Difficulty of Unimodal and Multimodal Landscapes in Genetic Programming

Leonardo Vanneschi[1], Marco Tomassini[1], Manuel Clergue[2], and Philippe Collard[2]

[1] Computer Science Institute, University of Lausanne, Lausanne, Switzerland
[2] I3S Laboratory, University of Nice, Sophia Antipolis, France

Abstract. This paper presents an original study of fitness distance correlation as a measure of problem difficulty in genetic programming. A new definition of distance, called structural distance, is used and suitable mutation operators for the program space are defined. The difficulty is studied for a number of problems, including, for the first time in GP, multimodal ones, both for the new hand-tailored mutation operators and standard crossover. Results are in agreement with empirical observations, thus confirming that fitness distance correlation can be considered a reasonable index of difficulty for genetic programming, at least for the set of problems studied here.

1 Introduction

The fitness distance correlation (*fdc*) coefficient has been used as a tool for measuring problem difficulty in genetic algorithms (GAs) and genetic programming (GP) with controversial results: some counterexamples have been found for GAs [15], but *fdc* has been proven an useful measure on a large number of GA (see for example [3] or [9]) and GP functions (see [2,16]). In particular, Clergue and coworkers ([2]) have shown *fdc* to be a reasonable way of quantifying problem difficulty for GP for a set of functions.

In this paper, we use a measure of structural distance for trees (see [7]) to calculate *fdc*. Then, we employ *fdc* to measure problem difficulty for two kinds of GP processes: GP using the standard Koza's crossover as the only genetic operator (that we call from now on *standard GP*) and GP using two new mutation operators based on the transformations on which structural distance is defined (*structural mutation genetic programming* from now on). The test problems that we have chosen are unimodal and multimodal trap functions, royal trees and two versions of the MAX problem. The present study is the first attempt to quantify problem difficulty of functions with multiple global optima with the same fitness in evolutionary algorithms by the *fdc*.

This paper is structured as follows: the next section gives a short description of the structural tree distance used in the paper, followed by the definition of the basic mutation operators that go hand-in-hand with it. Section 4 presents the main results and their discussion for a number of GP problems. Finally, Sect. 5 gives our conclusions and hints to future work.

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1788–1799, 2003.

2 Distance Measure for Genetic Programs

In GAs individuals are represented as strings of digits and typical distance measures are Hamming distance or alternation. Defining a distance between genotypes in GP is much more difficult, given the tree structure of the individuals. In [2] an *ad hoc* distance between trees was used. In the spirit of the *fdc* definition [9], it would be better to use GP operators that have a direct counterpart in the tree distance. For that reason, we adopt the structural distance for trees proposed in [7] and we define corresponding operators (see next section). According to this measure, given the sets $\mathcal{F}$ and $\mathcal{T}$ of functions and terminal symbols, a coding function c must be defined such that $c : \{\mathcal{T} \cup \mathcal{F}\} \rightarrow \mathbb{N}$. One can think of many ways for the specification of c, for example the "complexity" of the primitives or their arity. The distance between two trees T_1 and T_2 is calculated in three steps: (1) T_1 and T_2 are overlapped at the root node and the process is applied recursively starting from the leftmost subtrees (see [7] for a description of the overlapping algorithm). (2) For each pair of nodes at matching positions, the difference of their codes (eventually raised to an exponent) is computed. (3) The differences computed in the previous step are combined in a weighted sum. This gives for the distance between two trees T_1 and T_2 with roots R_1 and R_2 the following expression:

$$dist(T_1, T_2) = d(R_1, R_2) + k\sum_{i=1}^{m} dist(child_i(R_1), child_i(R_2)) \qquad (1)$$

