

# A Quantitative Study of Neutrality in GP Boolean Landscapes

Leonardo Vanneschi  
Dip. di Informatica S.Co.  
University of Milano-Bicocca  
Milan, Italy  
vanneschi@disco.unimib.it

Yuri Pirola  
Dip. di Informatica S.Co.  
University of Milano-Bicocca  
Milan, Italy  
yuri.pirola@gmail.com

Philippe Collard  
I3S Laboratory  
University of Nice  
Sophia Antipolis, France  
Philippe.COLLARD@unice.fr

Marco Tomassini  
Computer Systems Dept.  
University of Lausanne  
Lausanne, Switzerland  
Marco.Tomassini@unil.ch

Sébastien Verel  
I3S Laboratory  
University of Nice  
Sophia Antipolis, France  
verel@i3s.unice.fr

Giancarlo Mauri  
Dip. di Informatica S.Co.  
University of Milano-Bicocca  
Milan, Italy  
mauri@disco.unimib.it

## ABSTRACT

Neutrality of some boolean parity fitness landscapes is investigated in this paper. Compared with some well known contributions on the same issue, we define some new measures that help characterizing neutral landscapes, we use a new sampling methodology, which captures some features that are disregarded by uniform random sampling, and we introduce new genetic operators to define the neighborhood of tree structures. We compare the fitness landscape induced by two different sets of functional operators ( $\{\text{NAND}\}$  and  $\{\text{XOR}; \text{NOT}\}$ ). The different characteristics of the neutral networks seem to justify the different difficulties of these landscapes for genetic programming.

## Categories and Subject Descriptors

I.2 [ARTIFICIAL INTELLIGENCE]: Automatic Programming; D.2.8 [Software Engineering] Metrics – complexity measures, performance measures

## General Terms

Algorithms

## Keywords

Genetic Programming, Neutrality, Fitness Landscapes, Even Parity

## 1. INTRODUCTION

The role played by neutrality in determining the ability of evolutionary algorithms to find good quality solutions for a given problem has been a controversial issue in the last few years. A good introduction on the role of neutrality has been done by Reidys and Stadler in [10]. In [13], Toussaint and Igel claim the necessity of a certain degree of neutrality in fitness landscapes for self-adaptation. In [5], Geard compared the neutrality of some binary landscapes claiming a relationship between neutrality and performance of genetic algorithms (GAs). In [2], Collard *et al.* studied synthetic neutrality and its effects on the evolvability of GAs. The study of neutrality for genetic programming (GP) is mainly due to the work of Yu *et al.*: in [19], they showed that artificially introducing neutrality can help Cartesian GP to navigate some restricted fitness landscapes. These results have been recently criticized by Collins in [4]. Some other contributions on the importance of artificially introducing neutrality into fitness landscapes can be found in [17, 18, 20]. In this paper, we take up a different point of view. First of all, we study even parity fitness landscapes for standard tree based GP [7] instead of Cartesian GP. Secondly, instead of artificially introducing neutrality into the landscapes, we study them without modifying them, trying to infer some conclusions on the impact of neutrality on GP performance. Third, we introduce some new neutrality measures, such as the average neutrality ratio, the average  $\Delta$ -fitness of neutral networks and the ratio of some particular solutions contained into the neutral networks (see section 4 for the definitions of these measures). Fourth, we introduce some new genetic operators to define neighborhood between trees. Finally, instead of using a fixed set of functions to build solutions, we compare the landscapes induced by two different set of boolean operators ( $\{\text{NAND}\}$  and  $\{\text{XOR}; \text{NOT}\}$ ). Boolean spaces have already been studied, among others, in [3, 8]. In those contributions, either landscapes of small size have been studied exhaustively (i.e. taking into account all the possible solutions) or larger fitness landscapes have been studied by means of uniform random samplings. The shape and features of the boolean parity fitness landscapes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.  
Copyright 2006 ACM 1-59593-186-4/06/0007 ...\$5.00.

make them hard to study by means of uniform random samplings and thus more sophisticated sampling methodologies are needed. The first attempt to study them by means of some well known importance sampling techniques can be found in [14]. In this paper we define a new, and more elaborate, sampling methodology to study even parity fitness landscapes, but the techniques are general and can be used for any GP program space.

This paper is structured as follows: in section 2, we give some definitions that will be useful in the continuation of the paper. In section 3, we discuss some features of the even parity fitness landscapes. Section 4 contains an exhaustive investigation of a reduced size even parity landscape. Our new sampling technique is described in section 5. In section 6, we present some results on large even parity landscapes. Finally, in section 7, we offer our conclusions and ideas for future research.

