A Comprehensive View of Fitness Landscapes with Neutrality and Fitness Clouds

Leonardo Vanneschi[1], Marco Tomassini[2], Philippe Collard[3], Sébastien Vérel[3], Yuri Pirola[1], and Giancarlo Mauri[1]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione (D.I.S.Co.), University of Milan-Bicocca, Milan, Italy
[2] Computer Systems Department, University of Lausanne, Lausanne, Switzerland
[3] I3S Laboratory, University of Nice, Sophia Antipolis, France

Abstract. We define a set of measures that capture some different aspects of neutrality in evolutionary algorithms fitness landscapes from a qualitative point of view. If considered all together, these measures offer a rather complete picture of the characteristics of fitness landscapes bound to neutrality and may be used as broad indicators of problem hardness. We compare the results returned by these measures with the ones of negative slope coefficient, a quantitative measure of problem hardness that has been recently defined and with success rate statistics on a well known genetic programming benchmark: the multiplexer problem. In order to efficaciously study the search space, we use a sampling technique that has recently been introduced and we show its suitability on this problem.

1 Introduction

In the context of Evolutionary Algorithms (EAs), Neutrality of fitness landscapes has been widely studied in the last few years. Nevertheless, many contributions that have appeared use very different concepts and measures between them to express neutrality. For instance, in [11], Reidys and Stadler define the family of additive random landscapes where both neutrality and ruggedness of fitness landscapes can be tuned; in [14], Toussaint and Igel talk of the suitability of the design of neutral encodings to improve the efficiency of EAs; in [3], Collard et al. introduce the concept of synthetic neutrality and study its effects on the evolvability of Genetic Algorithms (GAs); in [22,20,21], Yu and Miller show that increasing the search space's size by artificially introducing neutral neighbors to some individuals, can help Cartesian Genetic Programming (GP) to navigate some restricted fitness landscapes, focusing on the choice of the representation and how it affects the amount of neutral neighborhood in a fitness landscape (these results have been recently criticized by Collins in [5]). If on the one hand this multiplicity of different concepts and formalisms has contributed to fortify the belief that neutrality plays an important role in the search process of EAs from many different points of view, on the other we think that uniformity in treating neutrality is missing and we fear that this may lead to ambiguous and sometimes confusing conclusions. In other words, we strongly agree with Geard [6] that *the way* in which neutrality is defined is crucial in determining its role and that the choice of different neutrality frameworks and formalizations may lead to different, and in some cases even conflicting, conclusions.

M. Ebner et al. (Eds.): EuroGP 2007, LNCS 4445, pp. 241–250, 2007.

One of the main goals of this paper is establishing a precise set of *neutrality measures*, each of which aimed at formalizing a particular aspect of fitness landscapes bound to neutrality. These measures (some of which already introduced in [16]) are: the *average neutrality ratio*, the *average Δ-fitness*, the *non-improvable* and "*non-worsenable*" *solutions ratio* and the *profitable* and *unprofitable mutations ratio*. Each of them is calculted on *neutral networks*. None of them brings a sufficient amount of information if considered alone, since each of them focus only on a particular feature of the landscape, but the joint analysis of all of them should allow us to have a rather complete picture of fitness landscapes, especially those related to neutrality. Furthermore, even though a bound between neutrality and problem difficulty has often been hypothesized, neutrality has never been presented together with other difficulty measures before, in order to check if the respective results are consistant between each other or not. In this paper, we compare the qualitative results returned by our neutrality measures with the quantitative results returned by the Negative Slope Coefficient (NSC) (a hardness measure that has recently been proposed in [18,15]) and we hope that the results returned by our neutrality measures may support and strengthen the ones of the NSC. For our empirical study, we use two different versions of the multiplexer problem, induced by two different sets of functional operators: {IF} and {NAND}. Finally, as discussed in [15], the shape and features of the boolean functions fitness landscapes make them hard to study by means of uniform random samplings and thus more sophisticated sampling methods are needed. In this paper we use a new, and more elaborate, sampling methodology of the search space and neighborhood that has been first defined in [16].

