The K Landscapes: a Tunably Difficult Benchmark for Genetic Programming

Leonardo Vanneschi Mauro Castelli Luca Manzoni

Department of Informatics, Systems and Communication (D.I.S.Co.)
University of Milano-Bicocca
Milan, 20126, Italy
{vanneschi,mauro.castelli,luca.manzoni}@disco.unimib.it

ABSTRACT

The NK landscapes are a well known benchmark for genetic algorithms (GAs) in which it is possible to tune the ruggedness of the fitness landscape by simply modifying the value of a parameter K. They have successfully been used in many theoretical studies, allowing researchers to discover interesting properties of the GAs dynamics in presence of rugged landscapes. A similar benchmark does not exist for genetic programming (GP) yet. Nevertheless, during the EuroGP conference debates of the last few years, the necessity of defining new benchmark problems for GP has repeatedly been expressed by a large part of the attendees. This paper is intended to fill this gap, by introducing an extension of the NK landscapes to tree based GP, that we call K landscapes. In this benchmark, epistasis are expressed as growing mutual interactions between the substructures of a tree as the parameter K increases. The fact that the problem becomes more and more difficult as the value of K increases is experimentally demonstrated. Interestingly, we also show that GP "bloats" more and more as K increases.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming

General Terms

Algorithms, Performance

Keywords

Genetic Programming, Benchmarks, Problem Difficulty, Epistasis

1. INTRODUCTION

As already pointed out by Altenberg in 1997 [1], in the first part of the twentieth-century Wright [21] discovered an interesting feature of evolutionary dynamics: when the effect on fitness from altering the state of one gene depends on the state of other genes, the population can often evolve into multiple basins of attraction. This kind of interaction between genes is called *epistasis*. In other words, the presence of epistasis makes it possible for a population to converge towards different genotypic configurations, depending on its initial state. Thus, it is possible to indentify two different levels of abstraction, that are different to, but dependent from, each other: the microscopic level (the level of the single individuals, in which fitness depends on the reciprocal interactions between genes) and the macroscopic one (the level of populations, where it is possible to identify multiple different attractors). To clarify these ideas, Wright brought up the analogy with a landscape with multiple peaks, in which the evolution of a population can be seen as the movement up hill of its individuals, until they reach a (local or global) fitness peak. The term "adaptive landscape" or "fitness landscape" is nowadays frequently used to describe the presence of multiple basins of attraction in the space of genotypes for evolutionary dynamics. Under this perspective, it becomes interesting to investigate how a population can escape from a local fitness peak. Wright proposed the idea of stochastic fluctuations, thus introducing an embryonic version of a stochastic process for the optimization of multimodal functions. More recently, based on the concepts introduced by Wright, Kauffman proposed the NK landscapes set of functions, to investigate the way epistasis controls the number of local peaks of a fitness landscape [8, 9]. In these functions, the ruggedness of the fitness landscape can be controlled by modifying a single parameter that influences the epistatic level of the genome. Thus, the NK landscapes is a stochastic set of tunably difficult optimization problems. Described in Section 2, the NK landscapes have often been used as a benchmark for theoretical studies of Genetic Algorithms (GAs) [7, 5].

NK landscapes are based on the idea that potential solutions are represented as fixed length strings of symbols, and thus they are particularly suitable for GAs. To the best of our knowledge, no extension to this problem has been proposed so far for Genetic Programming (GP) [11, 15], where individuals are characterized by a dynamic size representation. Nevertheless, as clearly stated by the panel members and attendees of the EuroGP 2008 debate on Grand Challenges of GP (which took place on 27 March 2008 at the Evo* event in Naples, and whose main ideas have recently been developed and published in [13]), and as reasserted during the subsequent EuroGP 2009 and EuroGP 2010 debates, the GP community has the pressing necessity of defining new and reliable benchmarks. Those benchmarks should possibly be of tunable difficulty, thus allowing practitioners and theoreticians to study the dynamics of GP for difficult, as well as easier problems. These benchmarks should, above all, be used to study and discover new properties of the GP method itself, as it has been the case for NK landscapes in GAs.

Indeed, some efforts have already been done in the direction of

GECCO'11, July 12–16, 2011, Dublin, Ireland.

defining useful benchmarks for GP (they are reviewed in Section 3), but still the goal of having many rigorous test problems is far from being achieved, probably due to the larger complexity of GP compared to GAs or other methods for parameters optimization. This paper wants to be a contribution in this direction; in fact, we extend the NK landscapes problem to GP. The new GP benchmark that we introduce is called K landscapes[1]. Its definition, together with a first experimental study aimed at showing the GP behavior on this benchmark, is contained in the continuation of the paper.

The paper is organized as follows: Section 2 contains a description of the NK landscapes for GAs. In Section 3 we revise previous contributions in which theoretically hand-tailored benchmarks for GP have been introduced. In Section 4 we describe the K landscapes benchmark for GP proposed in this paper. In Section 5, we present our experiments, including both a description of the used experimental settings and a discussion of the obtained results. Finally, Section 6 concludes the paper.