## 2. PRELIMINARY DEFINITIONS

Using a landscape metaphor to develop insight about the workings of a complex system originates with [16]. 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 (ordered according to a particular *neighborhood relationship*) and the points in the vertical direction represent the fitness of each one of these individuals [8]. Generally, the neighborhood relationship is defined in terms of the genetic operators used to traverse the search space [15, 8, 14]. 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. Many formal definitions of fitness landscape have been given so far (see for instance [12]). 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,  $\mathcal{V} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$  is a neighborhood function specifying, for each  $s \in \mathcal{S}$ , the set of its neighbors  $\mathcal{V}(s)$ , and  $f : \mathcal{S} \rightarrow \mathbb{R}$  is the fitness function. Given the set of variation operators,  $\mathcal{V}$  can be defined as  $\mathcal{V}(s) = \{s' \in \mathcal{S} | s' \text{ can be obtained from } s \text{ by a single variation}\}$ . In some cases, as for the even parity 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 solutions have the same fitness. In this case, we say that the landscape has a high degree of neutrality [10]. Given a solution  $s$ , the subset of  $\mathcal{V}(s)$  composed by neighbor solutions that are also neutral can be defined. 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* [11] can be defined as a graph connected component  $(\mathcal{S}, E_N)$  where  $E_N = \{(s_1, s_2) \in \mathcal{S}^2 | s_2 \in \mathcal{N}(s_1)\}$ . The set of all neutral networks in the fitness landscape will be denoted by  $NN$  from now on. Replacing each solution with its neutral network, we can define a fitness landscape by means of its *neutrality graph*  $G_{NN} = (NN, E_{NN})$ , where  $E_{NN}$  is a relation such that, given two neutral networks  $N_1$  and  $N_2$ :  $(N_1, N_2) \in E_{NN} \Leftrightarrow \exists s_1 \in N_1, s_2 \in N_2 : s_2 \in \mathcal{V}(s_1) \cap \mathcal{N}(s_1)$ . Finally, we define the *fitness of a network* (or *network fitness*) as the fitness value shared by all its individuals. A neutral net with fitness  $\varphi$  will be indicated as  $\varphi$ -network.

## 3. THE EVEN PARITY PROBLEM

The goal of the even- $k$  parity problem [7] is to find a boolean function of  $k$  variables that returns True if an even number of inputs is True and False otherwise. Fitness is computed as  $2^k$  minus the number of hits over the  $2^k$  fitness cases represented by all the possible combinations of the  $k$  input values. Here, all the fitness values will be normalized dividing them by  $2^k$ . Thus a perfect individual has fitness 0, while the worst individual has fitness 1. The set of all possible solutions is composed by all the well-formed trees that can be built using a function set  $\mathcal{F}$  and a terminal symbols set  $\mathcal{T}$  and having a depth smaller or equal than a given limit. The set  $\mathcal{T}$  is composed by  $k$  variables (where  $k$  is the order of the problem). Two different function sets are studied in this work:  $\{\text{XOR}; \text{NOT}\}$  and  $\{\text{NAND}\}$ . The first one is not a generator set (i.e. it is impossible to generate all the different boolean functions using only this set), but it easily enables to obtain a perfect solution for the even parity problem. The second one is a minimal generator set, but it is difficult (even though not impossible) to build an optimal solution using it (see [9] for a more detailed discussion). We have chosen to use these function sets because they are small enough to limit the cardinality of the search space but also rich enough to represent some perfect solutions. Furthermore, these function sets induce two fitness landscapes with different difficulties for GP [14]: the landscape induced by  $\{\text{XOR}; \text{NOT}\}$  is easy to search, while the one induced by  $\{\text{NAND}\}$  is generally hard. Thus, we can compare the two landscapes (indicated with  $\mathcal{L}_{(k,h)}^{\{\text{XOR}; \text{NOT}\}}$  and  $\mathcal{L}_{(k,h)}^{\{\text{NAND}\}}$  from now on, where  $k$  is the problem order and  $h$  is the prefixed tree depth limit) to find some interpretations of their different difficulties.

To define a neighborhood structure based on variation operators, we have to choose a suitable set of these operators. Standard crossover or subtree mutation [7] generate neighborhoods which are too wide and complex to be studied. In this paper, we consider a simplified version of the inflate and deflate mutation operators first introduced in [14, 15] (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 (that will be called *Strict-Structural*, or *StSt*, mutation operators from now on) 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 and, (ii) for each pair of solutions, a sequence of mutations which transforms the first one into the second exists (see [9] for the formal proof). Thus, the associated graph  $(\mathcal{S}, \mathcal{V})$  of fitness landscape is undirected (given the (i) property) and connected (given the (ii) property) graph.

Some other interesting properties of the even parity fitness landscapes have been discussed, among others, in [8, 3]. First of all, supposing that all fitness values have been normalized into the range  $[0, 1]$ , if an expression does not contain at least an occurrence of each variable, then its fitness value is exactly equal to 0.5. For this reason, the wide

majority of individuals in the even parity landscapes have fitness 0.5 [14]. Secondly, an expression in the  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}$  landscape can only have a fitness value equal to 0, 0.5 or 1. (see for instance [8, 9] for the formal proofs of these properties). The choice of the *StSt* mutation operators permits to define some other properties of the  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}$  landscape: (a) there is *only* one neutral network at fitness 0.5 (we call it the *central network*), (b) all the other networks are composed by *one* single individual (we call them the *peripheral networks*) and (c) all the peripheral networks are connected with the central one by one mutation. The proofs of these properties are omitted for lack of space; they can be found in [9].

