of iterations needed to find $\boldsymbol{g} \in \mathcal{G}$ such that $f(\boldsymbol{g})$ has a certain property, say, without loss of generality, is minimal. NFL theorems are concerned with non-repeating black-box search algorithms (deterministic and randomized ones), referred to simply as algorithms for brevity, that choose a new exploration point in the search space depending on the history of prior explorations (Wolpert and Macready, 1997; Schumacher et al., 2001; Igel and Toussaint, 2003). Let $\pi : \mathcal{G} \to \mathcal{G}$ be a permutation of $\mathcal{G}$. The set of all permutations of $\mathcal{G}$ is denoted by $\Pi(\mathcal{G})$. A set $F \subseteq \mathcal{F}^{\mathcal{G}}$ is said to be closed under permutation (c.u.p.) if for any $\pi \in \Pi(\mathcal{G})$ and any function $f \in F$ the function $f \circ \pi$ is also in F. It holds:

THEOREM 1 (Schumacher et al., 2001). *Any two algorithms a and b have the same performance averaged over F iff F is c.u.p.*

Now we ask: How long does an algorithm need to find a desirable solution on average and how does redundancy affect this time? The average number of evaluations needed to find an optimum depends on the cardinality of the search space $n = |\mathcal{G}|$ and the number of genotypes that are mapped to a desirable solution.

Let $F_m \subset \mathcal{F}^{\mathcal{G}}$ be the set of all functions where m elements in $\mathcal{G}$ are mapped to optimal solutions. It can easily be shown that $F_m \subset \mathcal{F}^{\mathcal{G}}$ is c.u.p. Because of theorem 1, the time to find an optimum averaged over all functions in F_m is the same for all algorithms. It holds:

THEOREM 2. *Given a search space of cardinality n, the expected number of evaluations $E\{T_{n,m}\}$ averaged over $F_m \subseteq \mathcal{F}^{\mathcal{G}}$ is equal to*

$$E\{T_{n,m}\} = \frac{n+1}{m+1} \quad . \tag{6}$$

Proof. The probability that the random variable $T_{n,m}$, representing the number of iterations needed to find an optimum, equals k is given by

$$P(T_{n,m} = k) \;=\; \frac{(n-m)!}{(n-m-k+1)!} \cdot m \cdot (n-k)! \cdot \frac{1}{n!} \tag{7}$$

$$= \binom{n-k}{m-1} \Big/ \binom{n}{m} \; . \tag{8}$$

There are $(n-m)!/(n-m-k+1)!$ non-optimal possibilities for the first $k-1$ explorations and m optimal ones for the k-th. The order of the remaining search points is arbitrary leading to the factor $(n-k)!$. The total number of orderings is $n!$. Non-distinguishable fitness values affect numerator and denominator in the same way and the corresponding factors can be canceled. The expectation of $T_{n,m}$ can be calculated, using basic rules for the manipulation of binomial coefficients (Graham et al., 1994), as follows:

$$\begin{aligned}
\mathrm{E}\{T_{n,m}\} &= \sum_{k=1}^{n-m+1} k \cdot P(T_{n,m} = k) = \binom{n}{m}^{-1} \cdot \sum_{k=0}^{n} k \binom{n-k}{m-1} \\
&= \binom{n}{m}^{-1} \cdot \sum_{k=0}^{n} (n+1-(n+1-k)) \binom{n-k}{m-1} \\
&= \binom{n}{m}^{-1} \cdot \sum_{k=0}^{n} \left[(n+1)\binom{n-k}{m-1} - m\binom{n+1-k}{m} \right] \\
&= \binom{n}{m}^{-1} \cdot \left[(n+1) \sum_{k=0}^{n} \binom{k}{m-1} - m \sum_{k=0}^{n+1} \binom{k}{m} \right] \\
&= \left[(n+1)\binom{n+1}{m} - m\binom{n+2}{m+1} \right] \Big/ \binom{n}{m} \\
&= \frac{(n+1)^2 - m(n+1)}{mn - m^2 + n + 1} = \frac{n+1}{m+1} \; . \qquad (9)
\end{aligned}$$

□

The expected number of function evaluations depending on n and m is visualized in figure 4. We get the intuitive result that if a single element is added to $\mathcal{G}$, then $\mathrm{E}\{T_{n,m}\}$ decreases or increases depending on whether the new element is mapped to an optimum or not.