This paper is structured as follows: in section 2 we introduce some notions that will be used in this paper and we present NSC results for the two chosen instances of the multiplexer problem. Section 3 presents the view of neutrality features of these two landscapes, as offered by our quantitative neutrality measures. Finally, section 4 discusses the results, concludes the paper and offers hints for future research activity.

2 Definitions and Preliminary Results

Fitness Landscapes and Neutrality. Using a landscape metaphor to gain insight about the workings of a complex system originates with the work of Wright on genetics [19]. A simple definition of fitness landscape in EAs is a plot where the points in the horizontal plane represent the different individual genotypes in a search space (placed according to a particular *neighborhood relationship*) and the points in the vertical direction represent the fitness of each one of these individuals [9]. Generally, the neighborhood relationship is defined in terms of the genetic operators used [17,9,15]. This can be done easily for unary genetic operators like mutation, but it is clearly more difficult if binary or multi-parent operators, like crossover, are considered. Formal definitions of fitness landscape have been given (e.g. in [13]). Following these definitions, in this work a fitness landscape is a triple $\mathcal{L} = (\mathcal{S}, \mathcal{V}, f)$ where $\mathcal{S}$ is the set of all possible solutions, $\mathcal{V} : \mathcal{S} \to 2^{\mathcal{S}}$ is a neighborhood function specifying, for each $s \in \mathcal{S}$, the set of its neighbors $\mathcal{V}(s)$, and $f : \mathcal{S} \to \mathbb{R}$ is the fitness function. Given the set of variation operators, $\mathcal{V}$ can be defined as $\mathcal{V}(s) = \{s' \in \mathcal{S} | s'$ can be obtained from s by a single variation$\}$. In some cases, as for many GP boolean problems, even though the size of the search space $\mathcal{S}$ is huge, f can only assume a limited set of values. Thus, a large number of different solutions have the same fitness. In this case, we say that the landscape has a high degree of neutrality [11]. Given a solution s, a particular subset of $\mathcal{V}(s)$ can be defined: the one composed by neighbor solutions that are also neutral. Formally, the *neutral neighborhood* of s is the set $\mathcal{N}(s) = \{s' \in \mathcal{V}(s) | f(s') = f(s)\}$. The number of neutral neighbors of s is called the *neutrality degree* of s and the ratio between neutrality degree and cardinality of $\mathcal{V}(s)$ is the *neutrality ratio* of s. Given these definitions, we can imagine a fitness landscape as being composed by a set of (possibly large) *plateaus*. More formally, a *neutral network* [12] can be defined as a graph connected component $(\mathcal{S}, E_{\mathcal{N}})$ where $E_{\mathcal{N}} = \{(s_1, s_2) \in \mathcal{S}^2 | s_2 \in \mathcal{N}(s_1)\}$. Finally, we define the *fitness of a neutral network* (or *network fitness*) as the fitness value shared by all individuals of this neutral network.

Negative Slope Coefficient. Evolvability of a solution related to an operator [1] can be studied by plotting the fitness values of individuals against the fitness values of their neighbours. Such a plot has been presented in [4,2] and called *fitness cloud*. A possible algorithm to generate fitness clouds was proposed in [15]. This algorithm essentially corresponds to the sampling produced by a set of n stochastic hill-climbers at their first iteration after initialisation. The Negative Slope Coefficient (NSC) has been defined to capture with a single number some interesting characteristics of fitness clouds. It can be calculated as follows: the abscissas of a fitness cloud can be partitioned into a certain number of separate bins $\{I_1, I_2, \dots, I_m\}$. Let $X_1, X_2, \dots, X_m$ be the averages of the abscissas of the points contained in bins $I_1, I_2, \dots, I_m$, respectively, and let $Y_1, Y_2, \dots, Y_m$ be the averages of the ordinates of the points in $I_1, I_2, \dots, I_m$. The set of points (X_i, Y_i) can be seen as the vertices of a polyline, which effectively represents the "skeleton" of the fitness cloud. For each of the segments of this, we can define a *slope*, $S_i = (Y_{i+1} - Y_i)/(X_{i+1} - X_i)$. Finally, the negative slope coefficient is defined as $\text{NSC} = \sum_{i=1}^{m-1} min\ (0, S_i)$. The hypothesis proposed in [15] is that the NSC should classify problems in the following way: if NSC$= 0$, the problem is easy; if NSC< 0 the problem is difficult and the value of NSC quantifies this difficulty: the smaller its value, the more difficult the problem. The justification put forward for this hypothesis was that the presence of a segment with negative slope would indicate a bad evolvability for individuals having fitness values contained in that segment as neighbours would be, on average, worse than their parents. Pros and cons of this measure have been discussed in [18,15].