2. THE NK LANDSCAPES FOR GAs

The NK family of landscapes, introduced by Kauffman between the end of the eighties and the first part of the nineties [8, 9], is a problem-independent model for constructing multimodal landscapes for GAs that can gradually be tuned from smooth to rugged. In the model, N refers to the number of (binary) genes in the genotype (i.e. the string length) and K to the number of genes that influence a particular gene (the epistatic interactions). By increasing the value of K from 0 to $N-1$, NK landscapes can be tuned from smooth to rugged. The fitness of a NK landscape is a function $f_{NK} : \{0,1\}^N \rightarrow [0,1)$, defined on binary strings with N bits. An "atom" with fixed epistasis level is represented by a fitness component $f_i : \{0,1\}^{K+1} \rightarrow [0,1)$ associated to each bit i. Its value depends on the allele at bit i and also on the alleles at K other epistatic positions. (K must fall between 0 and $N-1$). The fitness $f_{NK}(s)$ of $s \in \{0,1\}^N$ is the average of the values of the N fitness components f_i: $f_{NK}(s) = \frac{1}{N}\sum_{i=1}^{N} f_i(s_i, s_{i_1}, \ldots, s_{i_K})$, where $\{i_1, \ldots, i_K\} \subset \{1, \ldots, i-1, i+1, \ldots, N\}$. Many ways have been proposed to choose the K other bits from the N bits forming the bit string. Two possibilities are mainly used: adjacent and random neighborhoods. With an adjacent neighborhood, the K nearest bits to the bit i are chosen. With a random neighborhood, the K bits are chosen randomly on the bit string. Each fitness component f_i is specified by extension, *i.e.* a number $y^i_{s_i, s_{i_1}, \ldots, s_{i_K}}$ from $[0,1)$ is associated with each element $(s_i, s_{i_1}, \ldots, s_{i_K})$ from $\{0,1\}^{K+1}$. Those numbers are usually random numbers, uniformly distributed in the range $[0,1)$.

As reported by Kauffman, epistasis have a repercussion on fitness that is similar to a *house of cards* [10]: if a bit is modified in a given position, all the fitness components that interact with it are changed, without any correlation with their previous values. Thus, the modification of one single bit, for instance by the application of a bit-flip mutation operator, has on fitness the same disruptive effect as removing a card from a house of cards: to find the same fitness value again (i.e. to build the same house of cards once again) one has to restart from scratch and it is not possible to use any of the previously processed information for doing it. Thus, any algorithm that works by trying to optimize all genes at the same time clearly encounters serious problems in this kind of benchmark. The interested reader is referred to [1] for a survey on NK landscapes, where some properties of this benchmark, like the computational complexity, as well as some implementation details and alternative variants are discussed.

[1]The "N" in the NK landscapes model represents the length of the genome codifying GAs individuals. This does not make sense in GP, where individuals have typically a variable sized representation. For this reason, we just use the term "K" landscapes to indicate the proposed benchmark.

3. PREVIOUS GP BENCHMARKS

In Koza's 1992 book [11], several problems that can be solved with GP are defined: k-even parity, h-multiplexer, various forms of symbolic regression, the artificial ant on the Santa Fe trail, the intertwined spirals problem, etc. For many years, and with few exceptions, those problems have mainly represented the only benchmarks that have been used in GP experimental studies. Only recently, the GP community has begun to use a larger set of test functions. It is the case, for instance, of the UCI repository datasets suite [2]. Among them, one can identify problems of different complexities, from trivial ones, like the IRIS dataset, to more difficult ones, like the thyroid cancer datasets and many others. Also, contributions have appeared using as test problems applications like classification of network intrusion (see for instance [17, 14]). As stated in [13], we could broadly classify currently used test function in GP into three categories: regression, classification and design. Typical used regression functions are sinus, polynomials, mexican hat, Mackey-Glass time series, etc. For classification, one may quote the UCI examples, the intertwined spirals, the parity problems, the multiplexer, Protein Localization, etc. The class of test problems classified as design contain applications like adder, multiplier, several other circuits, trusses, tables and other structures.

All the above quoted test problems, from the early ones introduced by Koza in 1992 to the more recent ones, are definitely interesting, because they cover a set of different possible applications. Nevertheless, it is also true that compared to the set of typical test problems used for other optimization methods, like for instance GAs or particle swarm optimization, this set of benchmarks is still a restricted one and, even more importantly, lacks a rigorous evaluation. In particular, the majority of these problems have their own, so to say, "fixed" complexity: it is not possible to define several instances of them, of different difficulties (from very easy problems to very hard ones, and including many intermediary cases) by changing some parameters (this is partially false, for instance, for the even parity and multiplexer problems, as already pointed out in [20], but it is surely true for all the problems that consist in mining a given dataset like, for instance, the UCI ones). Furthermore, as remarked in [13], the above quoted set of test problems is composed by functions of similar nature, and thus they are too similar to each other. For instance, these functions would require too a restricted set of operators and terminals to build individuals, compared to the huge variety of possibilities offered by GP.