Suppose we add redundancy to an encoding in an unbiased way, i.e., the ratio m/n of optimal solutions stays constant (e.g., we add

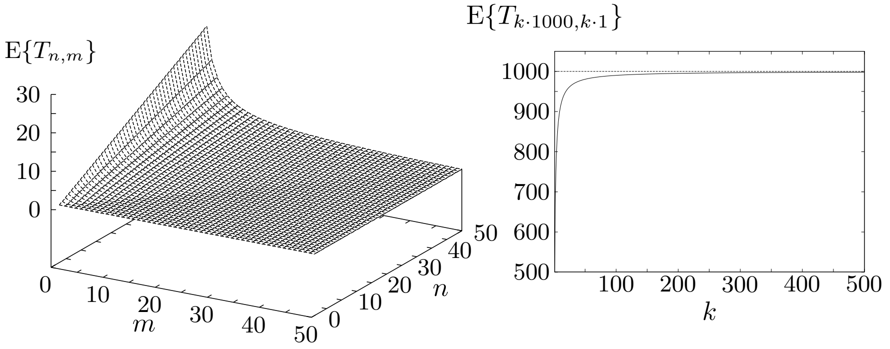

Figure 2. The left plot shows the average number of iterations needed to find an optimal solution depending on the size of the search space $|\mathcal{G}| = n$ and the number of genotypes mapping to an optimal solution m. The values for $m > n$ were set to one. The right plot displays the average number of iterations needed depending on the neutrality-factor k for $m = 1$ and $n = 1000$.

neutral strategy parameters to the genotype). This scenario appears to be suitable to analyze the effect of neutrality without a bias. It holds:

COROLLARY 1. *The average number of evaluations $E\{T_{k\cdot n,k\cdot m}\}$ needed to find a desirable solution increases with increasing* neutrality-factor *k and has the limit*

$$\lim_{k\to\infty} E\{T_{k\cdot n,k\cdot m}\} = \frac{n}{m} \quad . \tag{10}$$

For $k \geq 1$ it holds

$$E\{T_{(k+1)\cdot n,(k+1)\cdot m}\} - E\{T_{k\cdot n,k\cdot m}\} = \frac{n-m}{(1+km)(1+m+km)} > 0 \quad . \tag{11}$$

This means that in the considered NFL scenario enlarging the genotype space by adding redundancy without a bias does not considerably increase the average number of iterations needed to find a desirable solution if initially m is large enough. This follows from the existence of the limit (10), which is reached quickly as shown in the right plot in figure 4. Note that in (11) the numerator increases linearly with n, but the denominator increases quadratically with m. Hence, the larger the "initial" m, the smaller the effect of the additional genotypes. In the worst case, when initially only one element encodes a desirable solution, still the deterioration of the average search performance is bounded by a factor of two.

5. Conclusion

Neutrality is a necessity for self-adaptation. Actually, the design of neutral encodings to improve the efficiency of evolutionary algorithms is a well-established approach: strategy parameters are an example. Hence, there already exists clear evidence for the benefits of neutrality. The notion of neutrality provides a unifying formalism to embed approaches to self-adaptation in evolutionary computation. It is not restricted to the idea of self-adaptation using strategy parameters, where single genes are responsible for the exploration strategy. Generally, any local properties of the phenotypic search space—metric or topological—may vary along a neutral set. Thus, neutrality is not necessarily redundant: In general, a genotype encodes not only the information on the phenotype but also information on further explorations.

We derived the average number of fitness evaluations needed to find a desirable solution depending on the number of genotypes that encode a desirable solution and the cardinality of the search space. It turns out that the argument that neutrality is in general disadvantageous because it increases the search space has to be weakened as the effect of an unbiased introduction of neutrality on the average search performance in NFL scenarios can be neglected when the number of desirable genotypes is not too small.

Acknowledgements

The authors acknowledge support by the DFG under grant Solesys-II SCHO 336/5-2 and thank Stefan Wiegand for helpful remarks.

