Defining Locality in Genetic Programming to Predict Performance

Edgar Galván-López, James McDermott, Michael O'Neill and Anthony Brabazon

Abstract—A key indicator of problem difficulty in evolutionary computation problems is the landscape's locality, that is whether the genotype-phenotype mapping preserves neighbourhood. In genetic programming the genotype and phenotype are not distinct, but the locality of the genotype-fitness mapping is of interest. In this paper we extend the original standard quantitative definition of locality to cover the genotype-fitness case, considering three possible definitions. By relating the values given by these definitions with the results of evolutionary runs, we investigate which definition is the most useful as a predictor of performance.

I. Introduction

The concept of a fitness landscape [1] has dominated the way geneticists think about biological evolution and has been adopted within the Evolutionary Computation (EC) community. In simple terms, a fitness landscape can be seen as a plot where each point on the horizontal axis represents all the genes in an individual corresponding to that point. The fitness of that individual is plotted as the height against the vertical axis. Thus, a fitness landscape is a representation of a search space which may contain peaks, valleys, hills and plateaus.

How an algorithm explores and exploits such a landscape is a key element of evolutionary search. Rothlauf [2], [3] has described and analysed the importance of locality in performing an effective evolutionary search of landscapes. Locality refers to how well neighboring genotypes correspond to neighboring phenotypes, and is useful as an indicator of problem difficulty. This research distinguished two forms of locality, low and high. A representation has high locality if all neighboring genotypes correspond to neighboring phenotypes, that is small genotypic changes result in small phenotypic changes. On the other hand, a representation has low locality if many neighboring genotypes do not correspond to neighboring phenotypes. It is demonstrated that a representation of high locality is necessary for efficient evolutionary search. In Section III we further explain the concept of locality.

In his original studies, Rothlauf used bitstrings to conduct his experiments [4] (and more recently he further explored the idea of locality using grammatical evolution at the chromosome level [5]). To our knowledge, there are few explicit studies on locality[1] using the typical Genetic Programming

Edgar Galván-López, James McDermott, Michael O'Neill and Anthony Brabazon are with the University College Dublin, Natural Computing Research & Applications Group, UCD CASL, 8 Belfield Office Park, Beaver Row, Clonskeagh, Dublin 4, email: `edgar.galvan, james.mcdermott2, m.oneill, anthony.brabazon@ucd.ie`.
[1]Initial preliminary results include [6].

(GP) [7], [8] representation (i.e., tree-like structures). For this purpose we will extend the definition of locality to GP, and due to the lack of distinction between genotype and phenotype, we will study the locality of the genotype-fitness mapping. The principle of strong causality states that for successful search, a small change in genotype should result in a small change in fitness [9]. The goal of this paper then is to shed some light on the type of locality present in GP and to clarify the correct extension of Rothlauf's genotype-phenotype locality to the genotype-fitness case. We use three different mutation operators, three different genotypic distance measures, and two problems with significantly different landscape features: the Artificial Ant Problem (a multimodal deceptive landscape) [10] and the Even-3-Parity Problem (a highly neutral landscape) [11].

This paper is organised as follows. In the next section, previous work on prediction of performance is summarised. In Section III locality in EC is presented. In Section IV, we describe how we study the locality of the genotype-fitness mapping in GP. In Section V, we present and discuss our findings. Finally, in Section VI we draw some conclusions.

II. Related Work

Landscapes and problem difficulty have been the subject of a good deal of research in EC in general and GP in particular. Several approaches to investigating problem difficulty have been proposed. In this section we mention some of them, including their pros and cons, which have inspired our work.

A. Fitness Distance Correlation

Jones [12], [13] proposed the *fitness distance correlation* (fdc) to measure the difficulty of a problem on the basis of the relationship between fitness and distance. The idea behind *fdc* was to consider fitness functions as heuristic functions and to interpret their results as indicators of the distance to the nearest optimum of the search space. *fdc* is an algebraic measure to express the degree to which the fitness function conveys information about distance to the searcher.

The definition of *fdc* is quite simple: given a set $F = \{f_1, f_2, ..., f_n\}$ of fitness values of n individuals and the corresponding set $D = \{d_1, d_2, ..., d_n\}$ of distances of such individuals from the nearest optimum, *fdc* is given by the following correlation coefficient:

$$fdc = \frac{C_{FD}}{\sigma_F \sigma_D},$$

where:

$$C_{FD} = \frac{1}{n} \sum_{i=1}^{n} (f_i - \overline{f})(d_i - \overline{d})$$

is the covariance of F and D, and σ_F, σ_D, $\overline{f}$ and $\overline{d}$ are the standard deviations and means of F and D, respectively. The n individuals used to compute *fdc* are obtained via some form of random sampling.

According to [13], [12] a problem can be classified in one of three classes, depending on the value of *fdc*:

1) *misleading* ($fdc \geq 0.15$), in which fitness tends to increase with the distance from the global optimum;
2) *difficult* ($-0.15 < fdc < 0.15$), for which there is no correlation between fitness and distance; and
3) *easy* ($fdc \leq -0.15$), in which fitness increases as the global optimum approaches.

The threshold interval [-0.15, 0.15] was empirically determined by Jones. In [13], [12], Jones also proposed to use scatter plots (distance versus fitness) when *fdc* does not give enough information about the hardness of a problem.