where: $d(R_1, R_2) = (|c(R_1) - c(R_2)|)^z$, $child_i(Y)$ is the i^{th} of the m possible children of a generical node Y, if $i \leq m$, or the empty tree otherwise, and c evaluated on the root of an empty tree is 0. Constant k is used to give different weights to nodes belonging to different levels and z is a constant usually chosen in such a way that $z \in \mathbb{N}$. In most of this paper, except for the MAX function, individuals will be coded using the same syntax as in [2] and [14], i.e. considering a set of functions A, B, C, etc. with increasing arity (i.e. $arity(A) = 1$, $arity(B) = 2$, and so on) and a single terminal X (i.e. $arity(X) = 0$) as follows: $\mathcal{F} = \{A, B, C, D, \ldots\}, \mathcal{T} = \{X\}$ and the c function will be defined as follows: $\forall x \in \{\mathcal{F} \cup \mathcal{T}\} \quad c(x) = arity(x) + 1$. In our experiments we will always set $k = \frac{1}{2}$ and $z = 2$. By keeping $0 < k < 1$, the differences near the root have higher weight. This is convenient for GP as it has been noted that programs converge quickly to a fixed root portion [11].

3 Structural Mutation Operators

In [13] O'Reilly used the edit distance, which is related to, but not identical with the tree distance employed here. She also defined suitable edit operators and related them to GP mutation. She used the edit distance for a different purpose: how to measure and control the step size of crossover in order to balance explotation and exploration in GP. She also suggested that edit distance could

be useful in the study of GP fitness landscapes, but did not develop the issue. Here we take up exactly this last point.

The following operators have been inspired by the distance definition presented in the last section and by the work in [13]. Given the sets $\mathcal{F}$ and $\mathcal{T}$ and the coding function c defined in Sect. 2, we define c_{max} (respectively, c_{min}) as the maximum (respectively, the minimum) value taken by c on the domain $\{\mathcal{F} \cup \mathcal{T}\}$. Moreover, given a symbol n such that $n \in \{\mathcal{F} \cup \mathcal{T}\}$ and $c(n) < c_{max}$ and a symbol m such that $m \in \{\mathcal{F} \cup \mathcal{T}\}$ and $c(m) > c_{min}$, we define: $succ(n)$ as a primitive such that $c(succ(n)) = c(n) + 1$ and $pred(m)$ as a primitive such that $c(pred(m)) = c(m) - 1$. Then we can define the following editing operators on a generic tree T:

- **inflate mutation**. A primitive labelled with a symbol n such that $c(n) < c_{max}$ is selected in T and replaced by $succ(n)$. A new random terminal node is added to this new node in a random position (i.e. the new terminal becomes the i^{th} son of $succ(n)$, where i is comprised between 0 and $arity(n)$).
- **deflate mutation**. A primitive labelled with a symbol m such that $c(m) > c_{min}$, and such that at least one of his sons is a leaf, is selected in T and replaced by $pred(m)$. A random leaf, between the sons of this node, is deleted from T.

The terms *inflate* and *deflate* have been used to avoid confusion with the similar and well known *grow* and *shrink* mutations that have already been proposed in GP. The following property holds (the proof appears in [17]):

Property 1. Distance/Operator Consistency.
Let's consider the sets $\mathcal{F}$ and $\mathcal{T}$ and the coding function c defined in Sect. 2. Let T_1 and T_2 be two trees composed by symbols belonging to $\{\mathcal{F} \cup \mathcal{T}\}$ and let's consider the k and z constants of definition (1) to be equal to 1. If $dist(T_1, T_2) = D$, then T_2 can be obtained from T_1 by a sequence of $\frac{D}{2}$ editing operations, where an editing operation can be an inflate mutation or a deflate mutation.

From this property, we conclude that the operators of inflate mutation and deflate mutation are completely coherent with the notion of distance defined in Sect. 2, i.e. an application of these operators allow us to move on the search space from a tree to its neighbors according to that distance. We call the new GP process based on these operators structural mutation genetic programming (SMGP).

4 Experimental Results

In all experiments shown in the following, *fdc* has been calculated via a sampling of 40000 randomly chosen individuals. All the experiments have been done with generational GP, a total population size of 100 individuals, ramped half and a half initialization, tournament selection of size 10. When standard GP has been used, the crossover rate has been set to 95% and the mutation rate to 0%, while

when SMGP has been employed, no crossover has been used and the rate of the two new mutation operators has been set to 95%. The GP process has been stopped either when a perfect solution has been found (global optimum) or when 500 generations have been executed. All experiments have been performed 100 times.