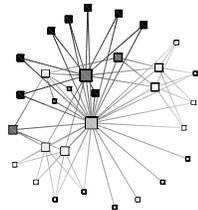
#### 4. EXHAUSTIVE ANALYSIS OF A “SMALL” EVEN-PARITY FITNESS LANDSCAPE

The first step of our study is to investigate a fitness landscape of small size, in order to be able to exhaustively generate all the possible individuals contained in it. We have built it by considering the even-2 parity problem and trees with a maximum depth equal to 3. The resulting  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$  and  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$  landscapes both contain at least one perfect solution. In table 1 some characteristics of these fitness landscapes are reported. In agreement with the theoretic-

**Table 1: Some characteristics of the “small” fitness landscapes that we have exhaustively studied**

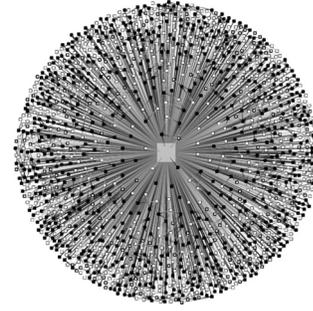
	$\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$	$\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$
No. of individuals	5552	1446
No. of optimal solutions	660	8
No. of neutral networks	1389	31
Average network size	3.99	46.64

cal observations presented in section 3,  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$  has a large number (1388) of neutral networks at fitness 0 and 1 composed by only one individual and one large (4164 individuals) central network at fitness 0.5. On the other hand,  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$  has smaller size and it has few networks, all of them medium-sized. Figures 1 and 2 report a graphical representation of the neutrality graphs of  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$  and  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$  respectively. Each square represents a neutral network, and



**Figure 1: Graphical representation of the neutrality graph of  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$ .**

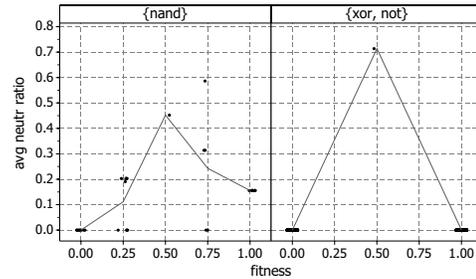
its size is proportional to the logarithm of the network size. The node colour indicates the fitness value of the network (from 1, black, to 0, white).



**Figure 2: Graphical representation of the neutrality graph of  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$ .**

#### 4.1 Experimental Results

The first parameter that we study is the *average neutrality ratio*,  $\bar{r}$ . It is defined as the mean of the neutrality ratios (as defined in section 2) of all the individuals included into a network. High values  $\bar{r}$  (near to 1) correspond to a large amount of neutral mutations. Figure 3 presents the scatterplot of  $\bar{r}$  against fitness in the two landscapes. In this figure,



**Figure 3: Scatterplot of the average neutrality ratio in  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$  (left part) and  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$  (right part).**

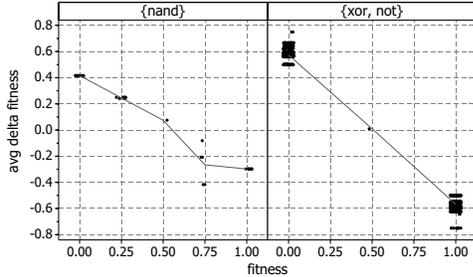
as in all the subsequent ones, a gray line is drawn, joining all the average points for each considered fitness value. This line should help readability. Furthermore, points at the same coordinates have been slightly displaced, so that they can be distinguished. In  $\mathcal{L}^{\{\text{XOR}; \text{NOT}\}}_{(2,3)}$ , the central network (fitness equal to 0.5) has high values of  $\bar{r}$ , while for the other networks  $\bar{r} = 0$ . Furthermore, the scatterplot is nearly symmetrical around fitness equal to 0.5. In  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$   $\bar{r}$  values are, on average, larger than 0.2 for some bad fitness values (fitness equal to 0.75) and smaller than 0.2 for good ones (fitness equal to 0 and 0.25): in general, in  $\mathcal{L}^{\{\text{NAND}\}}_{(2,3)}$  networks with bad fitness seem to be “more neutral” than networks with good fitness.

The second measure that we study is the *average  $\Delta$ -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  $\Delta$ -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 [6]. 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). As figure 4 shows, in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  the possible values of  $\Delta\bar{f}$  are included into a narrower range than in  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$ . We deduce that in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  mutations cannot



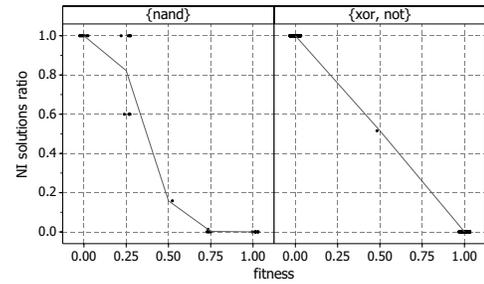
**Figure 4: Scatterplot of the average  $\Delta$ -fitness in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  (left part) and  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$  (right part).**

produce large fitness improvements (on average). Thus, to solve the problem, GP has to find individuals with many different fitness values. This is not the case for  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$ , where a mutation of an individual contained into the central network can produce an individual with a fitness equal to 0 (global optimum). Furthermore, in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  good fitness networks (fitness equal to 0.25 or 0.5) have positive values of  $\Delta\bar{f}$ . In other words, in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$ , it is unlikely that mutations of good individuals generate better offspring.