Genetic Operators and Neighborhood. Standard crossover or subtree mutation [8] generate very complex neighborhoods. In this paper, we consider a simplified version of the inflate and deflate mutation operators first introduced in [15,17] (also called structural mutation operators in those works): (1) *Strict deflate mutation*, which transforms a subtree of depth 1 into a randomly selected leaf chosen among its children. (2) *Strict inflate mutation*, which transforms a leaf into a tree of depth 1, rooted in a random operator and whose children are a random list of variables containing also the original leaf. (3) *Point terminal mutation*, that replaces a leaf with another random terminal symbol. This set of genetic operators (already introduced in [16] and called *Strict-Structural*, or *StSt*, mutation operators) is easy enough to study and provides enough exploration power to GP. For instance, *StSt* mutations present two important properties: (i) each mutation has an inverse: let M be the set of *StSt* mutation operators and let $\mathcal{S}$ be the set of all the possible individuals (search space). For each pair of individuals $(i, j) \in \mathcal{S}$, if an operator $m \in M$ exists such that $m(i) = j$, then an operator $m^{-1} \in M$ such that $m^{-1}(j) = i$ always exists; (ii) for each pair of solutions $(i, j) \in \mathcal{S}$, a sequence of mutations which transforms i into j exists. See [16] for the formal proofs of these properties. Thus, the associated graph $(\mathcal{S}, \mathcal{V})$ of fitness landscape is undirected (given the (i) property) and connected (given the (ii) property) graph.

The Multiplexer Problem. The goal of the k-multiplexer [8] problem is to design a boolean function with k inputs and one output. The first x of the k inputs can be considered as address lines. They describe the binary representation of an integer number. This integer chooses one of the 2^x $(= k - x)$ remaining inputs. The correct output for the multiplexer is the input on the line specified by the address lines. The terminals are the k variable inputs to the function. The fitness function of a GP individual E is calculated as the number of input data for which E does not return the same value as the target function. In this paper, the fitness values have always been normalized into the $[0, 1]$ range, by dividing them by 2^k, where k is the problem's order. Thus, from now on a solution with fitness equal to 0 represents an optimal solution, while 1 is the worst possible fitness value. In this paper, we have used two different sets of non-terminals: $\{\text{IF}\}$ (where $\text{IF}(x, y, z)$ is a ternary boolean function which returns y if x is *true* and z otherwise) and $\{\text{NAND}\}$. We have chosen these two sets because they are small enough to limit the cardinality of the search space but rich enough to represent some perfect solutions. These two sets of boolean operators induce two landscapes (indicated by $\mathcal{L}_{(k,h)}^{\{\text{IF}\}}$ and $\mathcal{L}_{(k,h)}^{\{\text{NAND}\}}$ from now on, where k is the problem order and h is the predetermined tree depth limit) the first of which is generally easy for GP, while the second is hard. This fact is confirmed by the experimental results shown in table 1, where the values of the success rate (*SR*) for three different mutation rates and NSC are reported for both landscapes. The success rate results have been obtained by executing 100 independent GP runs using the 6-multiplexer problem, maximum tree depth for the individuals equal to 6 for the landscape induced by $\{\text{NAND}\}$ and to 5 for the landscape induced by $\{\text{IF}\}$ (the choice of these values for the tree depths are motivated later), population of size 100, ramped half-and-half population initialization, tournament selection of size 10, *StSt* mutations as genetic operators. Only one *StSt* mutation operator has been applied with a certain probability p_m. 100 GP runs have been executed with $p_m = 0.95$ (column 2 of table 1), 100 separate runs have been executed with $p_m = 0.5$ (column 3)

Table 1. Values of the success rate for three different mutation rates and of the NSC for the 6-multiplexer problem using two different sets of operators to build the individuals. The fitness landscapes induced by these two sets of operators clearly have different difficulties for GP.