4.1 Fitness Distance Correlation

An approach proposed for GAs [9] states that an indication of problem hardness is given by the relationship between fitness and distance of the genotypes from known optima. Given a sample $F = \{f_1, f_2, ..., f_n\}$ of n individual fitnesses and a corresponding sample $D = \{d_1, d_2, ..., d_n\}$ of the n distances to the nearest global optimum, *fdc* is defined as: $fdc = \frac{C_{FD}}{\sigma_F \sigma_D}$, 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. Given the tree structure of genotypes, the normalization problem is not a trivial one in GP. Dividing all the distances in the sampling by the maximum of all the possible distances between two trees in the search space is not a practically applicable method, since it would give too large a number for the typically used maximum tree depth and operator arity values. This problem has been tackled in [4]. In [2], Clergue and coworkers used a "sufficiently large" integer constant to obtain the normalisation of distances. Here, similarly to [4], we obtain the normalized distance between two trees by dividing their distance by the maximum value between the distances of the same two trees from the empty one. As suggested in [9], GA problems can be empirically classified in three classes, depending on the value of the *fdc* coefficient: **misleading** ($fdc \geq 0.15$), in which fitness increases with distance, **unknown** ($-0.15 < fdc < 0.15$) in which there is virtually no correlation between fitness and distance and **straightforward** ($fdc \leq -0.15$) in which fitness increases as the global optimum approaches. The second class corresponds to problems for which the difficulty can't be estimated, because *fdc* doesn't bring any information. In this case, examination of the fitness-distance scatterplot may give information on problem difficulty (see [9]).

4.2 Unimodal Trap Functions

Trap functions [5] allow one to define the fitness of the individuals as a function of their distance from the optimum, and the difficulty of trap functions can be changed by simply modifying some parameters. A function $f : distance \rightarrow fitness$ is a unimodal trap function if it is defined in the following way:

$$f(d) = \begin{cases} 1 - \dfrac{d}{B} & \text{if } d \leq B \\ \dfrac{R \cdot (d - B)}{1 - B} & \text{otherwise} \end{cases}$$

where d is the distance of the current individual from the *unique* global optimum, normalized so as to belong to the range $[0, 1]$, and B and R are constants

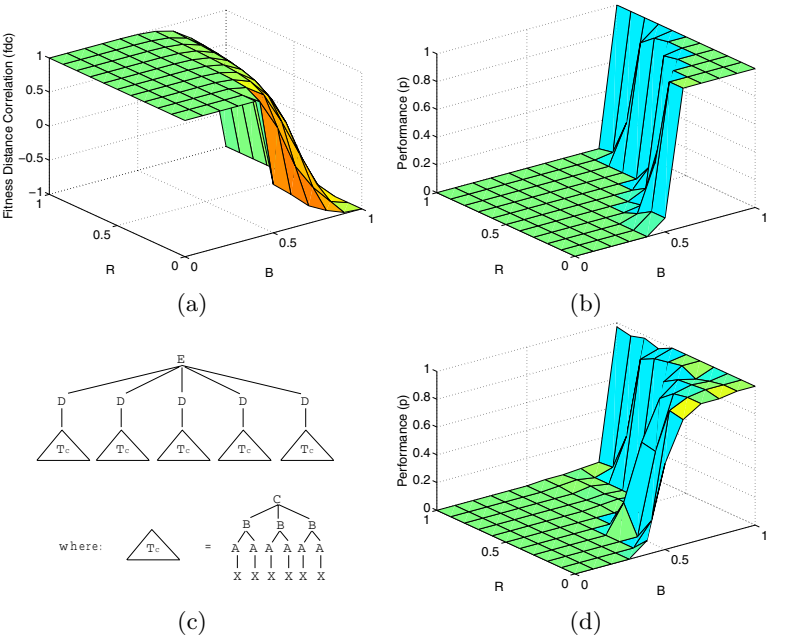


Fig. 1. (a): *fdc* values for some trap functions obtained by changing the values of the constants B and R. (b): Performance (p) values of SMGP for traps. (d): Performance (p) values of standard GP for traps. (c): Stucture of the tree used as optimum in the experiments reported in (a), (b) and (d).