Some attempts of defining new and tunably difficult test problems for GP have been made so far. One of the earliest ones probably consists in the introduction of the Royal Trees by Punch and coworkers [16]. The goal of that contribution was to devise a problem for GP that could be similar to, and share interesting properties with, the the Royal Road problem for GAs [12]. In particular, the Royal Trees are an example of "constructional problems", in the same vein as the Royal Roads, a concept that had already been introduced in GP by Tackett [18]. More or less in the same vein, contributions [20, 19] contain the extension of tunably difficult problems typical of GAs, like various forms of trap functions [4], to GP. In [6], Gustafson and coworkers introduced the Tree-String problem. The goal of this problem is to derive specific structure and content elements simultaneously. Instances are defined using a target solution consisting of a tree shape and content. Candidate solutions are then measured for their similarity to the target solution

with respect to both tree shape and content objectives. Separating these two concepts, the Tree-String problem makes thus possible to shade a light on the complex dynamics created by the interdependencies of solution structures and contents. Furthermore, the authors show how the difficulty of the Tree-String problem can be tuned by simply modifying the number of used nodes and the size of the alphabet employed to code contents. Another interesting contribution, that is related to, but slightly different from, the previously quoted ones, is [3], where Daida and coworkers introduce the Lid problem. The Lid problem is tunably difficult, but, contrarily to Royal Trees, trap functions, the K landscapes presented here, and others, it has this property because tuning is accomplished through changes in structure. So, the difficulty of the problem, and the possibility to tune it, depend on structural mechanisms and not on the fitness landscape. Daida's work on the effect of structures on the GP search is definitely interesting, but it is outside the scope of this paper, that is instead based on the concept of fitness landscape. Nevertheless, the way structural mechanisms influence the difficulty of the K landscapes proposed here deserves to be studied and it will be one of the subjects of our future research.

4. THE K LANDSCAPES FOR GP

We denote a GP individual by T, the root of T by $R(T)$, the depth of T by $D(T)$ and the set of children of a node N by $S(N)$. The set of all nodes of T is denoted by $N(T)$. The set of all possible subtrees of T is denoted by $\Psi(T)$. The set of all functional symbols used to code individuals is denoted by $\mathcal{F}$ and the set of terminal symbols by $\mathcal{T}$.

Given two numbers $a, b \in \mathbb{R}$, with $a < b$, let $v : \mathcal{F} \cup \mathcal{T} \mapsto [a, b]$ be a function chosen randomly between all (representable) functions from $\mathcal{F} \cup \mathcal{T}$ to $[a, b]$. The map v returns a number in $[a, b]$ for each possible tree node, i.e. for each possible element of the set $\mathcal{F} \cup \mathcal{T}$. Also, given two numbers $c, d \in \mathbb{R}$ such that $c < d$, let $w : \mathcal{F} \times (\mathcal{F} \cup \mathcal{T}) \mapsto [c, d]$ be a randomly chosen function that returns a number in $[c, d]$ for each possible connection between two nodes in a tree. In this paper we choose $a = -1$, $b = 1$, $c = 0$ and $d = 1$ and from now on all the considerations will be done assuming these values for a, b, c and d. Choosing a negative value for a, we give to the v function the possibility of returning negative values. For a given $K \in \mathbb{N}$, with K smaller or equal to the maximum admissible depth for the trees in the population (if any maximum depth is imposed) and a tree T, we define:

$$f_K(T) = \begin{cases} v(R(T)) + \sum_{C \in S(R(T))} (1 + w(R(T), C)) f_{K-1}(C) & \\ & \text{if } K > 0 \\ v(R(T)) & \text{otherwise} \end{cases}$$

When $K = 0$, $f_K(T)$ simply returns the value of the v function on the root of T. In all the other cases, $f_K(T)$ returns a weighted sum of the values of v applied to the nodes of T for the first K levels, where the weights are determined by the values of w.

The fitness function that we propose for the GP K landscapes is defined as the maximum value of f_K calculated over all the nodes of a GP tree, with a penalty given by a function of the difference between the depth of the tree and K:

$$F_K(T) = \frac{1}{1 + |K - D(T)|} \max_{T' \in \Psi(T)} \{f_K(T')\}$$

The problem that we define introducing this fitness function is a maximization one (i.e. larger values of the fitness are better).