1) Comments on Fitness Distance Correlation: Altenberg [14] argued that predicting the hardness of a problem when using only fitness and distance in an EC system presents some difficulties. For instance, neither crossover nor mutation are taken into account when *fdc* is calculated. Other works have also shown some weaknesses on fdc [15], [16]. As it can be seen from the previous examples, *fdc* presents some weakness to predict the hardness of a given problem. Both [17] and [18] construct examples which demonstrate that the *fdc* can be "blinded" by particular qualities of the search space, and that it can be misleading. There is, however, a vast amount of work where Jones' approach has been successfully used in a wide variety of problems [19], [20], [21]. Of particular interest is the work by Vanneschi and colleagues [22], [17], [23], [24] which concentrated on the use of *fdc* in the context of GP.

B. Fitness Clouds and Negative Slope Coefficients

Later work by Vanneschi and colleagues attempted to address weaknesses of the *fdc* with new approaches. *Fitness clouds* are scatter plots relating fitness with distance to the optimum, as used in the *fdc*. Here, however, a more sophisticated approach to sampling the data (intended to model the real sampling behaviour of an evolutionary algorithm) and its analysis is used. The *negative slope coefficient* [25], [26] and the variant *fitness-proportional negative slope coefficient* [27] allow estimation of problem difficulty without requiring knowledge of the global optimum, making an *fdc*-style approach practical on real-world problems for the first time.

C. Other Landscape Measures

Several other approaches to studying landscapes and problem difficulty have also been proposed, generally in a non-GP context, including: other measures of landscape correlation [28], [29]; *epistasis*, which measures the degree of interaction between genes and is a component of *deception* [18]; *monotonicity*, which is similar to *fdc* in that it measures how often fitness improves despite distance to the optimum increasing [18]; and *distance distortion* which relates overall distance in the genotype and phenotype spaces [4]. All of these measures are to some extent related. However, Rothlauf's *locality* approach, again related to all of the ideas above, is of particular interest and so is treated separately in the next section.

III. LOCALITY

Understanding of how well neighbouring genotypes correspond to neighbouring phenotypes is a key element in understanding evolutionary search [4], [3]. In the abstract sense, a mapping has *locality* if neighbourhood is preserved under that mapping[2]. In EC this generally refers to the mapping from genotype to phenotype. This topic is a worthy of study because if neighbourhood is not preserved, then the algorithm's attempts to exploit the information provided by an individual's fitness will be misled when the individual's neighbours turn out to be very different.

Rothlauf's work on locality [4], [3], [31] has shed new light on several problems. Rothlauf gives a quantitative definition: "the locality d_m of a representation can be defined as

$$d_m = \sum_{d^g(x,y)=d^g_{\min}} |d^p(x,y) - d^p_{\min}|$$

where $d^p(x, y)$ is the phenotypic distance between the phenotypes x_p and y_p, $d^g(x, y)$ is the genotypic distance between the corresponding genotypes, and $d^p_{\min}$ resp. $d^g_{\min}$ is the minimum distance between two (neighboring) phenotypes, resp. genotypes" [4] [p. 77; notation changed slightly]. Locality is thus seen as a continuous property rather than a binary one. The point of this definition is that it provides a single quantity which gives an indication of the behaviour of the genotype-phenotype mapping and can be compared between different representations. Note that the quantity d_m is a measure of phenotypic divergence, so it is low for situations of high locality, and vice versa.

It can be stated that there are two types of locality: low and high locality. A representation is said to have the property of high locality if all neighboring genotypes correspond to neighboring phenotypes. On the other hand, a representation has low locality if some neighboring genotypes do not correspond to neighboring phenotypes. Rothlauf claims that a representation that has high locality will be more efficient at evolutionary search. If a representation has high locality then any search operator has the same effects in both the genotype and phenotype space. It is clear then that the difficulty of the problem remains unchanged compared to an encoding in which no genotype-phenotype map is required.

This, however, changes when a representation has low locality. To explain how low locality affects evolution, Rothlauf considered three different categories, taken from the work presented in [12] and explained previously. These are:

- *easy*, in which fitness increases as the global optimum approaches,

---

[2]The term *locality* has also been used in an unrelated context, to refer to the quasi-geographical distribution of an EC population [30].

- *difficult*, for which there is no correlation between fitness and distance and,
- *misleading*, in which fitness tends to increase with the distance from the global optimum.

If a given problem lies in the first category (i.e., easy), a low-locality representation will change this situation by making it more difficult and now, the problem will lie in the second category. This is due to low locality randomising the search. This can be explained by the fact that representations with low locality lead to uncorrelated fitness landscapes, so it is difficult for heuristics to extract information.

If a problem lies in the second category, a low-locality representation does not change the difficulty of the problem. There are representations that can convert a problem from difficult (class two) to easy (class one). However, to construct such a representation typically requires an understanding of the landscape equivalent to solving the problem.

Finally, if the problem lies in the third category, a representation with low locality will transform it so that the problem will lie in the second category. That is, the problem is less difficult because the search has become more random. As can be seen, this is a mirror image of a problem lying in the first category and using a representation that has low locality.

Although Rothlauf does not provide a threshold value to distinguish high and low locality, nevertheless it is possible to make relative comparisons between representations.

Note in particular Rothlauf's treatment of neutrality. When distinct (but neighbouring) genotypes map to identical phenotypes, a quantity ($d_{\min}$) is *added* to the sum. This is the same quantity that is added when neighbouring genotypes diverge slightly (for bitstring phenotypes with hamming distance). That is, neutrality is regarded as a deviation from good locality. Whether neutrality is beneficial in general is a complex question and some works have shed light on this issue [19], [20], [21], [32], [33], [34], but this issue deserves consideration as we will see.