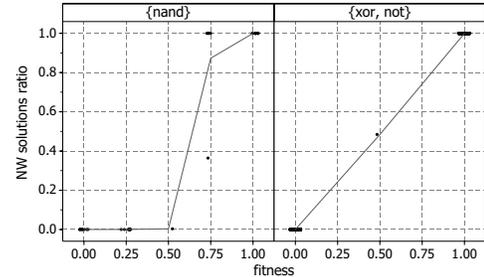
Now, we present two measures that we have called *Non Improvable (NI) Solutions ratio* ( $r_{ni}$ ) and *Non Worsenable*<sup>1</sup> (*NW Solutions ratio*) ( $r_{nw}$ ). The first one is defined as the number of non-improvable solutions, or non-strict local optima (i.e. individuals  $i$  which cannot 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  $i$ . Figures 5 and 6 present the scatterplots of  $r_{ni}$  and  $r_{nw}$  for each fitness value, respectively.

*NI* solutions ratio is 1 in 0-networks (they are composed of optimal solutions, so they cannot further improve) and it is 0 in 1-networks. Analogously, *NW* solutions ratio is 1 in 1-networks and it is 0 in 0-networks. In  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$ , there are some good networks (low fitness) with high  $r_{ni}$  values. At fitness 0.25, all the networks have an high value of  $r_{ni}$  (larger than 0.6) and 5 of them (over a total of 9 networks) have a value of  $r_{ni}$  equal to 1 and thus they are plateaus of non-strict local optima. We call these networks *trap networks*, since their individuals cannot generate better offspring and thus once GP has reached these networks, it cannot escape from them by means of a *StSt* mutation improving fitness. Trap networks do not exist in  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$ .

<sup>1</sup>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.

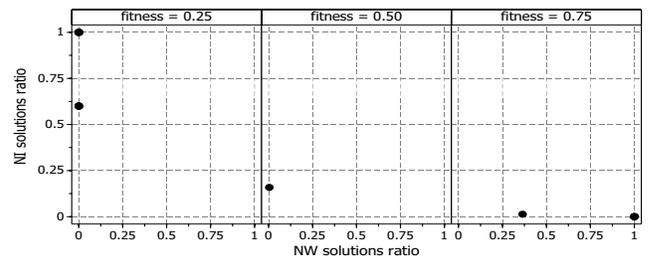


**Figure 5: Scatterplot of *NI* solutions ratio in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  (left part) and  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$  (right part).**



**Figure 6: Scatterplot of the *NW* solutions ratio in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  (left part) and  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$  (right part).**

Finally, we study the *NW* solutions ratio against *NI* for relevant<sup>2</sup> fitness values in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  (figure 7) and for all the possible fitness values in  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$  (figure 8). In  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$ ,



**Figure 7: Scatterplot of the *NW* solutions ratio vs. the *NI* solutions ratio in  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$ .**

all the points are disposed along the segment  $((1, 0), (0, 1))$ . In  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$ , the scatterplots are approximately parallel to the Cartesian axis and networks located at good fitness values have a large number of *NI* solutions. Thus, it is unlikely to mutate their individuals generating better offspring. This is not the case for  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$ . This may help explain why the  $\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$  landscape is hard for GP, while  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$  is easy.

<sup>2</sup>The scatterplot at fitness values equal to 1 and 0 are not reported for lack of space. However, they are obviously identical to the case of  $\mathcal{L}_{(2,3)}^{\{\text{XOR}; \text{NOT}\}}$  reported in figure 8.

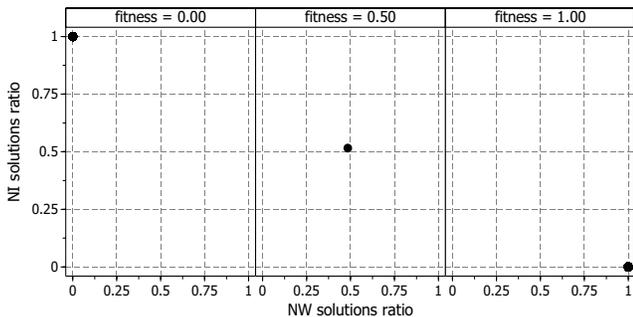


Figure 8: Scatterplot of the *NW* solutions ratio vs. the *NI* solutions ratio in  $\mathcal{L}_{(2,3)}^{\{XOR; NOT\}}$ .