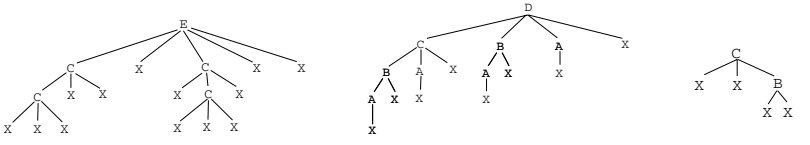


Fig. 2. The trees used as optima in experiments analogous to the ones of Fig. 1.

belonging to $[0, 1]$. B allows to set the width of the attractive basin for each one of the optima and R sets their relative importance. By construction, the difficulty of trap functions decreases as the value of B increases, while it increases as the value of R decreases. For a more detailed explanation of trap functions see for instance [2]. Figure 1 show values of the performance p (defined as the proportion of the runs for which the global optimum has been found in less than 500 generations over 100 runs) and of *fdc* for various trap functions obtained by changing the values of the constants B and R. These experiments have been performed considering the tree represented in Fig. 1c as the global optimum. The same experiments have been performed using as global optimum the trees shown in Fig. 2 and the results (not shown here for lack of space) are qualitatively analogous to the ones of Fig. 1. In all cases *fdc* is confirmed to be a reasonable

measure for quantifying problem difficulty both for SMGP and standard GP for unimodal trap functions.

4.3 "W" Multimodal Trap Functions

Unimodal trap functions described in Sect. 4.2 are characterized by the presence of a unique global optimum. The functions used here and first proposed in [6] (informally called "W" trap functions, given their typical shape shown in Fig. 3) are characterized by the presence of several global optima. They depend on 5 variables called B_1, B_2, B_3, R_1 and R_2 and they can be expressed by the following formula:

$$f(d) = \begin{cases} 1 - \dfrac{d}{B_1} & \text{if } d \leq B_1 \\ \dfrac{R_1 \cdot (d - B_1)}{B_2 - B_1} & \text{if } B_1 \leq d \leq B_2 \\ \dfrac{R_1 \cdot (B_3 - d)}{B_3 - B_2} & \text{if } B_2 \leq d \leq B_3 \\ \dfrac{R_2 \cdot (d - B_3)}{1 - B_3} & \text{otherwise} \end{cases}$$

where B_1, B_2, B_3, R_1 and R_2 are constants belonging to the interval $[0, 1]$ and the property $B_1 \leq B_2 \leq B_3$ must hold. A systematic study of problem difficulty for multimodal landscapes via an algebraic indicator such as *fdc* has, to our knowledge, never been performed before in evolutionary computation. Here is how our study has been performed: we choose a particular tree belonging to the search space and we call it *origin*. Then we artificially assign a maximum fitness (= 1) to this tree, so as to make it a global optimum. Then, if we set the

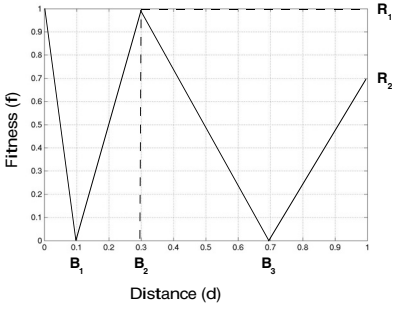


Fig. 3. Graphical representation of a "W" trap function with $B_1 = 0.1$, $B_2 = 0.3$, $B3 = 0.7$, $R_1 = 1$, $R_2 = 0.7$. Note that distances and fitness are normalized into the interval [0, 1].