Set Of Operators	$SR(p_m = 0.95)$	$SR_{(p_m=0.5)}$	$SR_{(p_m=0.25)}$	NSC
$\{\text{IF}\}$	1	0.98	0.71	0
$\{\text{NAND}\}$	0	0	0	-0.21

and 100 further runs have been executed with $p_m = 0.25$ (column 4). The choice of the particular mutation operator has been done each time uniformly at random between the three *StSt* mutations. A run has been considered successful when an individual with a lower fitness than 0.15 has been found. The results related to the NSC reported in table 1 (column 5) have been obtained by generating a sample of 40000 individuals with the Metropolis-Hastings algorithm and, for each of them, a neighbor by applying one *StSt* mutation. Once again, the choice of the particular mutation operator to generate each neighbor has been done uniformly at random between the three *StSt* mutations.

Sampling Methodology. In [9] uniform random samplings have been used for studying boolean function landscapes. In [15] importance sampling techniques such as Metropolis and Metropolis-Hastings [10] have been proposed. Even though the resuls obtained were satisfactory for the purposes of those works, still those samples did not capture some important characteristics of the fitness landscape (see [16] for a detailed discussion). In this paper, we use a methodology aimed at generating samples containing trees of many (possibly all) different fitness values and forming connected neutral networks, if possible. This technique is composed by three steps: *modified Metropolis*, *vertical expansion* and *horizontal expansion*. Modified Metropolis generates a sample S of individuals with as many different fitness values as possible. The vertical expansion tries to enrich S by adding to it some *non-neutral* neighbors of its individuals. Finally, the horizontal expansion tries to enrich S by adding to it some *neutral* neighbors of its individuals. This methodology has been presented in [16] and it is not described here to save space.

3 Neutrality Results

In this section we present a study of neutrality of $\mathcal{L}^{\{\text{NAND}\}}_{(6,6)}$ and $\mathcal{L}^{\{\text{IF}\}}_{(6,5)}$, which are the largest search spaces respectively induced by $\{\text{NAND}\}$ and $\{\text{IF}\}$ that we have been able to study with our computational resources. The difference in the tree depth limit for the two landscapes is due to the fact that NAND is an operator with arity 2 while IF is an operator of arity 3. Thus, given a fixed tree depth, the trees that can be built with IF are on average larger than the ones that can be built with NAND. Figure 1 shows the fitness distributions (that is the frequence of fitness value) of the samples of $\mathcal{L}^{\{\text{NAND}\}}_{(6,6)}$ and $\mathcal{L}^{\{\text{IF}\}}_{(6,5)}$ that we have generated with Metropolis-Hasting sampling technique. For $\mathcal{L}^{\{\text{IF}\}}_{(6,5)}$, all the sampled fitness values are included into the range $[0, 0.7]$; in other words, no *bad* individual has been sampled. This is probably a characteristic of the complete search space (and it is not due to a bias of our samplig technique); in fact, we have exhaustively generated all the possible individuals of $\mathcal{L}^{\{\text{IF}\}}_{(3,2)}$ and we have observed that no tree with fitness larger than 0.7 exists also in that (similar, although much smaller) search space. On the other hand, for $\mathcal{L}^{\{\text{NAND}\}}_{(6,6)}$ large part of the sampled individuals have a bad fitness value (included into the range $[0.75, 1]$). Also this characteristic is analogous to what happens in the similar, but smaller, search space $\mathcal{L}^{\{\text{NAND}\}}_{(3,3)}$ that we have been able to exhaustively generate, where the largest number of individuals had fitness equal to 0.75 and the majority of the individuals had bad fitness values. Finally, we point out