## 5. SAMPLING METHODOLOGY

The even parity fitness landscape is a very hard one to sample. In fact, as the order  $k$  of the problem increases, the percentage of individuals with a fitness equal to 0.5 increases too. In [3, 8] some uniform random samplings for these spaces have been presented. In [14] some well known importance sampling techniques such as Metropolis and Metropolis-Hastings have been used. Even though the results obtained were satisfactory for the purposes of those works, still the large majority of the individuals had fitness equal to 0.5 and too few ones with different fitness were considered. Thus, those samples did not capture some important characteristics of the fitness landscape, such as the number and size of the neutral networks at fitness values different from 0.5, the connectivity of optimal solutions to these networks, etc. In other words, those samplings did not offer a useful “view” of the fitness landscapes and did not allow us to completely understand the behavior of GP on them. In this paper, we present a new methodology to generate samples containing trees of many (possibly all the) different fitness values and forming connected neutral networks, if possible. This technique is composed by three steps: we have called them *modified Metropolis*, *vertical expansion* and *horizontal expansion*. Modified Metropolis generates a sample  $S$  of individuals. 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.

### 5.1 Modified Metropolis Sampling

Our sampling methodology has been inspired by the Metropolis technique. According to that technique, a solution is generated at random at the beginning and considered as the *current* solution  $P$ . Successively, a loop is executed. At each iteration of that loop, a new solution  $T$  is generated at random and accepted (and thus inserted into the sample and considered as the new current solution  $P$ ) or rejected according to a certain probability distribution. In the Metropolis technique, the distribution for accepting or rejecting individuals is equal to  $\alpha_M(f(P), f(T)) = \min\left(1, \frac{f(P)}{f(T)}\right)$ , where  $f$  is the fitness function. In this way, the Metropolis technique favors fitter solutions but it does not penalize solutions at fitness 0.5. In our methodology, we define a probability distribution  $\alpha$  that rewards solutions with a different fitness than the previously accepted one. In this way, we intend to reward solutions with a different fitness than 0.5. Let  $p_m$  be the minimum probability of accepting a solution, then our

definition of the  $\alpha$  function is:

$$\alpha(f(P), f(T)) = (1 - p_m) \cdot \log_{10} \left( \frac{9 \cdot |f(P) - f(T)|}{\max(f(P), 1 - f(P)) + 1} + 1 \right) + p_m \quad (1)$$

In this way, if  $|f(P) - f(T)|$  is equal to 0, the new solution  $T$  gets a small probability (equal to  $p_m$ ) of being accepted<sup>3</sup>. On the other hand, the larger the value of  $|f(P) - f(T)|$ , the higher the probability. In particular, if  $T$  has the most different possible fitness value from  $P$ , then  $|f(P) - f(T)| = \max(f(P), 1 - f(P))$ . In that case, the logarithmic term becomes  $\log_{10}(9 + 1) = 1$ , and thus  $\alpha(f(P), f(T)) = 1$ . We have chosen a logarithmic function because it increases very quickly for small differences and thus it also rewards solutions  $T$  with a slightly different fitness from  $P$ .

### 5.2 Vertical Expansion

The vertical expansion of our methodology takes as input the sample  $S$  generated by the modified Metropolis algorithm and enriches it by adding some new individuals. In synthesis, it works as follows: for each individual  $i \in S$ ,  $L$  different neighbors of  $i$  are generated by means of  $L$  *StSt* mutations. Each one of these neighbors can be accepted or rejected according to the probability distribution expressed by equation (1). All accepted neighbors are finally inserted in  $S$  which is returned as output of the vertical expansion phase. Since the value of  $p_m$  is “small”, there is a “small” probability of having neutral neighbors in  $S$  at the end of the vertical phase. The goal of the horizontal phase is basically to add some neutral neighbors to the individuals in  $S$ . The neighbors which have been generated by the vertical expansion phase will represent the “links” (or “bridges”) between these neutral networks.

### 5.3 Horizontal Expansion

Let an *incomplete neutral network* be a sample  $I_N$  of a neutral network  $N$  such that at least one neutral neighbor  $j$  of an individual  $i \in I_N$  exists such that  $j \notin I_N$ . The horizontal expansion phase of our sampling technique takes as arguments the sample  $S$  returned by the vertical expansion phase, the minimum admitted size of an incomplete neutral network  $I_{max}$  and the maximum size of the sample that has to be generated  $S_{max}$ . These last two measures are parameters of our sampling technique and have to be manually defined. The horizontal phase returns a new sample  $S$ , possibly enriched by some individuals that form some neutral networks. It can be defined by the pseudo-code in figure 9, where  $\text{rnd}(0, 1)$  is a random number generated with uniform probability from the range  $(0, 1)$ ,  $iter$  is a variable containing the number of iterations that have been executed and  $\beta(f(i), f(j), iter)$  is defined as follows:

$$\beta(f(i), f(j), iter) = \begin{cases} 1 & \text{if } f(i) = f(j), \\ k^{-iter} & \text{otherwise} \end{cases}$$

where  $k$  is a constant that has to be chosen in such a way that probability  $\beta$  decreases “quickly enough” with iterations (in this work,  $k = 4$ ). Horizontal expansion stops when the sample reaches the maximum size  $S_{max}$  or when an iteration does not add any new individual. This phase