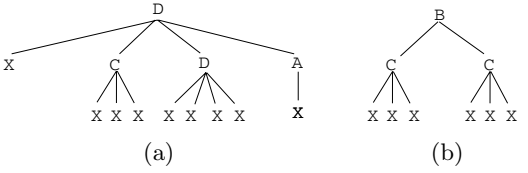


Fig. 4. Two trees having a normalized distance equal to 0.5 between them

R_1 constant to 1, all the trees having a normalized distance equal to B_2 from the origin, given the definition of "W" trap functions, have a fitness equal to 1 and thus are global optima too. Two series of experiments have been performed, using both SMGP and standard GP. In the first one, the tree represented in Fig. 4a is considered as the origin (let's call this tree T_1).

Since the tree in Fig. 4b (that we call T_2) has a normalized distance of 0.5 from T_1, if we set R_1 to 1 and B_2 to 0.5 then T_2 is a global optimum too. To calculate the *fdc*, the minimum of the distances from T_1 and T_2 to each tree in the sampling is considered (as suggested for GAs by Jones in [9], even though never practically experimented). The distribution of these "minimum distances" has been studied on a sample of 40000 individuals and it appears to have a regular shape, as it is the case for unimodal trap functions (results not shown here for lack of space).

Since these series of experiments represent a first attempt to use *fdc* to measure the difficulty of fitness landscapes with more than one global optimum, and since we want to be able to study the dynamics of the GP process in detail, we have decided, as a first step, to use fitness landscapes containing only two global optima. Thus, a random normalized fitness different from 1 has been arbitrarily assigned to each tree having a distance equal to 0.5 from T_1 and a genotype different from T_2. This choice, of course, alters the "W" trap functions definition and influences the fitness landscape, even though only marginally. Anyway, we consider this choice perfectly "fair", given that we are not interested in testing *fdc* on standard benchmarks, but rather on artificially defined fitness landscapes. Table 1 shows a subset of the experimental results that we have obtained with this first series of tests, by setting B_2 to 0.5 and R_1 to 1 and by varying the values of B_1, B_3 and R_2.

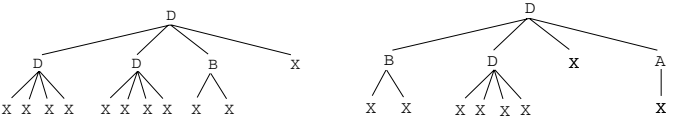
Given the enormous number of experiments performed (about 10000 GP runs) and the limited space available, we couldn't show all the results. Thus, we have decided to proceed as follows: for each trap function (identified by particular values of B_1, B_3 and R_2), we have calculated the *fdc*. Then, we have discarded all the traps for which the *fdc* is comprised between -0.15 and 0.15, because no experimental result would give us any useful information for these functions (see Sect. 4.1). For a subset of the other trap functions, 100 runs have been performed with both SMGP and standard GP. Results shown in Table 1 are encouraging and suggest that *fdc* could be a reasonable measure to predict problem difficulty for some typical "W" trap functions. The results not shown in Table 1 are qualitatively similar and lead us to the same conclusions.

Table 1. Results of *fdc* using SMGP and standard GP for a set of "W" trap functions where the origin is the tree shown in Fig. 4a and the second global optimum is the one shown in Fig. 4b; p stands for performance

	fdc	*fdc* prediction	p (SMGP)	p (stGP)
$B_1 = 0.4, B_3 = 0.9, R_2 = 0.5$	-0.62	straightf.	0.75	0.81
$B_1 = 0.5, B_3 = 0.8, R_2 = 0.4$	-0.88	straightf.	0.98	0.94
$B_1 = 0.3, B_3 = 0.9, R_2 = 0.7$	-0.61	straightf.	0.80	0.77
$B_1 = 0.2, B_3 = 0.9, R_2 = 0.1$	-0.69	straightf.	0.72	0.91
$B_1 = 0.1, B_3 = 0.9, R_2 = 0.3$	-0.72	straightf.	0.85	0.98
$B_1 = 0.5, B_3 = 0.6, R_2 = 0.9$	0.34	misleading	0.33	0.20
$B_1 = 0.4, B_3 = 0.6, R_2 = 0.9$	0.36	misleading	0.14	0.30
$B_1 = 0.3, B_3 = 0.6, R_2 = 0.9$	0.33	misleading	0.31	0.13