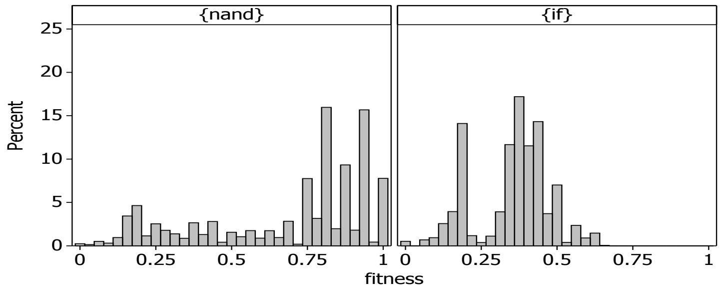

Fig. 1. Fitness distribution of $\mathcal{L}_{(6,6)}^{\{NAND\}}$ (left part) and $\mathcal{L}_{(6,5)}^{\{IF\}}$ (right part)

that with our sampling technique we have been able to generate individuals with many different fitness values, which is quite unusual for boolean landscapes (as pointed out, for instance, in [15]). Figure 2 reports the average neutrality ratio of neutral networks scatterplots for $\mathcal{L}_{(6,6)}^{\{NAND\}}$ (left part) and $\mathcal{L}_{(6,5)}^{\{IF\}}$ (right part) as a function of fitness value. The average neutrality ratio, $\bar{r}$ is defined as the mean of the neutrality ratios (as defined

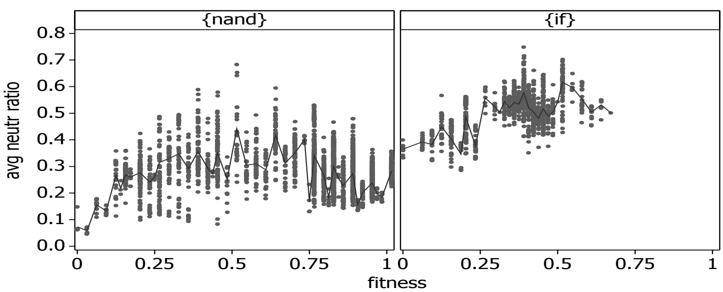

Fig. 2. Scatterplot of the average neutrality ratio in $\mathcal{L}_{(6,6)}^{\{NAND\}}$ (left part) and $\mathcal{L}_{(6,5)}^{\{IF\}}$ (right part)

in section 2) of all the individuals in a network. High values of $\bar{r}$ (near to 1) correspond to a large amount of neutral mutations. As figure 2 clearly shows, $\mathcal{L}_{(6,5)}^{\{IF\}}$ has a higher neutrality ratio than $\mathcal{L}_{(6,6)}^{\{NAND\}}$, in particular for networks at good fitness values. In other words, $\mathcal{L}_{(6,5)}^{\{IF\}}$ is "more neutral" than $\mathcal{L}_{(6,6)}^{\{NAND\}}$ in good regions of the fitness landscape. In this figure, as in all the subsequent ones, to guide the eye, a gray line is drawn, joining all the average points for each considered fitness value. These averages have been weighted according to the size of networks representing each point. Furthermore, points at the same coordinates have been artificially (slightly) displaced, so that they can be distinguished.

The second measure we study is the *average Δ-fitness* of the neutral networks. This measure is the average fitness gain (positive or negative) achieved after a mutation of the individuals belonging to the network. Formally, let N be a neutral network, then its *average* Δ-*fitness* can be defined as:

$$\Delta\bar{f}(N) := \frac{1}{|N|} \cdot \sum_{s \in N} \left[\frac{\sum_{v \in \mathcal{V}(s)} (f(v) - f(s))}{|\mathcal{V}(s)|} \right]$$

This measure is clearly related to the notions of evolvability [1] and innovation rate [7]. It also helps to statistically describe the graph $(\mathcal{S}, \mathcal{V})$. A negative value of $\Delta\bar{f}$ corresponds to a fitness improvement (because it reduces the error) while a positive one corresponds to a worsening (because it increases the error). The average Δ-fitness scatterplots are not reported here to save space, but we have studied them and we have observed that improving good individuals for $\mathcal{L}^{\{\mathrm{IF}\}}_{(6,5)}$ is easier than for $\mathcal{L}^{\{\mathrm{NAND}\}}_{(6,6)}$, in fact, for neutral networks at good fitness values, the value of the average Δ-fitness for $\mathcal{L}^{\{\mathrm{NAND}\}}_{(6,6)}$ is positive and much larger than the one for $\mathcal{L}^{\{\mathrm{IF}\}}_{(6,5)}$.