Now we want to enunciate and prove some properties of the K landscapes problem defined by such a fitness function. But before doing it, we need to introduce the following concept:

DEFINITION 1. *Given a tree T and a $K \in \mathbb{N}$, we indicate with the term* summit *of T of level K a structure T' that respects the following properties:*

- *if $K = 0$, T' is only composed by the node $R(T)$.*
- *otherwise, $R(T') = R(T)$ and if the subtrees of $R(T)$ in T are $S_1, S_2, ..., S_h$, then the subtrees of $R(T')$ in T' are $U_1, U_2, ..., U_h$, where for each $i = 1, 2, ..., h$ U_i is a summit of level $K - 1$ of S_i if S_i is different from the empty tree, and the empty tree otherwise.*

The concept of summit of a tree should be clarified by Figure 1: the structure in Figure 1(b) is a summit of the tree in Figure 1(a), while the one in Figure 1(c), even though it is a subtree of the tree in Figure 1(a), is not a summit of the tree in Figure 1(a). In very

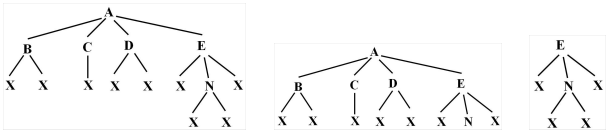


Figure 1: The structure in figure (b) is a summit of the tree in figure (a), while the one in figure (c) is not a summit of the tree in figure (a).

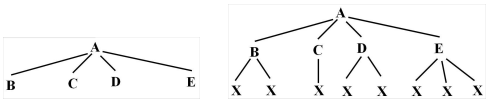
informal terms, we could say that the summit of a tree T is a "top part" of T: for each path from the root to a leaf in T, a summit of T contains a subset of this path, from the root to a given node up to a given prefixed level K, or the whole path if the length of the path is $\leq K$. It is also worth pointing out that a summit of level K of a tree T does not necessarily have a depth equal to K, but can also have a smaller depth than K if it is the whole tree T. Furthermore, it is immediate to remark that, if T is the representation of a GP individual, a summit of T may not represent any GP individual, given that it may lack some terminals (indeed, it can represent an individual only if it is equal to T itself). For instance, if we indicate with X a general terminal symbol, the summit in Figure 1(b) contains one branch that is not complete (it is not ended by a terminal symbol), and thus it cannot represent a GP individual. It is possible to "transform" a summit of a tree into a GP individual by adding terminal symbols at the last level, where needed. We call this process *embedding*[2]:

DEFINITION 2. *Given a summit T' of a tree T, an* embedding *of T' is a tree that is identical to T' except for the fact that terminal symbols are added in the branches that are not ended by terminal symbols in T'.*

Figure 2 should clarify the concept of embedding: the tree in Figure 2(b) is an embedding of the summit in Figure 2(a), given that a terminal symbol is added in all the branches that are not ended by a terminal symbol.

We are now ready to enunciate and prove some properties of the proposed K landscapes.

[2]An alternative, and maybe more suitable, term may be "extension", since with this process a tree is "extended" rather than "embedded" in another structure. Nevertheless, given the generality of the term "extension", we prefer to use the term "embedding".



(a) (b)

Figure 2: The tree in figure (b) is an embedding of the summit in figure (a).

PROPOSITION 1. *For each admissible value of K, if there exists a tree T such that $f_K(T) > 0$ then all the optima in the search space must necessarily have a depth of at most $K + 1$.*

Proof: Let us consider the definition of $f_K(T)$. It is composed by a sum of terms. To calculate each one of these terms, the nodes of T have to be considered from the root up to the level K, or from the root to the leaf for the paths where the leaf is at depth smaller than K. Let us consider the structure composed by all the nodes that have to be analyzed in order to calculate $f_K(T)$. It is clearly a summit of level K of T. This summit can be transformed into a tree T': if the summit is equal to T, then $T' = T$. Otherwise T' is an embedding of T. In the first case T' has a depth of at most K, in the second case T' has a depth equal to $K + 1$. It is interesting to remark that, in the first case, if the depth of T' is smaller than K, a tree T'' of depth K can be considered that has T' as one of its subtrees. By the definition of F_K, this new tree will have a better fitness value than T'. In fact, given that T' is a subtree of T'', we have that $\max_{S_1 \in \Psi(T'')}\{f_K(S_1)\} \geq \max_{S_2 \in \Psi(T')}\{f_K(S_2)\}$ and the term $\frac{1}{1+|K-D(T'')|}$ is equal to 1, while $\frac{1}{1+|K-D(T')|}$ is smaller than 1. Thus, we can say that for each tree T it is possible to generate a tree U that either has a depth equal to K or to $K + 1$: depending on the depth of the summit of T of level K, U can be equal to T itself, an embedding of its summit or a tree (like T'') that has its summit as a subtree.

Now we want to show that each tree that has a depth larger than $K + 1$ cannot be an optimum. Let T be a tree with a larger depth than $K + 1$. Let us consider the tree U obtained from T by applying the process described so far. Given that $f_K(T)$ is calculated using only the nodes that belong to the summit of level K of the subtrees of T, and given that the subtrees of T and the subtrees of U that maximize f_K have the same summit of level K, we have that $\max_{S_1 \in \Psi(T)}\{f_K(S_1)\} = \max_{S_2 \in \Psi(U)}\{f_K(S_2)\}$. But given that T has a larger depth than U and the depth of U is either K or $K + 1$, $\frac{1}{1+|K-D(T)|} < \frac{1}{1+|K-D(U)|}$. Thus the fitness of T must be smaller than the one of U. We conclude that T cannot be an optimum. ■

