

Chapter 9

Fitness Landscape Study on Syntactically Constrained Domains

In chapters 4 and 6, we have shown that, thanks to the non-fixed arity property of TAG derivation trees, it is possible to design operators such as insertion, deletion, and point replacement on TAG-based representations. These operators make bounded changes in the genotype space (G_{lex} derivation trees). Since TAG-based representation possesses the locality property (as argued in chapter 3), the changes in the phenotype space (G_{lex} derived trees) are also bounded. In this chapter, we argue that these properties of TAG-based representation are particularly useful in helping to understand problem difficulty, by investigating the characteristics of problem fitness landscapes.

The chapter starts with a brief description of the role of fitness landscape study in the field of evolutionary algorithms (EAs). It is followed by a discussion of the difficulties in fitness landscape study in genetic programming, and especially in grammar-guided genetic programming. Next comes an examination of how those issues relate to fitness landscape studies using TAG-based representation. Finally, two fitness landscape studies in TAG-based representation are presented. The purpose of the first is to understand the impact of changing target function

in a family of functions, and the second to understand the impact of changing grammars, on the fitness landscape characteristics of some standard problems.

9.1 Fitness Landscape Study in EAs

The idea of fitness landscape was first proposed in [Wri1932]. Since then, it has been used as a tool for analysing evolutionary theories [Kau1993], as well as to model the problem difficulties in evolutionary algorithms [Deb1997, Ree2003]. It uses metaphors from nature such as peaks, hills, valleys, ridges, basins, watersheds etc. to describe the characteristics of a search space that EAs might encounter when exploring it. Even for real-world problems, of generally much higher dimensionality than 2 and 3, the study of such metaphors in the higher dimensional space is still useful. It helps to understand the properties of the search space, such as its ruggedness (i.e. how each function value is correlated with the function values of neighbouring points) and modality (i.e. whether it has one or many local optima). By acquiring and exploiting knowledge about the fitness landscape of the problem, an EA can improve its search performance [Deb1997, Ree2003].

In practice, the concept of fitness landscapes is somewhat vague. As noted in [Ree2003], it has even been misused in the literature to describe fitness functions. Although the fitness function, which is problem dependent, is a defining component of fitness landscape, the topology of the search space must also be defined. In fact, one should not use the term “landscape” before defining the topological structure on the search space [Ree2003]. The topology of a search space in evolutionary computation consists of a genotype representation and a neighbourhood structure defined on it [Ree2003, Vas2000, Vas2003]. The neighbourhood structure is usually defined through a distance metric d on the genotype space [Ree2003]. A genotype l_2 is in the neighborhood of a genotype l_1 if and only if $d(l_1, l_2) < \delta$; when $\delta = 1$ (minimal distance), l_2 is called a neighboring point of l_1 . The third component of a fitness landscape is the genetic (search)

operators [Ree2003, Vas2000]. In order to investigate the fitness landscape, it is important to be able to “walk” on the genotype space, i.e. to transform one point to a neighbour in the search space. To properly investigate the characteristics of the fitness landscape, the genetic operators need to be consistent with the topological structure of the genotype (search) space [Ree2003]. In other words, the operators should transform a point ideally to its neighbour, or at least within its neighbourhood. In order to ensure that the genetic (search) operators were consistent with the metric defined on the genotype (search) space, some researchers ([Jon1995a, Jon1995b, Vas1999]) defined the metric in terms of the operators. However, the use of genetic operators to define the genotype metric makes the fitness landscape dependent on the operators used, different genetic operators giving different measures of the fitness landscape of a problem [Jon1995b]. On the other hand, the tree-edit distance metric defined in chapter 3 gives a natural distance metric on GP search spaces, independent of particular operators. Gaining an understanding of the fitness landscape with this underlying metric can help to design appropriate operators. Some recent fitness landscape studies in genetic programming have already followed this approach [Van2003a, Van2003b].

9.2 Problems with GP and GGGP in Fitness Landscape Study

In GAs, the task of defining a distance metric on the search space, and designing operators consistent with it, is not particularly challenging. For instance, with linear binary representation and using Hamming distance (the total number of different bits between two strings), one-point mutation is an obvious candidate. In genetic programming, things are more complicated because of the variability in shape and size. As pointed out in chapter 2, the expression tree in GP, and the CFG derivation tree of GGGP, make it difficult to design operators that make small, bounded changes. A similar conclusion was reached in [Lan2002] (page 24):

... However, often there is no natural ordering for the structures in fitness landscape. Also, even if there is one, the action of a search operator may be such that it does not respect the natural neighbourhood relationship available for the domain; i.e the operator may allow multi-step jumps...