Notes

[1] In the case of crossover with multiple children, the offspring are not created independently of each other. When each parent generates exactly one offspring, as in evolutionary programming, the probability of each offspring depends on its individual parent. As parents need not have the same genotypes, the individual offspring distributions usually differ, i.e., cannot be described by one and the same distribution $P_{\mathcal{G}}^{(t)}$.

[2] There are counter-examples in natural evolution that go beyond the scope of our formalism, for instance, individuals whose behavior has an influence on mutations (e.g., sexual behavior influencing crossover) or on the embryonic development (e.g., drug abuse). All of these influences might be considered as subject to evolution.

References

Angeline, P. J.: 1996, 'Two Self-Adaptive Crossover Operations for Genetic Programming'. In: P. Angeline and K. Kinnear (eds.): *Advances in Genetic Programming*, Vol. 2. Cambridge, MA, MIT Press.

Bäck, T.: 1992, 'Self–Adaptation in Genetic Algorithms'. In: F. J. Varela and P. Bourgine (eds.): *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. pp. 263–271, MIT Press.

Bäck, T.: 1998, 'An Overview of Parameter Control Methods by Self-Adaptation in Evolutionary Algorithms'. *Fundamenta Informaticae* **35**(1-4), 51–66.

Banzhaf, W., P. Nordin, R. E. Keller, and F. D. Francone: 1998, *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, dpunkt.verlag.

Barnett, L.: 1998, 'Ruggedness and neutrality — The NKp family of fitness landscapes'. In: C. Adami, R. K. Belew, H. Kitano, and C. E. Taylor (eds.): *Alive VI: Sixth International Conference on Articial Life*. pp. 18–27, MIT Press.

Conrad, M.: 1990, 'The geometry of evolution'. *Biosystems* **24**, 61–81.

Ebner, M., P. Langguth, J. Albert, M. Shackleton, and R. Shipman: 2001, 'On Neutral Networks and Evolvability'. In: *Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001)*. pp. 1–8, IEEE Press.

Eiben, A. E., R. Hinterding, and Z. Michalewicz: 1999, 'Parameter Control in Evolutionary Algorithms'. *IEEE Transactions on Evolutionary Computation* **3**(2), 124–141.

Fogel, D. B.: 1995, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press.

Fogel, L. J., P. J. Angeline, and D. B. Fogel: 1995, 'An Evolutionary Programming Approach to Self-Adaptation on Finite State Machines'. In: J. R. McDonnell, R. G. Reynolds, and D. B. Fogel (eds.): *Proceedings of the Fourth Annual Conference on Evolutionary Programming*. pp. 355–365, MIT Press.

Graham, R. L., D. E. Knuth, and O. Patashnik: 1994, *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 2. edition.

Huynen, M. A.: 1996, 'Exploring phenotype space through neutral evolution'. *Journal of Molecular Evolution* **43**, 165–169.

Igel, C. and M. Kreutz: 2003, 'Operator Adaptation in Evolutionary Computation and its Application to Structure Optimization of Neural Networks'. *Neurocomputing*. In press.

Igel, C. and P. Stagge: 2002, 'Effects of Phenotypic Redundancy in Structure Optimization'. *IEEE Transactions on Evolutionary Computation* **6**(1), 74–85.

Igel, C. and M. Toussaint: 2003, 'On Classes of Functions for which No Free Lunch Results Hold'. *Information Processing Letters*. In press.

Kimura, M.: 1968, 'Evolutionary rate at the molecular level'. *Nature* **217**, 624–626.

Mahner, M. and M. Kary: 1997, 'What Exactly Are Genomes, Genotypes and Phenotypes? And What About Phenomes?'. *Journal of Theoretical Biology* **186**(1), 55–63.

Newman, M. E. J. and R. Engelhardt: 1998, 'Effects of neutral selection on the evolution of molecular species'. *Proceedings of the Royal Society of London, Series B: Biological Sciences* **265**(1403), 1333–1338.

Radcliffe, N. J.: 1991, 'Equivalence Class Analysis of Genetic Algorithms'. *Complex Systems* **5**, 183–205.

Rechenberg, I.: 1994, *Evolutionsstrategie '94*, Werkstatt Bionik und Evolutionstechnik. Stuttgart: Frommann-Holzboog.