PROPOSITION 2. *For each $K > 0$, we have that $f_K(T) > 0$ for some tree T if and only if there exists $N \in \mathcal{F} \cup \mathcal{T}$ such that $v(N) > 0$.*

Proof: To prove this property, we prove separately both implications. First, let us prove that if $f_K(T) > 0$ for some tree T then $N \in \mathcal{F} \cup \mathcal{T}$ such that $v(N) > 0$ exists. The case $K = 0$ is obvious: if $f_K(T) > 0$ then v applied to the root of T must return a positive value. In the other cases, $f_K(T) > 0$ implies that at least one of the terms of the sum that defines f_K must be positive. f_K is defined as a sum of terms, where each term is composed by an application of function v to a node of T multiplied by an application of function w to a connection in T. Given that w can only return positive values, there must exist at least one $N \in N(T) \subseteq \mathcal{F} \cup \mathcal{T}$ such that $v(N) > 0$.

Let us now prove that if an $N \in \mathcal{F} \cup \mathcal{T}$ such that $v(N) > 0$ exists, then $f_K(T) > 0$ for some tree T. Let us assume first that there exists $N \in \mathcal{T}$ such that $v(N) > 0$. Then the tree having N as the only node has a value of f_K equal to $v(N) > 0$. Now let us assume that $N \in \mathcal{F}$ such that $v(N) > 0$ exists and let us consider a tree T of depth $K + 1$ with all the leaves at level $K + 1$ and all the internal nodes equal to N. The summit T' of level K rooted at $R(T)$ is composed only of nodes with weight $v(N)$. This means that $f_K(T) = f_K(T') \geqslant v(R(T)) > 0$. ■

We also remark that, when $f_K(T) \leqslant 0$ for all trees, and the tree depth is unbounded, for every tree it is always possible to find a tree with an higher fitness, even if this fitness is always negative. This is due to the penalty given by the term $\frac{1}{1+|K-D(T)|}$ that appears in the definition of F_K: as $D(T)$ increases, a negative value of the fitness increases too.

In our definition of the K landscapes, we want all the global optima to have a depth not larger than $K + 1$, because, as it will be clear shortly, this allows us to define an algorithm to compute the globally optimal fitness. Thus, for obtaining this goal, from now on we impose that the v function must assume a positive value for at least one element of its domain.

We are now ready to enunciate and prove the following property:

PROPOSITION 3. *At least one of the optima of the K landscapes is a tree T that has all the nodes at the same level identical to each other.*

The tree represented in Figure 3 is an example of a tree that respects the property of Proposition 3. In fact, level 1 is composed by only B symbols, level 2 is composed by only C symbols and level 3 is composed by only X symbols; thus only symbols that are identical to each other appear in the same level.

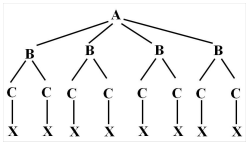


Figure 3: An example of a tree that respects the property of Proposition 3.

Now we prove Proposition 3:

Proof: Let us consider an optimum T for the K landscapes. Let $\Psi(T) = \{U_1, U_2, ..., U_h\}$ be the set of all the possible subtrees of T. For each $i = 1, 2, ..., h$, let U_i' be the tree of depth at most $K + 1$ obtained from U_i by considering its summit of level K and eventually embedding it (the process that allows us to generate U_i' from U_i is explained at the beginning of the proof of Proposition 1). Now, among the trees $U_1', U_2', ..., U_h'$, let us consider the one that maximizes f_K and let us call it T' (actually the nodes that compose the summit of level K of T' are the ones that are used to calculate the fitness of T). We call T' an f_K-optimum.

It is easy to convince oneself that Proposition 3 holds by looking at Figure 4: let us assume that the tree in Figure 4(a) is f_K-optimum. Let T_1 be the left subtree of the tree in Figure 4(a) and T_2 be its right subtree (T_1 and T_2 are the subtrees respectively surrounded by a rectangle and an oval in Figure 4(a)). First of all, we remark that the property $f_K(T_1) = f_K(T_2)$ must hold. In fact, (without loss of generality) let $f_K(T_1)$ be larger than $f_K(T_2)$. If this is true, it is possible to build a new tree that is identical to the one in Figure 4(a), but with the only difference that the occurrence of T_2 is replaced by another occurrence of T_1; this new tree, represented in Figure 4(b), would have a larger value of f_K

than the tree in Figure 4(a). But this would contradict the hypothesis that the tree in Figure 4(a) is f_K-optimum. We conclude that $f_K(T_1) = f_K(T_2)$ and also that the trees represented in Figures 4(b) and (c) have the same value of f_K as the one in Figure 4(a), and thus are also f_K-optima.

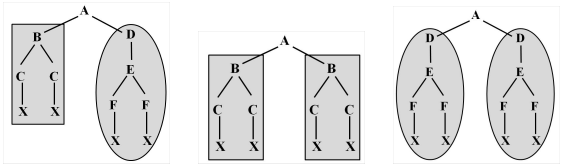


Figure 4: Suppose that (a) is a f_k-optimum. Then both (b) and (c) must also be f_k-optima (see the proof of Proposition 3).