In [Kin1994b], the autocorrelation function from random walks (described in appendix C) was used to analyze the fitness landscape on expression tree representation for a number of standard problems in GP. The results were somewhat inconclusive, the autocorrelation being blamed as an inaccurate estimator of fitness landscape smoothness. However, the problem might also lie in the disruptiveness of the operators Kinnear used, namely subtree crossover, subtree mutation, and hoist mutation. The operators take long step jumps, ignoring the natural topological structure on expression trees. For instance, in the previous chapter, it seems clear that the disruptiveness of subtree crossover on GP expression trees is an important factor in Daida's problem of structural difficulty in GP.

Recently, [Van2003a, Van2003b] proposed some smaller scale operators on GP expression tree representation, namely the inflate and deflate structural mutation operators. The operators were shown to be consistent with the tree alignment metric (mentioned in chapter 3 of the thesis); and some fitness landscape analysis was conducted using fitness distance correlation techniques. However the operators rely on incrementing and decrementing the arity of GP primitives, so they are inapplicable when all functions have the same arity. This difficulty is all the more pressing in syntactically constrained domains, where it is difficult to imagine how the operators might be made consistent with the grammar constraints.

For linear representations such as GEP, it is possible to use distance metrics and operators from GAs. However, if the representation does not have the locality property (as in GEP), the fitness landscapes study might tell us the characteristics of the genotype (search) space, but provide no guidance to the characteristics of the original (phenotype) space. In other words, although it can help to understand the search difficulty of the problem under the GEP repre-

sentation transformation, it does not help us to understand the difficulty of the original problem. As noted in [Alb2000, Lan2002], the change of representation using a genotype-phenotype mapping can change the fitness landscape significantly. [Rot2002] showed that this only applies to genotype-phenotype mappings lacking the locality property. Mappings satisfying the locality property do not change the fitness landscape much, since the neighborhood structure is preserved (i.e. if l_1 is in a neighborhood of l_2 then $f(l_1)$ is in a neighborhood of $f(l_2)$, where f is the transformation from genotypes to phenotypes). It is a direct consequence of “small changes in genotypes resulting in small changes in phenotypes” (locality property) [Pal1994b].

In the field of grammar guided genetic programming, where the problem domains are further constrained by grammar rules, it is even harder to design operators that make small and bounded changes. Whigham proposed the study of fitness landscape ([Whi1996] chapter 7) on syntactically constrained domains, so as to understand the impact of biasing the language of programs on the landscape. To the best of our knowledge, there has been no subsequent work on fitness landscape studies for syntactically constrained domains. For grammar guided genetic programming systems with linear representation such as GE, it is easy to define genotype metrics and consistent genetic operators. However if the representation does not possess the locality property (as with GE), the same problems arise as with GEP above. Moreover, in some circumstances, infeasible phenotypes can result from valid genotypes in GE, potentially creating serious difficulties in making a meaningful study of GE fitness landscapes, especially when there is a variety of infeasible phenotypes.

9.3 TAG-based Representation and Fitness Landscape Study

As mentioned in chapter 4 of the thesis, thanks to the non-fixed arity property, it is possible to design operators that make small and bounded changes on the

genotype level. Moreover, since the mapping from genotype space to phenotype space has the locality property as proven in chapter 3, if the change on the genotype is small and bounded, it is so on the phenotype space.

The topological structure chosen for the genotype as well as the phenotype spaces is the topology induced by the tree-edit distance described in chapter 3. A complete set of operators respecting that topological structure on the genotype space consists of insertion, deletion, and point replacement described in chapter 4 and 8. This is formulated in the following proposition.

Proposition 9.1. *Insertion, deletion, and point mutation make minimal change on the genotype space (G_{lex} derivation trees) and bounded change on the phenotype space (G_{lex} derived trees).*

Proof. From the definition of tree-edit distance, for any of the three operators o , and for any TAG-derivation tree t , $d(t, o(t)) = 1$, where d is the tree-editing distance. If in theorem 3.2 of chapter 3, we replace d by 1, then it follows that $d(f(t), f(o(t))) \leq M$, where M is the maximal number of nodes in an elementary tree, and f is the genotype to phenotype mapping.