We now consider the correct way to define the locality of the genotype-fitness mapping. Recall that locality, in general, is the property of preservation of neighbourhood under a mapping. Rothlauf assumes that a distance measure exists on both genotype and phenotype spaces, that for each there is a minimum distance, and that neighbourhood can be defined in terms of minimum distance. In standard GP, there are no phenotypes distinct from genotypes. It is common therefore to study instead the behaviour of the mapping from genotype to fitness [35], and we take this approach here. We will also regard two individuals to be neighbours in the genotype space if they are separated by a single mutation. That is, a mutation operator defines the neighbourhood of individuals at the genotype space, adhering to the Jones [12] principle that each operator induces its own landscape. We will also study very large spaces of GP-style trees, and therefore we must sample the space rather than enumerate it.

However, the most difficult issue in this context is how to define neighbourhood in the fitness space. There are three possibilities:

- The most straightforward extension of Rothlauf's definition might regard two individuals as neighbours if the difference of their fitness values is 1.[3] This leads to the following definition which we call *the $fd_{\min} = 1$ definition of locality*.

$$d_m = \frac{\sum_{i=1}^{N} |fd(x_i, m(x_i)) - fd_{\min}|}{N} \quad (1)$$

where $fd(x_i, m(x_i)) = |f(x_i) - f(m(x_i))|$ is the fitness distance between a randomly-sampled individual x_i and the mutated individual $m(x_i)$, $fd_{\min} = 1$ is the minimum fitness distance between two individuals, and N is the sample size.

- However, the above definition treats a fitness-neutral mutation as being just as bad for locality as a mutation causing a fitness divergence of two fitness units (assuming integer-valued fitness). It might be preferable to redefine the minimum distance in the fitness space as zero, giving the same locality definition as above but with $fd_{\min} = 0$. This we term *the $fd_{\min} = 0$ definition of locality*.

- Finally, it might be better to treat only true divergence of fitness as indicating poor locality. Therefore we might say that fitness divergence occurs only when the fitness distance between the pair of individuals is 2 or greater: otherwise the individuals are regarded as neighbours in the fitness space. This leads to the following definition, which we will call *the conditional definition of locality*).

$$d_m = \frac{\sum_{i=1: fd(x_i, m(x_i)) \geq 2}^{N} fd(x_i, m(x_i))}{N} \quad (2)$$

where $fd_{\min} = 1$.

Since we have no *a priori* reason to decide which of these three is the best definition of genotype-fitness locality, we will decide the issue by relating the values produced by each with performance achieved on EC runs.

IV. EXPERIMENTAL SETUP

For our analysis, we have used two well-known difficult problems for GP: the Artificial Ant Problem [7] (which has been shown to have multimodal deceptive features [10, Chapter 9]) and the Even-3-Parity problem (a problem that is difficult if no bias favorable is added in any part of the algorithm). To see and compare the locality present on these problems, we have decided to use two different function sets (see Table I for a description of them) for each of the problems used in this work. The idea here is that each function set will give a different locality and different performance.

The first problem, the Artificial Ant Problem [7, pp. 147–155], consists of finding a program that can successfully navigate an artificial ant along a path of 89 pellets of food on a 32 x 32 toroidal grid. When the ant encounters a