This argument can be generalized, abstracting from the example of the trees represented in Figure 4: given an f_K-optimum, it is always possible to build another f_K-optimum that has all the subtrees of its root identical to each other (simply by "replicating" one of them, as it has been done for transforming the tree in Figure 4(a) into the tree in Figure 4(b)). In case these subtrees do not respect the property of Proposition 3, it is possible to iterate the process, transforming them into trees that have all the subtrees of the root identical to each other and that have the same value of f_K. Iterating this process until the last level of the tree, it is possible to build an f_K-optimum that respects the property of Proposition 3.

So far we have shown that, for every K, there exists an f_K-optimum that respects the property of Proposition 3. Now, we have two possibilities to define a candidate optimum for the K landscapes problem: (1) consider an f_K-optimum chosen between the trees of depth at most K. If it has depth K, then it is the candidate optimum. Otherwise, if its depth is smaller than K, we candidate a tree of depth K that respects the property of Proposition 3 and contains it as a subtree; (2) consider an f_K-optimum chosen between the trees of depth $K+1$ and that respects the property of Proposition 3. One optimum T of the K landscapes problem can be obtained either by possibility (1) or by possibility (2), because in both cases T contains an f_K-optimum and the value of the term $\frac{1}{1+|K-D(T)|}$ is either 1 or $\frac{1}{2}$. In both cases, it respects the property of Proposition 3 and this allows us to conclude. ■

So far, we have proven some properties that at least one optimum of the proposed K landscapes must have. Using these properties, we are now able to calculate the optimal fitness of this problem. The argument that we use is similar to the one used in the proof of Proposition 3. Let $T_{\leq K}$ be an f_K-optimum of depth at most K. Let T_{K+1} be an f_K-optimum of depth $K+1$. To find an optimum for F_K we can suppose that both $T_{\leq K}$ and T_{K+1} respect the property of Proposition 3. Also, it is possible to consider $T_{\leq K}$ of depth K since, if its depth is less than K, it is possible to define a deeper tree that has $T_{\leq K}$ as a subtree. Thus, by the definition of F_K, it is possible to express the K landscapes optimal fitness as follows:

PROPOSITION 4. *The optimal fitness of the K landscapes is:*

$$\begin{aligned} F_{opt} &= \max\{\tfrac{1}{1+|K-K|} f_K(T_{\leq K}), \tfrac{1}{1+|K-(K+1)|} f_k(T_{K+1})\} \\ &= \max\{f_K(T_{\leq K}), \tfrac{1}{2} f_K(T_{K+1})\} \end{aligned}$$

Furthermore, we are also able to define an algorithm to compute the optimal fitness and the structure of at least one of the optima: for all the tree depths form 0 to $K+1$, we simply exhaustively consider all the trees that respect the property of Proposition 3 (the number of these trees is much smaller than the number of all the trees of depth at most $K+1$), finding the one with the maximum fitness. Because of the properties that we have proven so far, we are sure that it will be one of the global optima. Even if the algorithm is exponential in K, it is usable in practice because K cannot be larger than the maximum depth allowed for the trees in the population. Thus, only "limited" values of K are used in practice.

We finally point out an interesting characteristic of the K landscapes problem introduced here: given a f_K-optimum of depth d, its subtrees of depth $d-1$ are *not necessarily* f_{K-1}-optima. This fact implies that it is difficult for GP to build an optimum starting from smaller sub-optima, as it is the case of the NK landscapes for GAs. The following example shows a case where the subtrees of depth $d-1$ of a f_K-optimum of depth d are not f_{K-1}-optima.

Example. Let us consider the following set of functional and terminal symbols used to build GP individuals: $\mathcal{F} = \{A\}$, $\mathcal{T} = \{X, Y\}$. Let us consider the following values of the v and w functions: $v(A) = 1$, $v(X) = 0.75$, $v(Y) = 1$; $w(A, X) = 1$, $w(A, Y) = 0$. It is clear that it is possible to build only two trees of depth 1 that respect the property of Proposition 3. They are shown in Figure 5. Let T_1 be the tree represented in Figure 5(a) and T_2

Figure 5: The only two trees of depth 1 that respect the property of Proposition 3 and that can be built using the sets of symbols $\mathcal{F} = \{A\}$, $\mathcal{T} = \{X, Y\}$.

the one represented in Figure 5(b). The value of f_1 for these two trees is: $f_1(T_1) = 1 + 2 * (1 + 1) * 0.75 = 1 + 3 = 4$ and $f_1(T_2) = 1 + 2 * 1 * 1 = 3$. Thus, T_1 is an f_1-optimum. But, given that $v(Y) > v(X)$ the f_0-optimum is not the tree composed by the only X symbol (subtrees of depth 0 of T_1), but the one composed by the only Y symbol.