The three operators are complete in that they provide a path from any labeled tree to any other labeled tree [Pol1990]. Using the three operators with equal probability of application, one can walk from any point to any other point both in the genotype space (TAG-derivation tree) and in the phenotype space (since the genotype-phenotype mapping f is onto) of TAG-based representation. This makes these operators ideally suited to study fitness landscape on syntactically constrained domains.

9.4 Experiments

In this section, the triple operators (deletion, insertion, and point replacement) are used to create a random walk for fitness landscape studies on some syntactically constrained domains. At each step of the random walk, one of the three operators is chosen with equal probability (1/3).

The experiments in this section are used both to understand how the fitness landscape changes as the structural complexity of the target functions is varied, and to study the effects of changing grammars on the fitness landscape while retaining a fixed target function.

In standard GA problems, the structural complexity is fixed, so that any difference in difficulty between two problems must arise from their fitness landscapes. In GP, the structural complexity is variable; more complex target functions are harder to generate simply because there is more to learn. So there are two potential sources of differences in problem difficulty for GP, namely structural complexity and ruggedness of the fitness landscape.

In chapter 5, a family of polynomial functions (F_1 to F_4), with increasing structural difficulty, was used to test the robustness of TAG3P compared with GP, and CFG-GP systems. The results showed that as we pass from F_1 to F_4 , the problems become increasingly difficult for TAG3P and CFG-GP. Does this performance degradation stem purely from the increasing structural complexity of the target functions, or also from an increase in ruggedness of the fitness landscape? In the first set of experiments, the same grammars (G and G_{lex}) were used for all target functions. For the symbolic regression problem, where the task is to learn the target function from sampled data, the structure is not used in the fitness calculation. However, as indicated by Daida (chapter 8), structure alone could influence problem difficulty. A fitness landscape study of these symbolic regression problems, with fixed grammars but varying the target function from F_1 to F_4 , could cast light on the hypothesis in chapter 5 that the increasing difficulty of the symbolic regression problem from F_1 to F_4 stems from the increase in structural complexity of the target functions.

One of the generally-acknowledged advantages of the use of grammars in genetic programming is their ability to set a declarative bias on the search space. By changing the grammar, one can bias the search system towards a particular region of the search space. [Whi1995b, Whi1996] includes a study showing how incorporating increasing knowledge into the grammar can improve the perfor-

mance of CFG-GP. The test problem used was the 6-multiplexer problem, four grammars (G_1, G_2, G_3 , and G_4) being used to define four different levels of declarative bias on the search space. The grammars are as follows ([Whi1996], pages 59-61).

$G_1 = (\Sigma, N_1, P_1, S)$, $\Sigma = \{a_0, a_1, d_0, d_1, d_2, d_3\}$, $N = \{S, B\}$, and the rule set P_1 is

P_1 :

$S \rightarrow B$

$B \rightarrow \text{if } B B B$

$B \rightarrow B \text{ and } B$

$B \rightarrow B \text{ or } B$

$B \rightarrow \text{not } B$

$B \rightarrow a_0|a_1|d_0|d_1|d_2|d_3$

$G_2 = (\Sigma, N_2, P_2, S)$, Σ is the same as in G_1 , $N = \{S, B, ADDRESS\}$, and the rule set P_2 is

P_2 :

$S \rightarrow \text{if } ADDRESS B B$

$ADDRESS \rightarrow a_0|a_1 B \rightarrow \text{if } B B B$

$B \rightarrow B \text{ and } B$

$B \rightarrow B \text{ or } B$

$B \rightarrow \text{not } B$

$B \rightarrow a_0|a_1|d_0|d_1|d_2|d_3$

$G_3 = (\Sigma, N_3, P_3, S)$, Σ is the same as in G_1 , $N = \{S, B, ADDRESS, IFTHEN\}$, and the rule set P_3 is

P_3 :

$S \rightarrow \text{if } ADDRESS IFTHEN B$

$ADDRESS \rightarrow a_0|a_1 IFTHEN \rightarrow \text{if } B B B$

$B \rightarrow \text{if } B B B$

$B \rightarrow B$ and B

$B \rightarrow B$ or B

$B \rightarrow$ not B

$B \rightarrow a_0|a_1|d_0|d_1|d_2|d_3$

$G_4 = (\Sigma, N_3, P_3, S)$, Σ is the same as in G_1 , $N = \{S, B, IFA1\}$, and the rule set P_4 is