Now, we present two measures that we have called *Non Improvable (NI) Solutions ratio* and *Non Worsenable*[1] *(NW) Solutions ratio*. The first one is defined as the number of non-improvable solutions, or non-strict local optima (i.e. individuals i which cannot generate offspring j by applying a *StSt* mutation such that the fitness of j is better than the fitness of i) that are contained into a network divided by the size of the network. The second one is the ratio of the individuals i which cannot generate offspring j (by applying a *StSt* mutation) such that the fitness of j is worse than the fitness of i. The scatterplots of *NI* solutions ratios are reported in figure 3. $\mathcal{L}^{\{\mathrm{NAND}\}}_{(6,6)}$ presents some *NI*

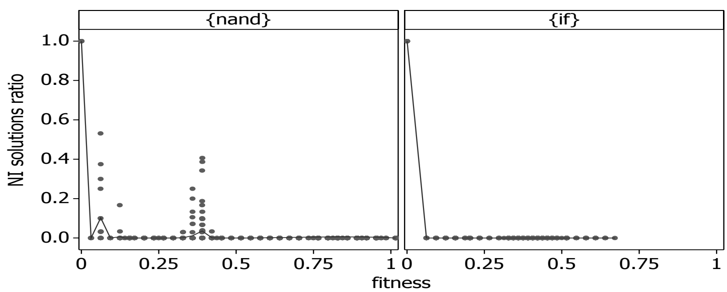

Fig. 3. Scatterplot of *NI* solutions ratio in $\mathcal{L}^{\{\mathrm{NAND}\}}_{(6,6)}$ (left part) and $\mathcal{L}^{\{\mathrm{IF}\}}_{(6,5)}$ (right part)

solutions ratios larger than 0.2 for some good fitness values (see for instance the peaks at fitness values approximately equal to 0.1, 0.125, 0.375). This indicates the presence of some trap neutral networks at this fitness values. This is not the case for $\mathcal{L}^{\{\mathrm{IF}\}}_{(6,5)}$ where *NI* solutions ratios are always equal to zero, except the obvious case of fitness equal to zero, where the *NI* solutions ratio is, of course, equal to one. In other words, for

[1] We are aware that the word "worsenable" does not exist in the English dictionary. Nevertheless we use it here as a contrary of "improvable", i.e. as something that cannot be worsened.

$\mathcal{L}^{\{NAND\}}_{(6,6)}$ some good individuals exist that cannot be improved by means of mutation, while this is not the case for $\mathcal{L}^{\{IF\}}_{(6,5)}$. *NW* solutions ratios scatterplot are not reported here to save space. Nevertheless, we have studied them and we point out that neutral networks in $\mathcal{L}^{\{IF\}}_{(6,5)}$ contain more *NW* solutions than for the ones in $\mathcal{L}^{\{NAND\}}_{(6,6)}$.

Figure 4 shows the scatterplot of unprofitable mutations ratios: for $\mathcal{L}^{\{NAND\}}_{(6,6)}$ and $\mathcal{L}^{\{IF\}}_{(6,5)}$: for each neutral network, we have calculated the number of mutations which do

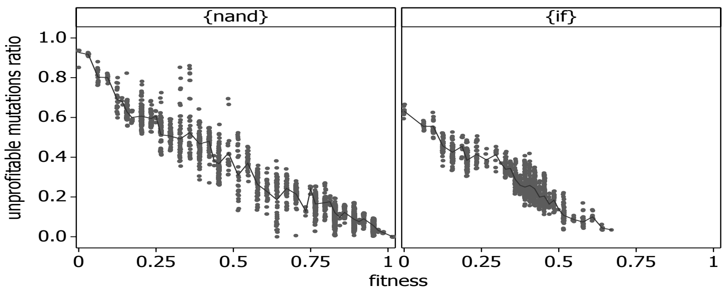

Fig. 4. Scatterplot of unprofitable mutations ratio in $\mathcal{L}^{\{NAND\}}_{(6,6)}$ (left part) and $\mathcal{L}^{\{IF\}}_{(6,5)}$ (right part)