---

[3]Notice that in this work we are using problems of discrete values and so, it is reasonable to adopt this notion of fitness distance.
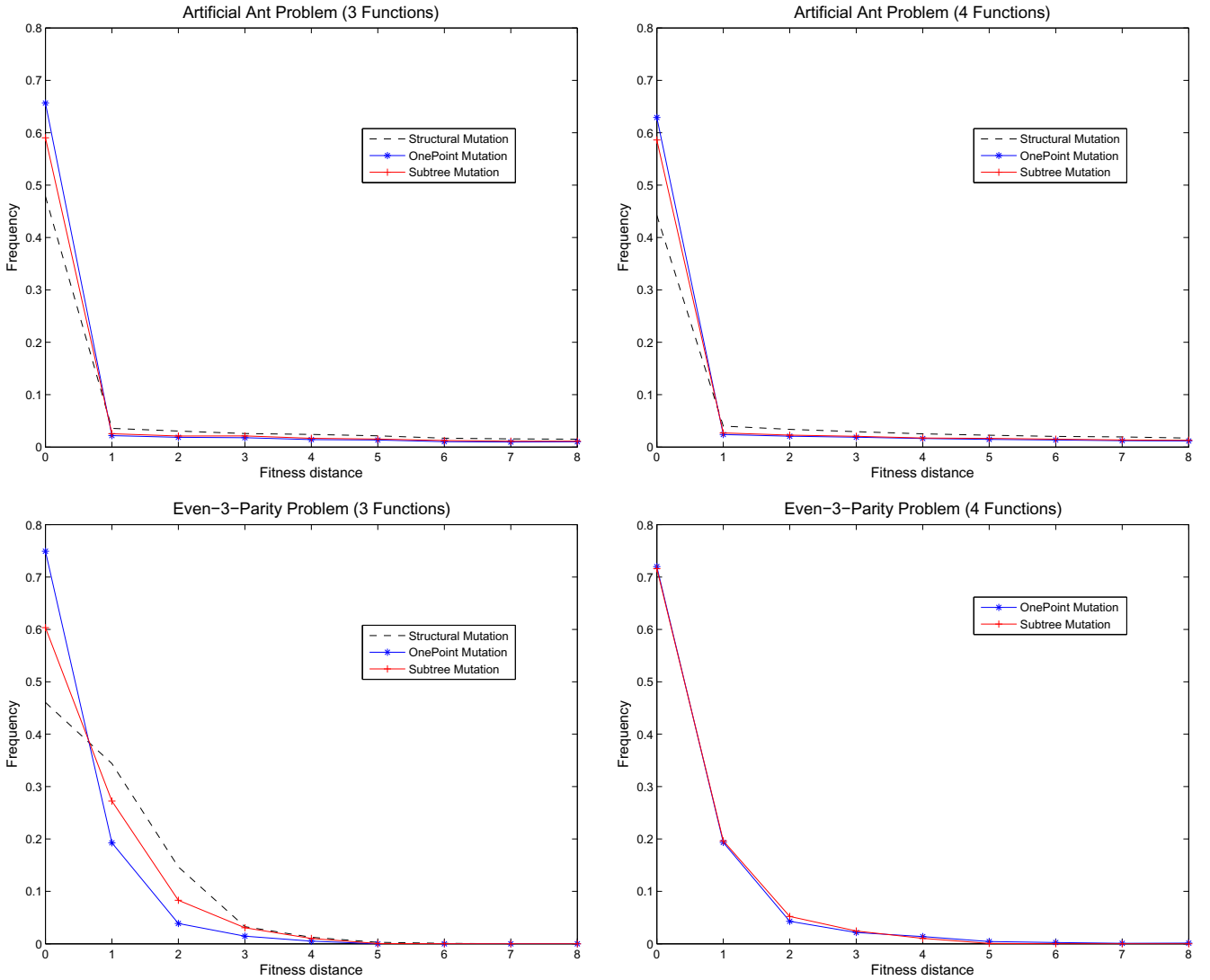
Fig. 1. Distribution of fitness distance values on the Artificial Ant Problem (top) and on the Even-3-Parity problem (bottom) using structural, one-point and subtree mutation, with standard and alternative function sets (left and right). For clarity we focus on the first 9 fitness distance values for the Artificial Ant Problem. Note also that structural mutation cannot be applied when using the 4-function set since all functions are of the same arity.

TABLE I

FUNCTION SETS USED ON THE ARTIFICIAL ANT AND THE EVEN-3-PARITY PROBLEM.

Number of functions	Artificial Ant Problem	Even-3-Parity
3 Functions	$F_{A3} = \{IF, PROG2, PROG3\}$	$F_{E3} = \{NOT, AND, OR\}$
4 Functions	$F_{A3} = \{IF, PROG2, PROG3, PROG4\}$	$F_{E4} = \{AND, OR, NAND, NOR\}$

food pellet, its (raw) fitness increases by one, to a maximum of 89. The problem is in itself challenging for many reasons. The ant must eat all the food pellets (normally in 600 steps) scattered along a twisted track that has single, double and triple gaps along it. The terminal set used for this problem is $T = \{Move, Right, Left\}$. The standard function set is $F_{A3} = \{IfFoodAhead, Prog2, Prog3\}$ (see [7] for a full description). In order to have an alternative encoding (with, as we will see, different performance and locality characteristics), we now propose a second function set: $F_{A4} = \{IfFoodAhead, Prog2, Prog3, Prog4\}$. The only difference is the addition of an extra sequencing function, Prog4, which runs each of its four subtree arguments in order.

The second problem is the Boolean Even-3-Parity where the goal is to evolve a function that returns true if an even number of the inputs evaluate to true, and false otherwise. The maximum fitness for this problem is 8 (2^3). The terminal set used for this problem is the set of inputs, often called

$T = \{D_0, D_1, D_2\}$. The standard function set is $F_{E3} = \{NOT, OR, AND\}$ and again, we propose a new function set for comparison: $F_{E4} = \{AND, OR, NAND, NOR\}$. Each of the function sets is complete and sufficient to represent an optimal solution (indeed, any boolean function). Again our assumption, shown below to be correct, is that they will differ in performance and in locality characteristics.

For our studies we have considered the use of three different mutation operators:

- Subtree mutation replaces a randomly selected subtree with another randomly created subtree [7].
- One-Point mutation replaces a node (leaf or internal) in the individual by a new node chosen randomly among those of the same arity, taking the arity of a leaf as zero. In standard GP one-point mutation is generally applied with a per-node probability, but in our experiments, since we define genotypic neighbourhood in terms of single mutations, we will apply a single one-point mutation per mutation event.
- Structural mutation is composed of two complementary parts, inflate and deflate mutation. The former consists of inserting a terminal node beneath a function whose arity a is lower than the maximum arity defined in the function set and replacing the function by another of arity $a + 1$; the latter consists of deleting a terminal beneath a function whose arity is at least 1 and replacing that function by another of arity $a - 1$ [23]. Note that this mutation operator can only be applied when functions of the appropriate arity exist: in particular, the alternative boolean function set ($F_{E4} = \{AND, OR, NAND, NOR\}$) contains functions all of arity 2, and so structural mutation will not be applicable.

To have sufficient statistical data, we created 1,250,000 individuals for each of the three mutation operators described previously (in total 3,750,000 individuals). These samplings were created using traditional ramped half-and-half initialisation method described in [7] using depths = $[1, 8]$. By using this method, we guarantee that we will use trees of different sizes and shapes, so no bias is imposed in our sampling.