P_2 :

$S \rightarrow$ if a_0 IFA1 B

IFA1 \rightarrow if a_1 B B

$B \rightarrow$ if B B B

$B \rightarrow B$ and B

$B \rightarrow B$ or B

$B \rightarrow$ not B

$B \rightarrow a_0|a_1|d_0|d_1|d_2|d_3$

The corresponding TAGs (G_{lex}) derived from them are given in Appendix A of the thesis. From the grammar descriptions, it can be seen that G_1 , which is similar to the grammar in chapter 5, is the most general. It generates all Boolean functions that can be formed from the function symbols *if*, *and*, *or*, *not* and the Boolean variables $a_0, a_1, d_0, d_1, d_2, d_3$. The functions generated by G_2 are a subset of those generated by G_1 . G_2 only generates functions starting with an if-then-else statement, with the condition part being an address variable (a_0 or a_1). G_3 is biased even further than G_2 since functions are restricted to start with two nested if-then-else statements, and the condition of the first *if* must be an address variable. G_4 incorporates the strongest bias of all, only generating expressions beginning with two nested if-then-else statements whose condition parts are address variables. As one moves from G_1 to G_4 , the search space is biased towards smaller and smaller subspaces. The significant reduction in search space size was argued as the reason for the problem becoming easier as the bias changed from G_1 to G_4 [Whi1995b, Whi1996]. Note that even though

the grammar changed, the target function in these experiments was fixed.

However apart from the reduction in search space size, the change of grammar might also change the characteristics of the fitness landscape, through changing the function representation. It is important to decouple these two effects (search space size and fitness landscape). Our second set of experiments conducts a fitness landscape study on the effect of changing the grammars while fixing the target function for this problem.

9.4.1 Experiment Setup

Two experiments were conducted for two problems, symbolic regression and 6-multiplexer. For the first experiment, the grammar is taken from chapter 5, while the target function is varied from F_1 (cubic polynomial function) to F_4 (polynomial function of degree 6 - see chapter 5 for more details). The aim was to determine whether increasing problem difficulty might stem partly from changing fitness landscape, as well as from the increasing structural complexity of the target functions. In the second experiment, the analysis aimed to determine whether decreasing problem difficulty with more specific grammars might arise partly from changing fitness landscape, or whether it was purely a consequence of the reduction in search space size reduction. The problem chosen was the 6-multiplexer and the grammars were G_1 ($G1_{lex}$), G_2 ($G2_{lex}$), G_3 ($G3_{lex}$), and G_4 ($G4_{lex}$).

For each experiment, a random walk of 10,000 steps was conducted using the triple operators (insertion, deletion, and point replacement, with equal probability) to create the walk. All fitness values of individuals encountered during the random walk were recorded. For each problem, 300 random walks were conducted, making a total of 3,000,000 fitness evaluations for each experiment. The data were then analysed using two common techniques from the literature, namely the autocorrelation function and the information content. These techniques are described in Appendix C.

9.4.2 Results and Discussion

Tables 9.1 and 9.2 show the autocorrelation values and the information content of the random walks for the symbolic regression problem – Length is the correlation length; Optima No. is the number of optima encountered during the random walks)

Table 9.1. Autocorrelation analysis for symbolic regression problem.

Length	F_1	F_2	F_3	F_4
1	0.7892 ± 0.0798	0.8002 ± 0.0715	0.7964 ± 0.0703	0.8025 ± 0.0717
2	0.6470 ± 0.1019	0.6620 ± 0.1046	0.6528 ± 0.1025	0.6656 ± 0.1059
3	0.5390 ± 0.1241	0.5561 ± 0.1213	0.5429 ± 0.1216	0.5626 ± 0.1243
4	0.4572 ± 0.1293	0.4750 ± 0.13	0.4593 ± 0.13	0.4819 ± 0.1337
5	0.3924 ± 0.1314	0.4099 ± 0.1332	0.3932 ± 0.1347	0.4157 ± 0.1388
6	0.3409 ± 0.13	0.3570 ± 0.1333	0.3407 ± 0.1354	0.3624 ± 0.14
7	0.2973 ± 0.1280	0.3145 ± 0.1306	0.2976 ± 0.1344	0.3185 ± 0.1390
8	0.2609 ± 0.1254	0.2794 ± 0.1271	0.2619 ± 0.1314	0.2811 ± 0.1376
9	0.2310 ± 0.1209	0.2488 ± 0.1237	0.2328 ± 0.1282	0.2502 ± 0.1339
10	0.2053 ± 0.11157	0.2224 ± 0.1201	0.2086 ± 0.1237	0.2243 ± 0.1309

Table 9.2. Information content analysis for symbolic regression problem.