not generate better offspring and divided it by the total number of possible mutations of the individuals in that network. Values of the unprofitable mutation ratios are higher for good fitness values in $\mathcal{L}^{\{NAND\}}_{(6,6)}$ than in $\mathcal{L}^{\{IF\}}_{(6,5)}$. In particular, for fitness values between 0 and 0.25, the majority of the possible mutations in $\mathcal{L}^{\{NAND\}}_{(6,6)}$ are unprofitable, while for $\mathcal{L}^{\{IF\}}_{(6,5)}$ only about half of the possible mutations appear to be unprofitable. All the neutrality measures that we have studied indicate that $\mathcal{L}^{\{IF\}}_{(6,5)}$ should be easier than $\mathcal{L}^{\{NAND\}}_{(6,6)}$ for GP. Furthermore, separate studies that we have done exhaustively generating all the possible individuals of the similar but smaller $\mathcal{L}^{\{IF\}}_{(3,2)}$ and $\mathcal{L}^{\{NAND\}}_{(3,3)}$ landscapes lead us to the same conclusions. Thus we hypothesize that our sampling technique is a suitable one to study our neutrality measures (i.e. the qualitative trends of our neutrality measures are kept as in the original complete landscape by our sampling technique).

4 Conclusions and Future Work

Some characteristics of fitness landscapes related to neutrality have been investigated in this paper for two different versions of the multiplexer problem. In particular, we have defined: the *average neutrality ratio*, the *average* Δ*-fitness*, the *non-improvable* and "*non-worsenable*" *solutions ratio* and the *profitable* and *unprofitable mutations ratio* of neutral networks. Each one of these measures, if considered alone, gives too a particular vision of the fitness landscape to allow us to draw strong conclusions about its difficulty. But considered all together, they have allowed us to have a clear and rather complete picture of the characteristics of multiplexer functions landscapes. In particular, all these measures have contributed to give an interpretation of the fact that the set of operators {IF} induces an easier fitness landscape than {NAND} for the multiplexer problem. This facts have also been experimentally demonstrated by executing 100 independent GP runs for each one of these problems and calculating the success rate. As a further confirmation, we have also calculated the value of another GP hardness indicator, called Negative Slope Coefficient. Another interesting result that we have obtained with our measures is that the landscapes induced by {IF} appear to be "more neutral" than the corresponding ones induced by {NAND}, in particular in correspondance of neutral networks with good (although not optimal) fitness values. In many recent contributions, a bound between neutrality and GP performance has been hypothesized and neutrality has been presented as a profitable [14,6,22] or unprofitable [5] characteristic of fitness landscapes. What may often be misleading in these discussions is, in our opinion, *what kind* of neutrality is being considered: many different ways of intending and formalizing the concept of neutrality may exist and each one of them may lead to different, and in some cases conflicting, conclusions. Our opinion is that, to study the relationship between neutrality and difficulty of a fitness landscape, a *pool* of neutrality measures is needed. All our results considered, we argue that our measures may be helpful in studying neutrality and relate it to GP problem hardness. Results shown in this paper hold both for "small" fitness landscapes, that we have been able to study by exhaustively generating all the individuals, and for "large" fitness landscapes, obtained by increasing the problem order and the maximum size of the individuals, and that we have sampled using a new methodology. This methodology is based on a modified version of the Metropolis algorithm, enriched by two further algorithms that we have called *vertical* and *horizontal* expansion. By this strategy, it has been possible to generate and to study a large number of individuals that would not (or would very rarely) have been generated by means of a uniform random sampling or a standard Metropolis algorithm. Since our techniques are general and can be used for any GP program space, future work includes extending this kind of study to other problems and possibly defining new measures of problem hardness based on neutrality.