For each data point in the sample data, we created an offspring via mutation, as in our locality definitions (Section III). In the following section we present and describe the results on locality using these mutations on the two problems using the two function sets and three mutation operators for each.

V. RESULTS

We begin by visually examining the distributions of fitness distances induced by the mutation operators. Figure 1 shows the frequency of each possible fitness distance (fd) between individuals, for the two problems and two function sets for each problem.

For the Artificial Ant Problem, fitness differences of up to 89 are possible, but larger values are rare and decrease roughly linearly, continuing the trend shown in Figure 1 (top). We have therefore omitted values above 8 to make

TABLE II
PARAMETERS USED TO CONDUCT OUR EXPERIMENTS.

Selection	Tournament (size 7)
Initial Population	Ramped half and half (depth 1 to 8)
Population size	50, 100, 125, 200, 250, 500
Generations	500, 250, 200, 125, 100, 50
Runs	50
Mutations	One Point, Subtree, Structural
Mutation rate	One single mutation per individual
Termination	Maximum number of generations

the important values easier to visualise. We can see that a high number of mutations are fitness-neutral (fitness distance = 0), regardless of the function set or mutation used. Using four function (F_{A4}) produces slightly fewer fitness-neutral mutations, with a proportional increase for larger fitness differences. We can also see that only a small proportion of individuals are fitness-neighbours (defined as $fd = 1$). For larger fitness difference values (i.e., $fd > 1$), it is very difficult to see a difference between the 3- and 4-member function sets F_{A3} and F_{A4}. As we will see, however, the locality equations do distinguish between these cases.

The Even-3-Parity Problem has 8 fitness cases; 9 possible fitness values, including 0; and 9 possible fitness distances. The frequency of occurrence of each fitness distance is shown in Figure 1 (bottom). The number of neutral mutations ($fd = 0$) is again much larger compared to non-fitness-neutral mutations ($fd > 0$), regardless of the function set used. For the standard function set (F_{E3}), it seems that one-point mutation produces the greatest number of fitness-neutral mutations (i.e., $fd = 0$), followed by subtree and structural mutation. Structural mutation seems to produce the greatest number of fitness neighbours (i.e., $fd = 1$). When using F_{E4}, the situation is less clear and again, one really needs to take a look at the results on locality by using the locality equations, discussed next.

A. Evolutionary Runs

To see if the prediction made by locality (Equation 1) is correct, we performed actual runs for each of the problems presented in Section IV and the two instances on each on them (i.e., two different function sets). For this purpose, we used a mutation-GP system (using each of the three mutation operators in separate runs). According to the one-operator, one-landscape principle [12], and since we define locality using mutation only, crossover was not used. This methodology is standard [5]. To obtain meaningful results, we performed 50 independent runs, using different combinations of population sizes and generations. Runs were stopped when the maximum number of generations was reached. The rest of the parameters are shown in Table II. The best fitness per generation, averaged over 50 runs, is shown in Figure 2.

B. Comparing alternative encodings

Next we consider the locality properties of the alternative encodings, that is alternative function sets, comparing the results given by the three locality definitions, as explained in Section III.
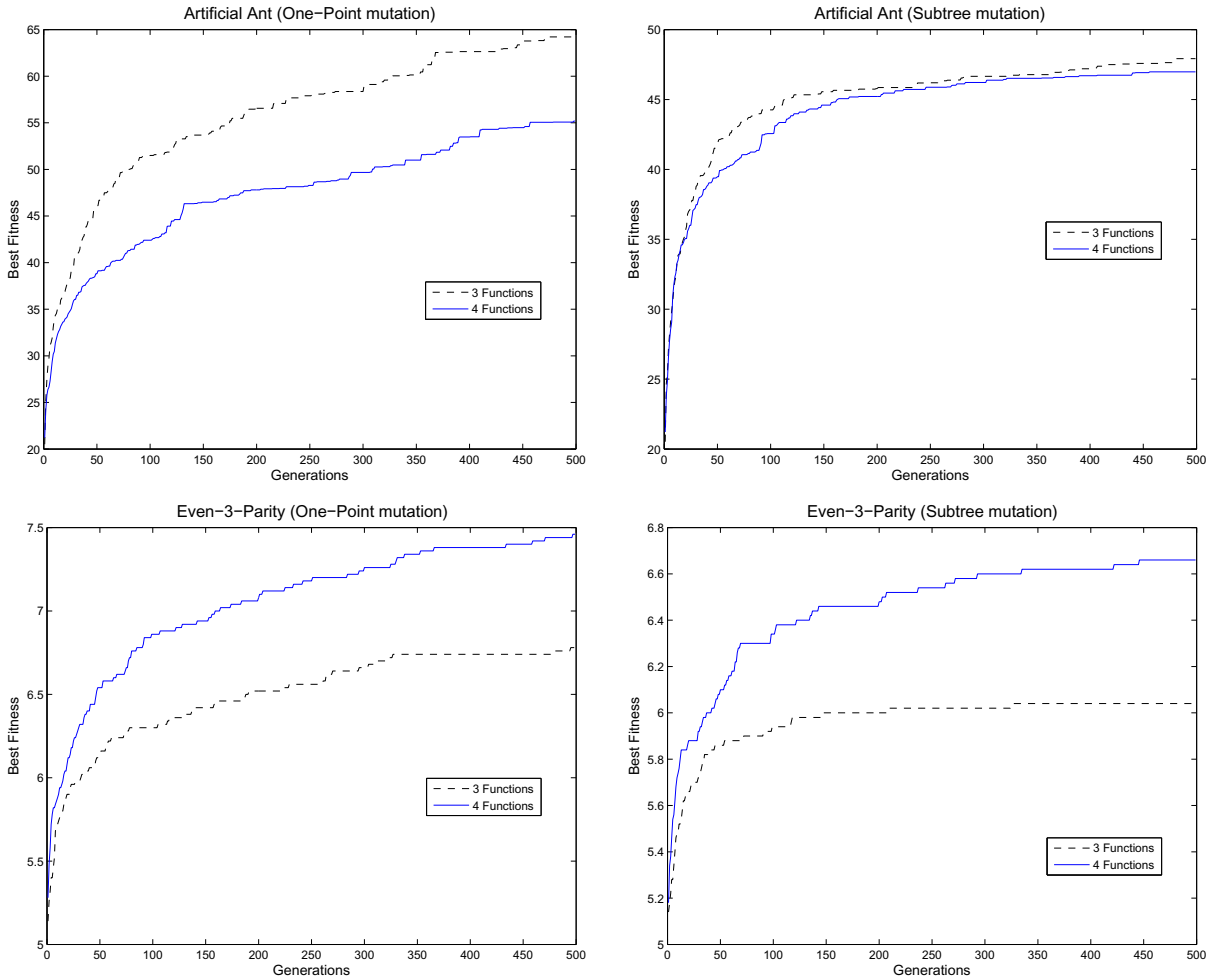
Fig. 2. Best fitness for the Artificial Ant Problem (top) and the Even-3-Parity problem (bottom) and using two mutation operators: one-point (left) and subtree mutation (right).