Schaffer, J. D. and A. Morishima: 1987, 'An Adaptive Crossover Distribution Mechanism for Genetic Algorithms'. In: J. J. Grefenstette (ed.): *Proceedings of the Second International Conference on Genetic Algorithms (ICGA'87)*. Cambridge, MA, pp. 36–40, Lawrence Erlbaum Associates.

Schumacher, C., M. D. Vose, and L. D. Whitley: 2001, 'The No Free Lunch and Description Length'. In: L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke (eds.): *Genetic and Evolutionary Computation Conference (GECCO 2001)*. pp. 565–570, Morgan Kaufmann.

Schuster, P.: 1996, 'Landscapes and molecular evolution'. *Physica D* **107**, 331–363.

Schuster, P.: 2002, 'Molecular Insights into Evolution of Phenotypes'. In: J. P. Crutchfield and P. Schuster (eds.): *Evolutionary Dynamics—Exploring the Interplay of Accident, Selection, Neutrality and Function*, Santa Fe Institute Series in the Science of Complexity. Oxford University Press. In press.

Schwefel, H.-P.: 1977, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, Vol. 26 of *Interdisciplinary Systems Research*. Basel: Birkhäuser Verlag.

Schwefel, H.-P.: 1995, *Evolution and Optimum Seeking*, Sixth-Generation Computer Technology Series. New York: John Wiley & Sons.

Shackleton, M. A., R. Shipman, and M. Ebner: 2000, 'An investigation of redundant genotype-phenotype mappings and their role in evolutionary search'. In: A. Zalzala, C. Fonseca, J.-H. Kim, and A. Smith (eds.): *Proceedings of the International Congress on Evolutionary Computation (CEC 2000)*. pp. 493–500, IEEE Press.

Shipman, R.: 1999, 'Genetic Redundancy: Desirable or problematic for evolutionary adaptation'. In: M. A. Bedau, S. Rasmussen, J. S. McCaskill, and N. H. Packard (eds.): *Proceedings of the 4th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA)*. pp. 337–344, Springer-Verlag.

Smith, J. E. and T. C. Fogarty: 1997, 'Operator and parameter adaptation in genetic algorithms'. *Soft Computing* **1**(2), 81–87.

Stephens, C. R. and H. Waelbroeck: 1999, 'Codon Bias and Mutability in HIV Sequences'. *Journal of Molecular Evolution* **48**, 390–397.

Toussaint, M.: 2001, 'Self-adaptive exploration in evolutionary search'. Technical Report IRINI 2001-05, Institut für Neuroinformatik, Lehrstuhl für Theoretische Biologie, Ruhr-Universität Bochum, 44780 Bochum, Germany.

Toussaint, M.: 2003, 'On the evolution of phenotypic exploration distributions'. In: C. Cotta, K. De Jong, R. Poli, and J. Rowe (eds.): *Foundations of Genetic Algorithms 7 (FOGA VII)*. Morgan Kaufmann. In press.

Toussaint, M. and C. Igel: 2002, 'Neutrality: A Necessity for Self-Adaptation'. In: *IEEE Congress on Evolutionary Computation 2002 (CEC 2002)*. pp. 1354–1359, IEEE Press.

van Nimwegen, E., J. P. Crutchfield, and M. Huynen: 1999, 'Neutral Evolution of Mutational Robustness'. *Proceedings of the National Academy of Sciences* **96**, 9716–9720.

Weicker, K. and N. Weicker: 2000, 'Burden and Benefits of Redundancy'. In: W. N. Martin and W. M. Spears (eds.): *Foundations of Genetic Algorithms 6 (FOGA 6)*. pp. 313–333, Morgan Kaufmann.

Wilke, C. O.: 2001, 'Adaptive evolution on neutral networks'. *Bulletin of Mathematical Biology* **63**, 715–730.

Wolpert, D. H. and W. G. Macready: 1997, 'No Free Lunch Theorems for Optimization'. *IEEE Transactions on Evolutionary Computation* **1**(1), 67–82.

Zhigljavsky, A. A.: 1991, *Theory of global random search*. Kluwer Academic Publishers.