5. EXPERIMENTAL STUDY

Experimental Settings. All the results presented in this section have been obtained performing 100 runs of GP with a population size of 100 individuals running for 100 generations. The initialization method was ramped half-and-half with an initial maximum tree depth of 2. The selection method used was tournament selection with a tournament size of 10. The crossover method used was the standard subtree swapping crossover [11] with a crossover probability of 0.9. The mutation method used was point mutation [11] with a probability of 0.1. During the experiments elitism (i.e. unchanged copy of the best individual in the next population at each generation) was used and the maximum allowed tree depth was 17. The experiments were performed using values for the K parameter ranging from 2 to 13. Individuals have been built using 2 functional symbols, both with arity 2, and 4 terminal symbols. Since the fitness is computed considering only the values of the v and w functions, it was unnecessary to define a semantic for the functional and terminal symbols used. At each generation the fitness, size and depth of the best individual were recorded. Also the average fitness, depth and size of the population were recorded. The median of these values over the 100 runs is reported in this section. A t-test has been performed over the fitness values obtained in the last generation for every value of K with the equality of the means considered as the null hypothesis.

Experimental Results. In all the figures from 7 to 11, different curves represent different values of K. For the same value of K we have used, in all these figures, the same line. For this reason, we report the legend of this figures only once, in Figure 6.

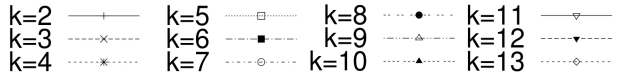


Figure 6: The legend for the plots reported in figures from 7 to 11.

In Figure 7 the median of the average depth of the GP trees in the population for the different values of K is presented. We can see that, for every considered K, the average depth after generation 20 is slightly above K. This result shows that the largest part of the search process is concentrated on trees of a depth near K. This means that the region of the search space that contains the individuals with the best fitness increases exponentially with K. This can be considered as a first hint of the fact that the problems get more difficult as K increases. It is also interesting to remark that the average size rises quickly from the initial value. This means that the penalization of individuals with depth different from K is effective in forcing the search process to sample trees of depth close to K.

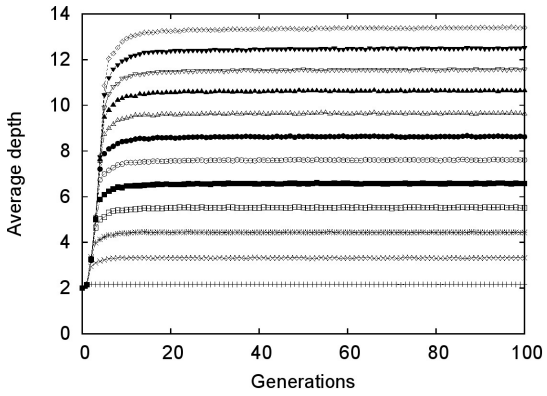


Figure 7: Median of the average depth of the GP trees in the population against generations.

Another indication of the different difficulty of the problem for different values of K is given by the plots of the median of the average number of nodes of the trees in the population, reported in Figure 8. The average number of nodes increases generation by generation for all the considered values of K. The main difference between the various cases is that, for small values of K, the increase in size stops after some generations, while for high values of K the process continues up to the last studied generation. This curves show that even if the search process is quickly directed towards the "right tree depth", obtaining the optimum is still a process that increases in difficulty with K. In other words, while reaching the depth of the optimal tree is easy, producing an optimal tree of a given depth remains a difficult process for GP.

Given this information on the average size and the depth of the trees, it is interesting to explore the same information on the best element of the population. In Figure 9 the median of the depth of the best individual is presented. As it is possible to see, for each studied value of K, the depth quickly rises to the corresponding value of K and remains constant for all the considered generations. This is consistent with the previous observations and shows that the best individual in the population after few generations has a depth that is close to the depth of at least one global optimum.

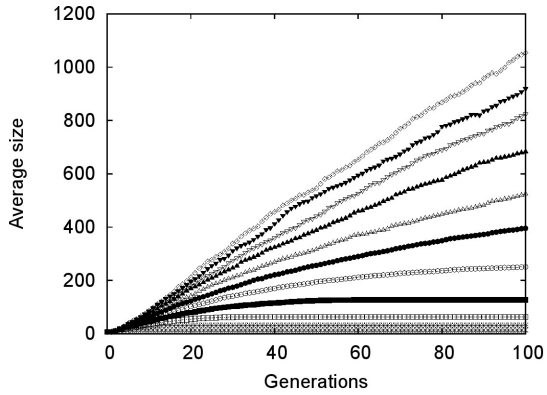


Figure 8: Median of the average number of nodes of the GP trees in the population against generations.

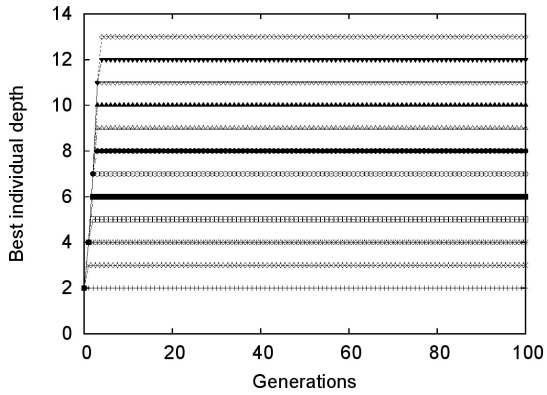


Figure 9: Median of the depth of the best GP tree at each generation.