Table 2. Results of *fdc* using SMGP and standard GP for a set of "W" trap functions where the origin is the tree shown in Fig. 5a and the second global optimum is the one shown in Fig. 5b; p stands for performance

	fdc	*fdc* prediction	p (SMGP)	p (stGP)
$B_1 = 0.1, B_3 = 0.1, R_2 = 0.1$	-0.76	straightf.	0.91	0.88
$B_1 = 0.1, B_3 = 0.9, R_2 = 0.1$	-0.93	straightf.	0.97	0.99
$B_1 = 0.05, B_3 = 0.7, R_2 = 0.1$	-0.90	straightf.	0.98	0.92
$B_1 = 0.1, B_3 = 0.8, R_2 = 0.5$	-0.81	straightf.	0.90	0.83
$B_1 = 0.1, B_3 = 0.5, R_2 = 0.1$	-0.71	straightf.	0.91	0.90
$B_1 = 0.05, B_3 = 0.2, R_2 = 0.9$	0.89	misleading	0.02	0.13
$B_1 = 0.05, B_3 = 0.4, R_2 = 0.9$	0.74	misleading	0.35	0.17
$B_1 = 0.05, B_3 = 0.1, R_2 = 0.6$	0.78	misleading	0.25	0.23



(a) (b)

Fig. 5. Two trees having a normalized distance equal to 0.1 between them

Analogous experiments have also been performed by considering the tree in Fig. 5a as the origin and the tree in Fig. 5b as the second global optimum. Since the normalized distance between these two trees is equal to 0.1, this time B_2 is set to 0.1 and R_1 to 1. Once again, a random normalized fitness different from 1 is assigned to all the trees having a distance equal to 0.1 from the one in Fig. 5a and a genotype different from the one in Fig. 5b. A subset of the results of this second series of experiments are shown in Table 2. These results are encouraging too and seem to confirm the validity of the *fdc* both for SMGP and standard GP.

In any event, the amount of results shown in this section doesn't allow us to draw definitive conclusions: many more cases should be studied. For example, in the case of fitness landscapes containing two global optima, a larger set of

differently shaped trees should be considered as origin and as second global optimum. Moreover, fitness landscapes with more than two global optima should be investigated. However, a methodology for the study of difficulty of multimodal landscapes has been proposed, and some non-trivial choices have been performed, as the one of calculating *fdc* by considering the minimum of the distances from the global optima for every individual of the sampling. These choices needed an experimental confirmation. Results shown here are encouraging and should pave the way for a deeper study of problem difficulty for multimodal lanscapes.

4.4 Royal Trees

The next functions we study are the royal trees proposed by Punch *et al.* [14]. The language used is the same as in Sect. 2, and the fitness of a tree (or any subtree) is defined as the score of its root. Each function calculates its score by summing the weighted scores of its direct children. If the child is a perfect tree of the appropriate level (for instance, a complete level-C tree beneath a D node), then the score of that subtree, times a *FullBonus* weight, is added to the score of the root. If the child has a correct root but is not a perfect tree, then the weight is *PartialBonus*. If the child's root is incorrect, then the weight is *Penalty*. After scoring the root, if the function is itself the root of a perfect tree, the final sum is multiplied by *CompleteBonus* (see [14] for a more detailed explanation). Values used here are as in [14] i.e. *FullBonus* = 2, *PartialBonus* = 1, *Penalty* = $\frac{1}{3}$, *CompleteBonus* = 2. Different experiments, considering different nodes as the node with maximum arity allowed, have been performed. Results are shown in Table 3.

Predictions made by *fdc* for level-A, level-B, level-C and level-D functions are correct. Level-E function is "difficult" to be predicted by the *fdc* (i.e. no correlation between fitness and distance is observed). Finally, level-F and level-G functions are predicted to be "misleading" (in accord with Punch in [14]) and they really are, since the global optimum is never found before 500 generations. Royal trees problem spans all the classes of difficulty as described by the *fdc*.