Function	ϵ	$H(\epsilon)$	$h(\epsilon)$	$M(\epsilon)$	Optima No.
F_1	0	0.5920 ± 0.0137	0.6708 ± 0.0112	0.5071 ± 0.0075	2535 ± 37
F_2		0.5911 ± 0.0142	0.6706 ± 0.0106	0.5064 ± 0.007	2531 ± 35
F_3		0.5909 ± 0.0135	0.6711 ± 0.0108	0.5059 ± 0.0068	2529 ± 33
F_4		0.5912 ± 0.0146	0.6714 ± 0.0105	0.5062 ± 0.0067	2530 ± 33
F_1	1	0.723 ± 0.182	0.6882 ± 0.0111	0.4571 ± 0.0125	2285 ± 63
F_2		0.7235 ± 0.0189	0.6895 ± 0.112	0.4555 ± 0.0126	2276 ± 63
F_3		0.7247 ± 0.0178	0.6891 ± 0.0114	0.455 ± 0.0125	2274 ± 62
F_4		0.7253 ± 0.0183	0.6895 ± 0.0115	0.4546 ± 0.0128	2272 ± 64
F_1	2	0.7588 ± 0.0143	0.6945 ± 0.0118	0.4276 ± 0.0138	2137 ± 69
F_2		0.7571 ± 0.0144	0.6977 ± 0.0119	0.4256 ± 0.0139	2127 ± 70
F_3		0.7580 ± 0.0143	0.6968 ± 0.0122	0.4253 ± 0.0143	2126 ± 71
F_4		0.7584 ± 0.0148	0.6976 ± 0.0125	0.4239 ± 0.0141	2119 ± 70
F_1	13	0.6132 ± 0.0282	0.4756 ± 0.0317	0.186 ± 0.0161	930 ± 80
F_2		0.6139 ± 0.0277	0.4785 ± 0.0307	0.1869 ± 0.0163	934 ± 81
F_3		0.6113 ± 0.0295	0.4764 ± 0.0326	0.1857 ± 0.0172	927 ± 86
F_4		0.6077 ± 0.0302	0.4736 ± 0.0348	0.1841 ± 0.0177	920 ± 88
F_1	22	0.4888 ± 0.0364	0.3495 ± 0.0367	0.1293 ± 0.0135	646 ± 67
F_2		0.4928 ± 0.0358	0.3547 ± 0.0353	0.1307 ± 0.0129	653 ± 64
F_3		0.4882 ± 0.0365	0.3509 ± 0.036	0.1290 ± 0.0133	644 ± 66
F_4		0.4866 ± 0.0349	0.3489 ± 0.0353	0.1286 ± 0.0127	642 ± 63
F_1	71	0.2660 ± 0.0337	0.1737 ± 0.0303	0.0579 ± 0.0097	289 ± 48
F_2		0.1609 ± 0.0339	0.1763 ± 0.03	0.0585 ± 0.0098	292 ± 49
F_3		0.2646 ± 0.0344	0.1741 ± 0.0291	0.0577 ± 0.0096	288 ± 48
F_4		0.2650 ± 0.0335	0.174 ± 0.0297	0.058 ± 0.0097	289 ± 47

The results in Tables 9.1 and 9.2 clearly show that the characteristics of the fitness landscape vary only slightly when the target (learning) function is changed from F_1 to F_4 . In fact, the autocorrelation results suggest that the fitness landscape may become slightly smoother from F_1 to F_4 , though the change is not statistically significant. It is not entirely surprising. On the interval of interest all four functions have similar values, as depicted in Figure 9.1. When sampling the values of the function to build the fitness cases, the fitness cases are likely to be similar. Thus if an individual approximates one function well, it is likely to approximate the other three as well. Thus it appears that the increasing difficulties of the problem instances stem directly from the increasing structural complexity of the target functions, rather than from a change in the ruggedness of the fitness landscape, supporting the arguments in chapter 5 regarding the scalable difficulty of the families of polynomial functions. However this discussion is not regarded as conclusive; future work, detailed at the end of the chapter, is intended to provide more evidence for that argument.