<sup>3</sup>If we set  $p_m = 0$  and  $f(P)$  is equal to 0.5, then the algorithm is likely to never terminate. Thus, a value of  $p_m$  larger than zero, even though “as small as possible”, has to be used.

```

iter ← 1;
while (at least one incomplete neutral network exists
in S) and (|S| < Smax) do
  N ← set of incomplete networks in S of size
  less than Imax;
  forall N ∈ N do
    forall i ∈ N do
      forall j ∈ V(i) do
        if (rnd(0,1) < β(f(i), f(j), iter)) and
(|S| < Smax) then
          S ← S ∪ {j};
        endif
      endforall
    endforall
  endforall
  iter ← iter + 1;
endwhile
return(S);

```

**Figure 9:** The pseudo-code describing the horizontal expansion of our sampling methodology.

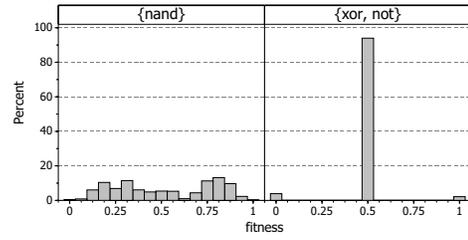
expands the size of neutral networks contained in the sample, by adding some new neutral neighbors to them. This is very useful to study neutrality, but it has some bias: for instance, a large neutral network could be represented in our sample by many smaller ones. It is the case, for instance, of the central network for the  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  landscape. However, in this particular case, this is not a problem, since we are aware about the unicity of the central network because of the theoretical results presented in section 3 (property (a)). Those theoretical results should contribute to understand the real shapes of the fitness landscapes under study.

## 6. ANALYSIS OF A “REALISTIC” FITNESS LANDSCAPE BY MEANS OF SAMPLING

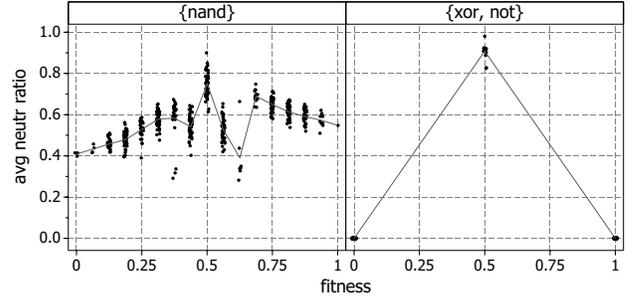
The largest search spaces that we have been able to study using our computational resources<sup>4</sup> correspond to the even-4 parity problems using trees of a maximum depth equal to 8. We indicate with  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  and  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  the landscapes using {XOR; NOT} and {NAND} as function sets respectively. Both these spaces contain optimal solutions for the even-4 parity. Nevertheless, for  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  they are difficult to automatically generate (either by a Metropolis algorithm or by GP). Thus, in order to sample all the possible fitness values, one of them has been manually added to the  $S$  sample that has been given as input to the vertical expansion phase. For the same reason, we have manually added to  $S$  an individual with the worst possible fitness (fitness equal to 1). Table 2 summarizes the parameters used to generate the samples of the two landscapes. Figure 10 monitors the fitness distributions and shows that our samples have covered the whole range of possible fitness values for the two landscapes.

In figure 11, we present the *average neutrality ratios* ( $\bar{r}$ ). The ratios calculated over the sample of  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  are not affected by the presence of multiple 0.5-networks (caused by the bias of our sampling methodology) instead of having only one central network: all the ratios of these networks are close to the “large” single one observed for the even-2 parity

<sup>4</sup>A PC 2.6 GHz with Pentium<sup>TM</sup> IV Processor, with 512MB central memory and 30GB hard disk.



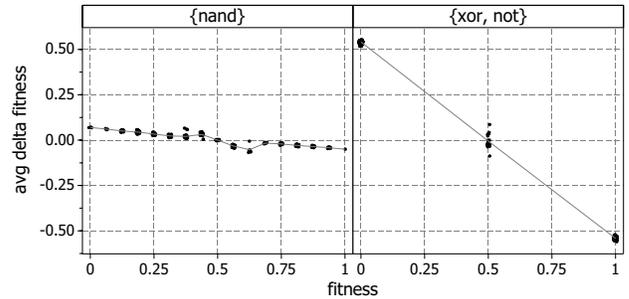
**Figure 10:** Fitness distribution of the sampled  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  (left part) and  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  (right part).



**Figure 11:** Scatterplot of the average neutrality ratio in the sampled  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  (left part) and  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  (right part).

(figure 3). Furthermore, as for  $\mathcal{L}_{(2,3)}^{\{NAND\}}$  (figure 3), also in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  the networks with good fitness values have a lower  $\bar{r}$  than ones with bad fitness values. The networks with good fitness in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  seem to be “less neutral” than ones with bad fitness.