4.5 MAX Problem

The task of the MAX problem for GP, defined in [8] and [10], is "to find the program which returns the largest value for a given terminal and function set

Table 3. Results of *fdc* for the Royal Trees; p stands for performance

Root	*fdc*	*fdc* prediction	p (SMGP)	p (stGP)
B	-0.31	straightf.	1	1
C	-0.25	straightf.	1	1
D	-0.20	straightf.	0.76	0.70
E	0.059	unknown	0	0.12
F	0.44	misleading	0	0
G	0.73	misleading	0	0

Table 4. Results of *fdc* for the MAX problem using SMGP and standard GP. The first column shows the sets of functions and terminals used in the experiments; p stands for performance

MAX problem	*fdc*	*fdc* prediction	p (SMGP)	p (stGP)
{+} {1}	-0.87	straightf.	1	1
{+} {1,2}	-0.86	straightf.	1	1

with a depth limit d, where the root node counts as depth 0". We set d equal to 8 and we use the set of functions $\mathcal{F} = \{+\}$ and the set of terminals $\mathcal{T}_1 = \{1\}$ or $\mathcal{T}_2 = \{1, 2\}$. When using $\mathcal{T}_1$, we specify the coding function c as: $c(1) = 1$, $c(+) = 2$, when using $\mathcal{T}_2$, we pose: $c(1) = 1$, $c(2) = 2, c(+) = 3$. The study of standard GP for these MAX functions comports no particular problem, while for the case of SMGP, the definitions of the operators of inflate and deflate mutation given in Sect. 3 must be slightly modified, since we are considering a different coding language. The inflate and deflate mutations are now defined in such a way that, when using $\mathcal{T}_1$, a terminal symbol 1 can be transformed in the subtree $T_1 = +(1, 1)$ by one step of inflate mutation and the *vice-versa* can be done by the deflate mutation. When using $\mathcal{T}_2$, the inflate mutation can transform a 1 node into a 2 node and a 2 node into the subtrees $T_2 = +(2, 1)$ or $T_3 = +(1, 2)$ (with a uniform probability). On the other hand, the deflate mutation can tranform T_1 or T_2 into a leaf labelled by 2, and a 2 node into a 1 node. Table 4 shows the *fdc* and p values for these test cases. Both problems are correctly classified as straightforward by *fdc*, both for SMGP and standard GP.

5 Conclusions and Future Work

Two new kinds of tree mutations corresponding to the operations of the structural distance are defined in this paper. Fitness distance correlation (calculated using this structural distance between trees) has been shown to be a reasonable way of quantifying the difficulty of unimodal trap functions, of a restricted set of multimodal trap functions, of royal trees, and of two MAX functions for GP using these mutations as genetic operators, as well as for GP based on standard crossover. The results show that, for the functions studied, using crossover or our mutation operators, does not seem to have a marked effect on difficulty as measured by *fdc*. Thus the remarks of [1] and [12], and other work where it is claimed that the standard GP crossover does not seem to markedly improve performance with respect to other variation operators seem to be confirmed. Although *fdc* has confirmed its validity in the present study, in view of some counterexamples for GAs mentioned in the text, it remains to be seen whether the use of *fdc* extends to other classes of functions, such as typical GP benchmarks. In the future we plan to improve the study of problem difficulty of multimodal landscapes and MAX functions (e.g. when $\mathcal{F} = \{+, *\}$ and $\mathcal{T} = \{0.5\}$), and look for a measure for GP difficulty that can be calculated without prior knowledge of the global optima, thus eliminating the strongest limitation of *fdc*. Moreover, since *fdc* is

surely not an infallible measure, we plan to build a counterexample for *fdc* in GP. Another open problem consists in taking into account in the distance definition the phenomenon of introns (whereby two different genotypes can lead to the same phenotypic behavior). Finally, we intend to look for a better measure than performance to identify the success rate of functions, possibly independent from the maximum number of generations chosen.

Acknowledgments. This work has been partially sponsored by the Fonds National Suisse pour la recherche scientifique.