In Tables III and IV, we show the locality of the two problems (Artificial Ant and Even-3-Parity) and their instances (i.e., two different function sets shown in Table I) and three mutation operators. For the Artificial Ant Problem (Table III), we can see that the function set F_{A3} has better properties of locality (i.e., lower number) compared to F_{A4}. That is, the typical function set used for this problem and using any of the mutations used in this work (one-point, subtree and structural) is predicted to be better. For the Even-3-Parity Problem, the best properties of locality are when using F_{E4} (i.e., $AND, OR, NAND, NOR$).

We now relate these predictions to performance (e.g., average of best fitness per run or finding a solution). For this, we used six different values for the population size as well as for numbers of generations, giving a total of 25,000 individuals in every case. Let us start our analysis by looking at the results found on the Artificial Ant Problem (see Table V). According to the results on locality shown in Table III, we should expect to see a better performance (in terms of the average of the best fitnesses out of 50 independent runs) when using F_{A3} and using any of the three mutation

operators. In fact, we can see that the best performance is achieved when using F_{A3}. This was correctly predicted using all three definitions of locality: in all combinations of population size and number of generations, they gave a better locality value for F_{A3} (i.e., $IF, PROG2, PROG3$). In fact, this is consistent with the best fitness per generation on the 50 independent runs using 50 individuals and 500 generations[4] (see top of Figure 2).

For our second problem (Even-3-Parity), the prediction obtained using the $fd = 1$ definition of locality was again correct, but the other definitions did not predict the differing performance (see Table IV). In this case, we can see that there are better properties of locality when using F_{E4} (i.e., $AND, OR, NAND, NOR$) on the one-point and subtree mutation (structural mutation being inapplicable with this function set). This is, in fact, the case for all the combinations we tested on this problem (see Table VI, where the best performance (measured in terms of finding the global optimum)

[4]Due to space constraints we show only these combinations of values for population size and generations. However, it is worth mentioning that the same trend is seen when using the rest of the values shown in Table II.

	$fd_{\min} = 0$		$fd_{\min} = 1$		$fd_{\min} \leq 1$	
	F_{A3}	F_{A4}	F_{A3}	F_{A4}	F_{A3}	F_{A4}
One Point	1.8790	2.0896	**2.1467**	**2.2628**	1.5128	1.6764
Subtree	2.1653	2.2804	**2.3199**	**2.3708**	1.7426	1.8256
Structural	2.1771	2.4162	**2.3197**	**2.4124**	1.7484	1.9143

	$fd_{\min} = 0$		$fd_{\min} = 1$		$fd_{\min} \leq 1$	
	F_{E3}	F_{E4}	F_{E3}	F_{E4}	F_{E3}	F_{E4}
One Point	0.0968	0.1435	**0.9196**	**0.8904**	0.0082	0.0169
Subtree	0.1368	0.1368	**0.8780**	**0.8760**	0.0064	0.0069
Structural	0.1769	NA	0.8740	NA	0.0254	NA

is seen when using F_{E4}. Again, as for the Artificial Ant Problem, this is consistent with the best fitness per generation on the runs performed to corroborate the prediction obtained with locality (see bottom of Figure 2 where population size = 50 and generations = 500).

C. Comparing alternative mutation operators

We next compare the locality properties of the three mutation operators, comparing the results given by the three locality definitions with the performance in actual GP runs.

Comparing Tables III and V, we see that on the Artificial Ant problem, locality is best for the one-point operator according to all three definitions. However, performance was mixed: the one-point operator sometimes performed better in evolutionary runs than the other operators, sometimes worse (using either encoding). Thus none of the three locality definitions predicted performance correctly here. There was little to choose between subtree and structural mutation in this case.