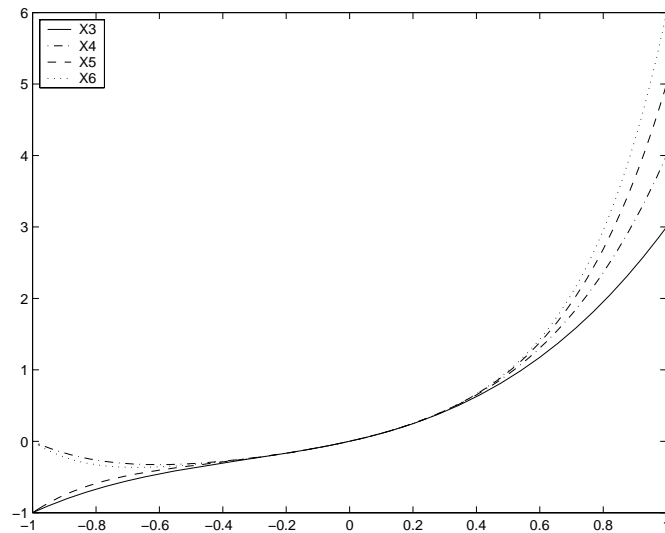


Figure 9.1: Graph of functions $F_1(X3)$ to $F_4(X6)$

For the second experiment, Table 9.3 and 9.4 show the autocorrelation values and information content for the random walks for the 6-Multiplexer problem.

Table 9.3. Autocorrelation analysis for 6-Multiplexer problem.

Length	G_1	G_2	G_3	G_4
1	0.8640 ± 0.0325	0.8899 ± 0.0242	0.901 ± 0.0193	0.9057 ± 0.0208
2	0.7719 ± 0.0450	0.8133 ± 0.0372	0.8291 ± 0.0301	0.8361 ± 0.0334
3	0.7131 ± 0.0531	0.7531 ± 0.0459	0.7715 ± 0.0387	0.7797 ± 0.0428
4	0.6603 ± 0.0581	0.7041 ± 0.0519	0.7236 ± 0.0048	0.7326 ± 0.0496
5	0.6165 ± 0.0618	0.6656 ± 0.0561	0.6826 ± 0.0497	0.6924 ± 0.0551
6	0.5796 ± 0.0644	0.6269 ± 0.0595	0.6473 ± 0.0537	0.6574 ± 0.0596
7	0.5475 ± 0.0665	0.5958 ± 0.0624	0.6165 ± 0.0570	0.6268 ± 0.0630
8	0.5193 ± 0.0677	0.5681 ± 0.0645	0.5888 ± 0.0598	0.5994 ± 0.0662
9	0.4945 ± 0.0686	0.5437 ± 0.0664	0.5642 ± 0.0621	0.575 ± 0.0688
10	0.4723 ± 0.0693	0.5216 ± 0.0681	0.5420 ± 0.0642	0.5533 ± 0.0712

Table 9.4. Information content analysis for 6-Multiplexer problem.