References

1. K. Chellapilla. Evolutionary programming with tree mutations: Evolving computer programs without crossover. In J.R. Koza, K. Deb, M. Dorigo, D.B. Fogel, M. Garzon, H. Iba, and R.L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 431–438, Stanford University, CA, USA, 1997. Morgan Kaufmann.
2. M. Clergue, P. Collard, M. Tomassini, and L. Vanneschi. Fitness distance correlation and problem difficulty for genetic programming. In *Proceedings of the genetic and evolutionary computation conference GECCO'02*, pages 724–732. Morgan Kaufmann, San Francisco, CA, 2002.
3. P. Collard, A. Gaspar, M. Clergue, and C. Escazut. Fitness distance correlation as statistical measure of genetic algorithms difficulty, revisited. In *European Conference on Artificial Intelligence (ECAI'98)*, pages 650–654, Brighton, 1998. John Witley & Sons, Ltd.
4. E.D. de Jong, R.A. Watson, and J.B. Pollak. Reducing bloat and promoting diversity using multi-objective methods. In L. Spector et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO '01*, San Francisco, CA, July 2001. Morgan Kaufmann.
5. K. Deb and D.E. Goldberg. Analyzing deception in trap functions. In D. Whitley, editor, *Foundations of Genetic Algorithms, 2*, pages 93–108. Morgan Kaufmann, 1993.
6. K. Deb, J. Horn, and D. Goldberg. Multimodal deceptive functions. *Complex Systems*, 7:131–153, 1993.
7. A. Ekárt and S.Z. Németh. Maintaining the diversity of genetic programs. In J.A. Foster, E. Lutton, J. Miller, C. Ryan, and A.G.B. Tettamanzi, editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 162–171, Kinsale, Ireland, 3–5 April 2002. Springer-Verlag.
8. C. Gathercole and P. Ross. An adverse interaction between crossover and restricted tree depth in genetic programming. In J.R. Koza, D.E. Goldberg, D.B. Fogel, and R.L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 291–296, Stanford University, CA, USA, 28–31 July 1996. MIT Press.
9. T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, Albuquerque, 1995.

10. W.B. Langdon and R. Poli. An analysis of the max problem in genetic programming. In J.R. Koza, K. Deb, M. Dorigo, D.B. Fogel, M. Garzon, H. Iba, and R.L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference on Genetic Programming*, pages 222–230, San Francisco, CA, 1997. Morgan Kaufmann.
11. N.F. McPhee and N.J. Hopper. Analysis of genetic diversity through population history. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1112–1120, Orlando, Florida, USA, 13–17 July 1999. Morgan Kaufmann.
12. U.-M. O'Reilly. *An Analysis of Genetic Programming.* PhD thesis, Carleton University, Ottawa, Ontario, Canada, 1995.
13. U.-M. O'Reilly. Using a distance metric on genetic programs to understand genetic operators. In *Proceedings of 1997 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4092–4097. IEEE Press, Piscataway, NJ, 1997.
14. B. Punch, D. Zongker, and E. Goodman. The royal tree problem, a benchmark for single and multiple population genetic programming. In P. Angeline and K. Kinnear, editors, *Advances in Genetic Programming 2*, pages 299–316, Cambridge, MA, 1996. The MIT Press.
15. R.J. Quick, V.J. Rayward-Smith, and G.D. Smith. Fitness distance correlation and ridge functions. In *Fifth Conference on Parallel Problems Solving from Nature (PPSN'98)*, pages 77–86. Springer-Verlag, Heidelberg, 1998.
16. V. Slavov and N.I. Nikolaev. Fitness landscapes and inductive genetic programming. In *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA97)*, University of East Anglia, Norwich, UK, 1997. Springer-Verlag KG, Vienna.
17. L. Vanneschi, M. Tomassini, P. Collard, and M. Clergue. Fitness distance correlation in structural mutation genetic programming. In C. Ryan et al., editor, *Genetic Programming, 6th European Conference, EuroGP2003*, Lecture Notes in Computer Science. Springer-Verlag, Heidelberg, 2003.