The scatterplot of the *average  $\Delta$ -fitness* is shown in figure 12. In  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  this scatterplot reflects the behaviour



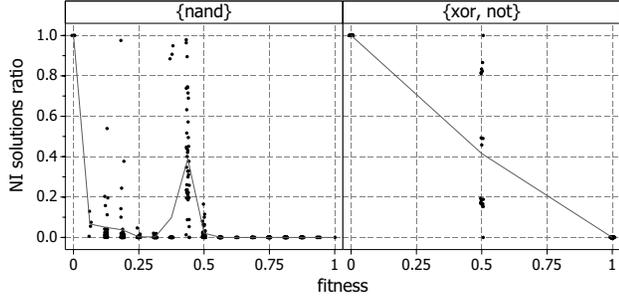
**Figure 12:** Scatterplot of average  $\Delta$ -fitness in the sampled  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  (left part) and  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  (right part).

observed for the even-2 parity (figure 4), whereas in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  it varies over a more limited range of values. Our interpretation is that to significantly improve a solution is more difficult in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  than in  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  because the majority of the mutations generate solutions with similar fitness. Thus the optimum in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  can be found by GP only generating individuals of many different fitness values, i.e. GP cannot perform “large jumps” as in  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$ .

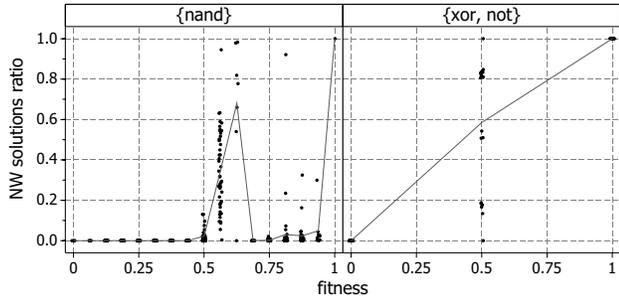
	{XOR; NOT}	{NAND}
$p_m$ for Modified Metropolis	0.005	
$p_m$ for vertical expansion	0.00005	
$k$ for horizontal expansion	4	
Minimal size of an incomplete network	2	
Sample size of Modified Metropolis	10	3
$L$ of vertical expansion	100	10
Size of generated sample	14589	30238
No. of networks contained into the sample	898	492

**Table 2: Parameters used to sample the  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  and  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  landscapes for the even-4 parity problem.**

The scatterplot of  $NI$  and  $NW$  solutions (figures 13 and 14 respectively) present some differences with respect to the ones observed for the landscape studied exhaustively (figures 5 and 6), especially for  $\mathcal{L}_{(4,8)}^{\{NAND\}}$ . Nevertheless, as in



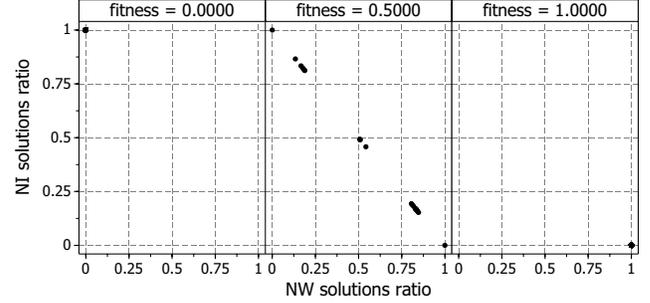
**Figure 13: Scatterplot of  $NI$  solutions ratio in the sampled  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  (left part) and  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  (right part).**



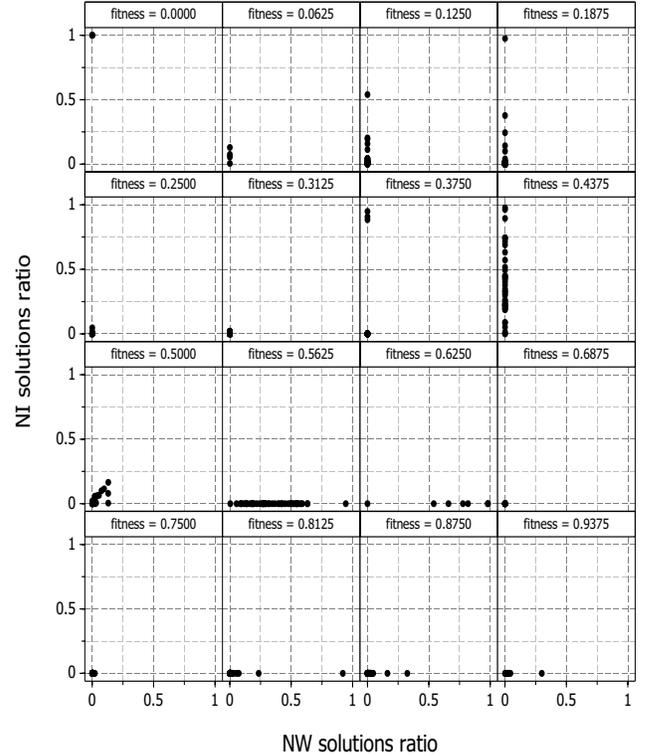
**Figure 14: Scatterplot of  $NW$  solutions ratio in the sampled  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  (left part) and  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  (right part).**

$\mathcal{L}_{(2,3)}^{\{NAND\}}$ , networks with good fitness contain a large number of  $NI$  solutions, which confirms that, in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$ , it is unlikely to mutate individuals belonging to good fitness neutral networks generating better offspring.