References

1. L. Altenberg. The evolution of evolvability in genetic programming. In K. Kinnear, editor, *Advances in Genetic Programming*, pages 47–74, Cambridge, MA, 1994. The MIT Press.
2. L. Barnett. *Evolutionary Search on Fitness Landscapes with Neutral Networks*. PhD thesis, University of Sussex, 2003.
3. P. Collard, M. Clergue, and M. Defoin-Platel. Synthetic neutrality for artificial evolution. In *Artificial Evolution*, pages 254–265, 1999.
4. P. Collard, S. Verel, and M. Clergue. Local search heuristics: Fitness cloud versus fitness landscape. In R. L. D. Mántaras and L. Saitta, editors, *2004 European Conference on Artificial Intelligence (ECAI04)*, pages 973–974, Valence, Spain, 2004. IOS Press.
5. M. Collins. Finding needles in haystacks is harder with neutrality. In H.-G. Beyer *et al.*, editor, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 2, pages 1613–1618, Washington DC, USA, 25-29 June 2005. ACM Press.

6. N. Geard. A comparison of neutral landscapes – nk, nkp and nkq. In *Congress on Evolutionary Computation (CEC'02)*, Honolulu, Hawaii, USA, 2002. IEEE Press, Piscataway, NJ.
7. M. Huynen. Exploring phenotype space through neutral evolution. *J. Mol. Evol., 43:165–169*, 1996.
8. J. R. Koza. *Genetic Programming*. The MIT Press, Cambridge, Massachusetts, 1992.
9. W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, Berlin, Heidelberg, New York, Berlin, 2002.
10. N. Madras. *Lectures on Monte Carlo Methods*. American Mathematical Society, Providence, Rhode Island, 2002.
11. C. M. Reidys and P. F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2–3):321–350, 2001.
12. P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker. From sequences to shapes and back: a case study in RNA secondary structures. In *Proc. R. Soc. London B.*, volume 255, pages 279–284, 1994.
13. P. F. Stadler. Fitness landscapes. In M. Lässig and Valleriani, editors, *Biological Evolution and Statistical Physics*, volume 585 of *Lecture Notes Physics*, pages 187–207. Springer, Berlin, Heidelberg, New York, 2002.
14. M. Toussaint and C. Igel. Neutrality: A necessity for self-adaptation. In *Congress on Evolutionary Computation (CEC'02)*, pages 1354–1359, Honolulu, Hawaii, USA, 2002. IEEE Press, Piscataway, NJ.
15. L. Vanneschi. *Theory and Practice for Efficient Genetic Programming*. Ph.D. thesis, Faculty of Sciences, University of Lausanne, Switzerland, 2004.
16. L. Vanneschi, Y. Pirola, P. Collard, M. Tomassini, S. Verel, and G. Mauri. A quantitative study of neutrality in GP boolean landscapes. In M. Keijzer *et al.*, editor, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'06*, volume 1, pages 895–902. ACM Press, 2006.
17. L. Vanneschi, M. Tomassini, P. Collard, and M. Clergue. Fitness distance correlation in structural mutation genetic programming. In Ryan, C., *et al.*, editor, *Genetic Programming, 6th European Conference, EuroGP2003*, Lecture Notes in Computer Science, pages 455–464. Springer, Berlin, Heidelberg, New York, 2003.
18. L. Vanneschi, M. Tomassini, P. Collard, and S. Vérel. Negative slope coefficient. a measure to characterize genetic programming fitness landscapes. In Collet, P., *et al.*, editor, *Genetic Programming, 9th European Conference, EuroGP2006*, Lecture Notes in Computer Science, LNCS 3905, pages 178–189. Springer, Berlin, Heidelberg, New York, 2006.
19. S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In D. F. Jones, editor, *Proceedings of the Sixth International Congress on Genetics*, volume 1, pages 356–366, 1932.
20. T. Yu. "Six degrees of separation" in boolean function networks with neutrality. In R. Poli *et al.*, editor, *GECCO 2004 Workshop Proceedings*, Seattle, Washington, USA, 26-30 June 2004.
21. T. Yu and J. Miller. Neutrality and the evolvability of boolean function landscape. In J. Miller *et al.*, editor, *Proceedings of the Fourth European Conference on Genetic Programming (EuroGP-2001)*, volume 2038 of *LNCS*, pages 204–217, Lake Como, Italy, 2001. Springer, Berlin, Heidelberg, New York. Lecture notes in Computer Science vol. 2038.
22. T. Yu and J. F. Miller. Finding needles in haystacks is not hard with neutrality. In J. A. Foster *et al.*, editor, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 13–25. Springer-Verlag, 2002.