On the Even-3 Parity problem, the one-point mutation operator now has worse locality, according to the $fd_{\min} = 1$ and the conditional definitions of locality (Figure IV). These predictions are seen to be correct when we look at performance (Figure VI). The $fd_{\min} = 0$ definition of locality gives mixed messages concerning performance (Figure IV).

VI. CONCLUSIONS

Rothlauf [4] described and analysed the importance of locality in performing an effective evolutionary search of landscapes. According to Rothlauf, a representation that has high locality is necessary for an efficient evolutionary search.

In this work, we have extended Rothlauf's quantitative definition of genotype-phenotype locality to the genotype-fitness mapping, considering the issues of fitness neighbour-

	OnePoint		Subtree		Structural	
	F_{A3}	F_{A4}	F_{A3}	F_{A4}	F_{A3}	F_{A4}
P(500), G(50)	57.92	49.32	59.10	53.78	57.10	54.26
P(250), G(100)	58.22	52.40	55.06	52.14	61.01	54.82
P(200), G(125)	59.66	53.66	59.00	53.76	58.12	55.24
P(125), G(200)	58.67	53.05	58.79	52.12	58.12	55.24
P(100), G(250)	59.24	52.12	52.58	48.86	60.46	55.52
P(50), G(500)	61.36	53.62	52.32	47.06	61.22	58.74

	OnePoint		Subtree		Structural	
	F_{E3}	F_{E4}	F_{E3}	F_{E4}	F_{E3}	F_{E4}
P(500), G(50)	4%	18%	0%	4%	-	NA
P(250), G(100)	4%	18%	10%	16%	-	NA
P(200), G(125)	2%	28%	6%	14%	-	NA
P(125), G(200)	6%	32%	4%	12%	-	NA
P(100), G(250)	6%	28%	2%	16%	-	NA
P(50), G(500)	6%	46%	6%	28%	-	NA

hood and neutrality in particular. For this purpose, we have used two problems with significantly different landscape features: a multimodal deceptive landscape and a highly neutral landscape (both believed to be common in many real-world problems). We have used two instances for each of these problems and analysed the locality present on them by defining neighbourhood in the fitness space in three different ways: when the fitness distance is 0 ($fd = 0$), when it is 1 ($fd = 1$) and a combination of these (i.e., when $fd < 2$, the individuals are treated as neighbours).

We have seen that the correct prediction was obtained more often when neighbourhood was defined with $fd = 1$, which corresponds to the definition given by Rothlauf in his genotype-phenotype mapping studies using bitstrings [4]. To corroborate this finding, we performed independent runs and used different combinations of population sizes and generations (i.e., six in total). In all of them, this definition of locality correctly predicted the differing performance of the two function sets.

In the case of mutation operators, predicting performance was much more difficult, and none of the locality definitions achieved very good predictions. It is acknowledged that the differences between mutation operators are not entirely summarised by the impact they have on locality. Instead, it is natural to see (for example) subtree mutation as being more explorative and one-point as more exploitative. These "side effects" may be the cause of the mixed results obtained when comparing the locality predictions with actual performance

of the mutation operators.

Nevertheless, we can conclude that performance is best predicted by a genotype-fitness locality definition which takes individuals as fitness neighbours if they differ by 1 fitness unit, and which treats fitness-neutrality as detrimental to locality.

ACKNOWLEDGMENTS

REFERENCES

[1] S. Wright, "The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution," in *Proceedings of the Sixth International Congress on Genetics*, D. F. Jones, Ed., vol. 1, 1932, pp. 356–366.

[2] F. Rothlauf and D. Goldberg, "Redundant Representations in Evolutionary Algorithms," *Evolutionary Computation*, vol. 11, no. 4, pp. 381–415, 2003.

[3] F. Rothlauf and M. Oetzel, "On the locality of grammatical evolution," in *Proceedings of the 9th European Conference on Genetic Programming*, ser. Lecture Notes in Computer Science, P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekárt, Eds., vol. 3905. Budapest, Hungary: Springer, 10 - 12 Apr. 2006, pp. 320–330. [Online]. Available: http://link.springer.de/link/service/series/0558/papers/3905/39050320.pdf

[4] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*, 2nd ed. Physica-Verlag, 2006.

[5] F. Rothlauf and M. Oetzel, "On the Locality of Grammatical Evolution," in *EuroGP*, ser. Lecture Notes in Computer Science, P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekart, Eds., vol. 3905. Springer, 2006, pp. 320–330.

[6] E. Galvan-Lopez, M. O'Neill, and A. Brabazon, "Towards Understanding the Effects of Locality in GP," *Mexican International Conference on Artificial Intelligence*, pp. 9–14, 2009.

[7] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, Massachusetts: The MIT Press, 1992.

[8] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*. Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk, 2008, (With contributions by J. R. Koza). [Online]. Available: http://www.gp-field-guide.org.uk

[9] H. Beyer and H. Schwefel, "Evolution strategies–A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[10] W. B. Langdon and R. Poli, *Foundations of Genetic Programming*. Berlin: Springer, 2002.

[11] M. Collins, "Finding needles in haystacks is harder with neutrality," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2005, pp. 1613–1618.

[12] T. Jones, "Evolutionary algorithms, fitness landscapes and search," Ph.D. dissertation, University of New Mexico, Albuquerque, 1995.