Function	ϵ	$H(\epsilon)$	$h(\epsilon)$	$M(\epsilon)$	Optima No.
G_1	0	0.6052 ± 0.05	0.4389 ± 0.0463	0.1832 ± 0.028	915 ± 140
G_2		0.5586 ± 0.0436	0.3898 ± 0.0404	0.1575 ± 0.0213	787 ± 106
G_3		0.5377 ± 0.0435	0.3708 ± 0.0416	0.1468 ± 0.0206	734 ± 103
G_4		0.5101 ± 0.045	0.3426 ± 0.0428	0.1342 ± 0.0195	670 ± 97
G_1	1	0.5179 ± 0.055	0.3451 ± 0.0504	0.1428 ± 0.0268	713 ± 134
G_2		0.4808 ± 0.0486	0.3099 ± 0.0436	0.1245 ± 0.0211	622 ± 105
G_3		0.4488 ± 0.0469	0.2804 ± 0.0408	0.110 ± 0.0186	554 ± 93
G_4		0.4268 ± 0.0483	0.2621 ± 0.0430	0.1023 ± 0.0183	511 ± 91
G_1	2	0.3652 ± 0.0507	0.1988 ± 0.0373	0.0849 ± 0.0156	424 ± 93
G_2		0.3445 ± 0.0473	0.1877 ± 0.0362	0.0763 ± 0.0162	381 ± 81
G_3		0.2959 ± 0.043	0.1513 ± 0.0307	0.0603 ± 0.0128	301 ± 64
G_4		0.2971 ± 0.0456	0.1535 ± 0.0340	0.0609 ± 0.0135	304 ± 67
G_1	8	0.0477 ± 0.0115	0.0136 ± 0.0039	0.0053 ± 0.0018	26 ± 8
G_2		0.0320 ± 0.0148	0.0086 ± 0.0147	0.0036 ± 0.0022	18 ± 11
G_3		0.0196 ± 0.01	0.0048 ± 0.0028	0.002 ± 0.0013	10 ± 6
G_4		0.0227 ± 0.0134	0.0058 ± 0.0041	0.0025 ± 0.0019	12 ± 9
G_1	13	0.017 ± 0.0072	0.004 ± 0.0019	0.0017 ± 0.0009	8 ± 4
G_2		0.0167 ± 0.0127	0.0041 ± 0.0032	0.0017 ± 0.0014	8 ± 7
G_3		0.0089 ± 0.0074	0.002 ± 0.0018	0.0009 ± 0.0008	4 ± 4
G_4		0.0122 ± 0.0104	0.0029 ± 0.0028	0.0012 ± 0.0012	6 ± 6
G_1	18	0.003 ± 0.0002	0.0006 ± 0.0005	0.0003 ± 0.0002	1 ± 1
G_2		0.0006 ± 0.0001	0.0001 ± 0.0002	0 ± 0	0 ± 0
G_3		0.0002 ± 0.0001	0 ± 0.0001	0 ± 0	0 ± 0
G_4		0.0002 ± 0.0001	0 ± 0.00001	0 ± 0	0 ± 0

The results of both autocorrelation analysis and information content analysis consistently show that the ruggedness of the fitness landscape decreases as the grammar changes from G_1 to G_4 , indicated by the increase in autocorrelation values and the decrease in information content (H) and partial information content (M). The statistical significance was tested using the one-tailed test for differences between two binomial variables, with confidence level $\alpha = 0.05$. For small autocorrelation length and small ϵ , when the analyses are most sensitive, the differences between G_1 and G_4 were found to be significant. Thus the bias induced by changing the grammar may have influenced the search not only by reducing in the search space size, but also by changing the characteristics of the fitness landscape. The improved performance of CFG-GP with the stronger biases was not solely due to the reduction in search space as claimed in [Whi1996], but also to the resulting smoothing of fitness landscape.

9.5 Conclusion

In this chapter, we argued that because of the fixed-arity tree structure of standard GP, and the even more constrained nature of GGGP derivation trees, it is difficult to define a topological structure, or to design operators that respect this structure. By transforming to the space of TAG-derivation trees, it is easier to define a natural topology and design operators that respect this topological structure, making small and bounded changes. Moreover, since the mapping from the genotype space to the phenotype space possesses the locality property, the changes on the phenotype space respect the corresponding topology in the phenotype space. Using insertion, deletion and point replacement, it is possible to investigate problem difficulty through characterising the fitness landscape on the genotype space, and in effect also on the original (phenotype) search space.

We showed some experiments which, for the first time in the literature, characterise the fitness landscape on a syntactically constrained domain. We were able to distinguish the effects of fitness landscape and of structural complexity on the

problem difficulty. In addition, we investigated the effect on fitness landscape of changing search space bias by changing grammar, showing that the changing bias smoothes the fitness landscape in addition to reducing the search space. The results have shed some light on the performance of CFG-GP, as well as TAG3P, on those problems.

In the fitness landscape study for symbolic regression, future work includes a study of autocorrelation near the optima. With the fitness function used for these symbolic regression problems (total error over 20 sampling points), the search landscapes far from the optima are likely to be very similar for all functions, because most randomly sampled individuals are likely to be very far from any of the functions, so that the differences between the functions are masked. The similar autocorrelation values may stem simply from most sampled individuals being very far from the optima; but GP runs for these symbolic regression problems almost invariably converge to a small error within a few generations, so that most of the search is conducted close to the optima, and fitness landscapes far from the optima give a poor indication of the overall problem difficulty. There are some indications from the similar basin-density information values (h) in the information content analysis that the fitness landscapes may also be similarly close to the optima, but further work is needed to confirm this.