Figures 15 and 16 show the mutual correlation between  $NW$  and  $NI$  solutions ratios for  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$  and  $\mathcal{L}_{(4,8)}^{\{NAND\}}$  respectively. In the sample of  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$ , as in  $\mathcal{L}_{(2,3)}^{\{XOR; NOT\}}$ , the 0.5-networks are approximately disposed over the segment  $((0, 1), (1, 0))$ . In the sample of  $\mathcal{L}_{(4,8)}^{\{NAND\}}$ , as in  $\mathcal{L}_{(2,3)}^{\{NAND\}}$ , the scatterplots of networks with fitness values smaller than 0.5 are approximately parallel to the  $y$ -axis and the ones of networks with larger fitness values are approximately parallel to the  $x$ -axis. As for  $\mathcal{L}_{(2,3)}^{\{NAND\}}$  (section 4), also in  $\mathcal{L}_{(4,8)}^{\{NAND\}}$



**Figure 15: Scatterplot of  $NW$  vs.  $NI$  solutions ratio in the sampled  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$ .**



**Figure 16: Scatterplot of  $NW$  vs.  $NI$  ratio in the sampled  $\mathcal{L}_{(4,8)}^{\{NAND\}}$ .**

networks with good fitness values have a large number of  $NI$  solutions and thus it is unlikely to escape from them mutating its individuals, which is not the case for  $\mathcal{L}_{(4,8)}^{\{XOR; NOT\}}$ .

## 7. CONCLUSIONS AND FUTURE WORK

Some new characteristics of fitness landscapes related to neutrality have been defined in this paper and studied for different versions of the boolean parity problem. In particular, we have defined: (1) the average neutrality ratio of a neutral network, which quantifies the amount of possible neutral mutations of its individuals; (2) the average  $\Delta$ -fitness of a neutral network, which quantifies the average fitness gain achieved by mutating its individuals; (3) the non-improvable solutions ratio, which quantifies the amount of solutions that cannot generate better offspring in a neutral network; (4) the “non-worsenable” solutions ratio, which quantifies the amount of solutions that cannot generate worse offspring in a neutral network. Studying measure (1), we have observed that networks with bad fitness values seem to be “more neutral” than networks with good fitness values if {NAND} is used as the set of operators, while this is not the case if {XOR; NOT} is used. Studying measures (2), (3) and (4), we have observed that it is unlikely to improve fitness mutating individuals of neutral networks with good fitness values if {NAND} is used, which is not the case if we use {XOR; NOT}. These results may help explain why the even parity problem is easy for GP if we use {XOR; NOT} and hard if we use {NAND}. These results hold both for a “small” fitness landscape that we have been able to study exhaustively and for a “large” fitness landscape that we have sampled using a new methodology defined in this paper. This fact may suggest the suitability of our sampling methodology for the boolean parity problems. 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. Finally, studying neutrality induced by inflate and deflate mutations separately may provide some useful information about *bloat*.

## 8. REFERENCES

- [1] L. Altenberg. The evolution of evolvability in genetic programming. In K. Kinneer, editor, *Advances in Genetic Programming*, pages 47–74, Cambridge, MA, 1994. The MIT Press.
- [2] P. Collard, M. Clergue, and M. Defoin-Platel. Synthetic neutrality for artificial evolution. In *Artificial Evolution*, pages 254–265, 1999.
- [3] M. Collins. Counting solutions in reduced boolean parity. In R. Poli *et al.*, editor, *GECCO 2004 Workshop Proceedings*, Seattle, Washington, USA, 2004.
- [4] 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.
- [5] 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.
- [6] M. Huynen. Exploring phenotype space through neutral evolution, 1996.
- [7] J. R. Koza. *Genetic Programming*. The MIT Press, Cambridge, Massachusetts, 1992.
- [8] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, Berlin, Heidelberg, New York, Berlin, 2002.
- [9] Y. Pirola. Studio della neutralità degli spazi di ricerca booleani in programmazione genetica. Master’s thesis, Università di Milano-Bicocca, Milano, 2006.
- [10] C. M. Reidys and P. F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2–3):321–350, 2001.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] L. Vanneschi. *Theory and Practice for Efficient Genetic Programming*. Ph.D. thesis, Faculty of Science, University of Lausanne, Switzerland, 2004.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] T. Yu and J. F. Miller. The role of neutral and adaptive mutation in an evolutionary search on the onemax problem. In E. Cantú-Paz, editor, *Late Breaking Papers at the Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 512–519, New York, NY, July 2002. AAAI.