[13] T. Jones and S. Forrest, "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms," in *Proceedings of the 6th International Conference on Genetic Algorithms*, L. J. Eshelman, Ed. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1995, pp. 184–192.

[14] L. Altenberg, "Fitness Distance Correlation Analysis: An Instructive Counterexample," in *Proceedings of the Seventh International Conference on Genetic Algorithms*, T. Back, Ed. San Francisco, CA, USA: Morgan Kaufmann, 1997, pp. 57–64.

[15] R. J. Quick, V. J. Rayward-Smith, and G. D. Smith, "Fitness Distance Correlation and Ridge Functions," in *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1998, pp. 77–86.

[16] M. Clergue and P. Collard, "GA-Hard Functions Built by Combination of Trap Functions," in *CEC 2002: Proceedings of the 2002 Congress on Evolutionary Computation*, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Schackleton, Eds. IEEE Press, 2002, pp. 249–254.

[17] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue, "A study of fitness distance correlation as a difficulty measure in genetic programming," *Evolutionary Computation*, vol. 13, no. 2, pp. 213–239, 2005.

[18] B. Naudts and L. Kallel, "A comparison of predictive measures of problem difficulty in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 1, pp. 1–15, April 2000.

[19] E. Galván-López and R. Poli, "Some Steps Towards Understanding How Neutrality Affects Evolutionary Search," in *Parallel Problem Solving from Nature (PPSN IX). 9th International Conference*, ser. LNCS, T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds., vol. 4193. Reykjavik, Iceland: Springer-Verlag, 9-13 Sep. 2006, pp. 778–787.

[20] R. Poli and E. Galván-López, "On The Effects of Bit-Wise Neutrality on Fitness Distance Correlation, Phenotypic Mutation Rates and Problem Hardness," in *Foundations of Genetic Algorithms IX*, ser. Lecture Notes in Computer Science, C. R. Stephens, M. Toussaint, D. Whitley, and P. Stadler, Eds. Mexico city, Mexico: Springer-Verlag, 8-11 Jan. 2007, pp. 138–164.

[21] E. Galván-López, S. Dignum, and R. Poli, "The Effects of Constant Neutrality on Performance and Problem Hardness in GP," in *EuroGP 2008 - 11th European Conference on Genetic Programming*, ser. LNCS, M. ONeill, L. Vanneschi, S. Gustafson, A. I. E. Alcazar, I. D. Falco, A. D. Cioppa, and E. Tarantino, Eds., vol. 4971. Napoli, Italy: Springer, 26–28 Mar. 2008, pp. 312–324.

[22] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue, "A Study of Fitness Distance Correlation as a Difficulty Measure in Genetic Programming," *Evolutionary Computation*, vol. 13, no. 2, pp. 213–239, Summer 2005.

[23] L. Vanneschi, "Theory and practice for efficient genetic programming," Ph.D. dissertation, Faculty of Science, University of Lausanne, Switzerland, 2004.

[24] L. Vanneschi, M. Tomassini, P. Collard, and M. Clergue, "Fitness distance correlation in structural mutation genetic programming," in *EuroGP*, ser. Lecture notes in computer science. Springer, 2003, pp. 455–464.

[25] L. Vanneschi, M. Clergue, P. Collard, M. Tomassini, and S. Verel, "Fitness clouds and problem hardness in genetic programming," in *EuroGP*, ser. LNCS. Springer, 2004, pp. 690–701.

[26] L. Vanneschi, M. Tomassini, P. Collard, S. Verel, Y. Pirola, and G. Mauri, "A comprehensive view of fitness landscapes with neutrality and fitness clouds," in *Proceedings of EuroGP 2007*, ser. LNCS, vol. 4445. Springer, 2007, pp. 241–250.

[27] R. Poli and L. Vanneschi, "Fitness-proportional negative slope coefficient as a hardness measure for genetic algorithms," in *Proceedings of GECCO '07*, London, UK, 2007, pp. 1335–1342.

[28] E. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biological Cybernetics*, vol. 63, no. 5, pp. 325–336, 1990.

[29] W. Hordijk, "A measure of landscape," *Evolutionary Computation*, vol. 4, no. 4, pp. 335–360, 1996.

[30] P. D'haeseleer and J. Bluming, "Effects of locality in individual and population evolution," in *Advances in Genetic Programming*, K. E. Kinnear, Ed. MIT Press, 1994, pp. 177–198.

[31] F. Rothlauf, "On the bias and performance of the edge-set encoding," *IEEE transactions on evolutionary computation*, vol. 13, no. 3, pp. 486–499, June 2009.

[32] E. Galván-López, "An Analysis of the Effects of Neutrality on Problem Hardness for Evolutionary Algorithms," Ph.D. dissertation, School of Computer Science and Electronic Engineering, University of Essex, United Kingdom, 2009.

[33] P. K. Lehre and P. C. Haddow, "Phenotypic Complexity and Local Variations in Neutral Degree," *BioSystems*, vol. 87, no. 2-3, pp. 233–42, 2006.

[34] M. Toussaint and C. Igel, "Neutrality: A necessity for self-adaptation," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, 2002, pp. 1354–1359.

[35] W. Langdon and R. Poli, "Why ants are hard," in *Proceedings of the Third Annual Conference on Genetic Programming*, J. R. Koza, Ed. Morgan Kaufmann, Madison, USA, 1998, pp. 193–201.