Figure 10 reports the median of the number of nodes of the best individual at each generation. These results are consistent with the ones of the average number of nodes in the population reported so far. This means that the depth of the best individual, as the depth of many other individuals in the population, quickly rises to K, and then GP concentrates on finding the optimal number of nodes, focusing on the space of trees of depth close to K. Since at least one of the optima is a tree of a depth close to K that respects the property of Proposition 3, the number of nodes of that optimum increases exponentially with K, making the search process more difficult for high K values.

The fact that the difficulty of the problem increases with K is confirmed by Figure 11, where the median of the best fitness in the population at each generation is reported. Fitness values have been normalized by dividing them by the optimal fitness (that has been calculated using the algorithm reported at the end of Section 4). In this way, the global optimum has a normalized fitness equal to 1. The figure clearly shows that the difficulty increases with K, and this is visible since the very first generations. Furthermore, we point out that for $K = 2$, $K = 3$, $K = 4$ and $K = 5$ a global optimum has been reached by GP before generation 100 in the majority of the studied runs, while this is not the case for higher values of K. In particular, for values of K higher than 10 the displacement from a random individual (that has expected fitness equal to 0 because of the uniform random choice of the v function) is very low. This indicates that for values of K larger than 10, GP is behaving in a way that is comparable to random search, and the optimization process is extremely slow.

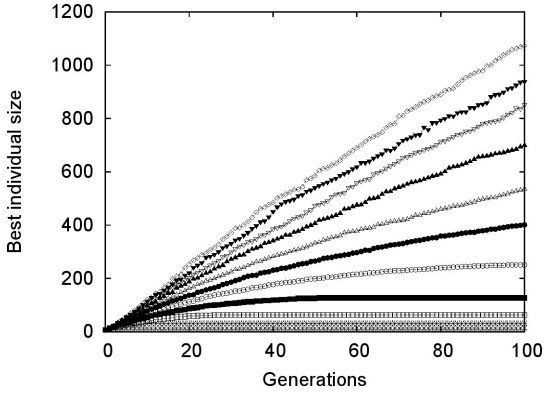


Figure 10: Median of the number of nodes of the best GP tree at each generation.

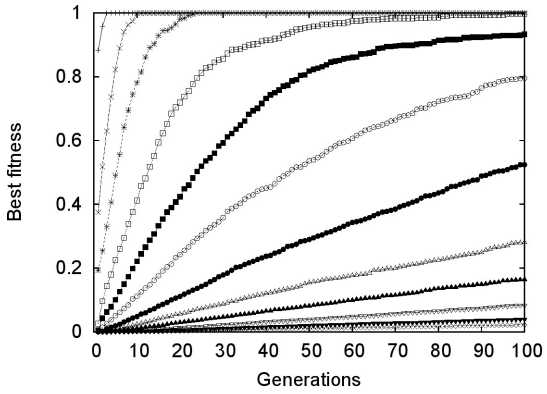


Figure 11: Median of the fitness of the best GP tree in the population at each generation. All fitness values have been normalized dividing them by the optimal fitness.

A more in-depth analysis has been done for the last studied generation of the runs. In the box plot in Figure 12 the average depth of the GP trees at the last generation is considered. It is interesting

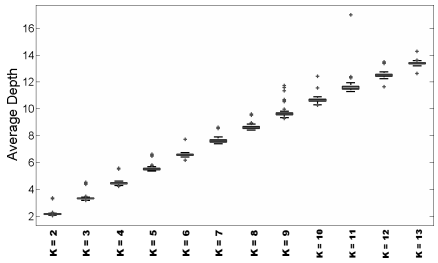


Figure 12: Average depth of the trees in the population at the last generation.

to note that the variabilities of the measured depths are low. This means that the search almost always remains near the desired value.

The box plot of the average number of nodes at the last generation is presented in Figure 13. It shows that the variability on the number of nodes increases with K. This behavior can be explained by the fact that the number of trees of a given depth increases with K.

The box plot of the best fitness at the last generation is presented in Figure 14. This figure shows that for different values of K there are different behaviors: for low values of K the fitness remains near 1. For intermediate values of K the fitness decreases steadily

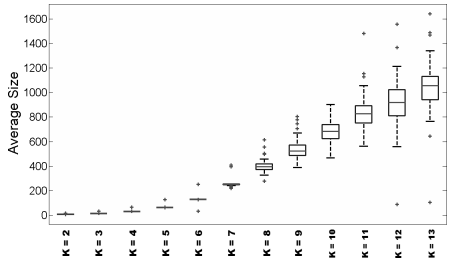


Figure 13: Average number of nodes of the trees in the population at the last generation.

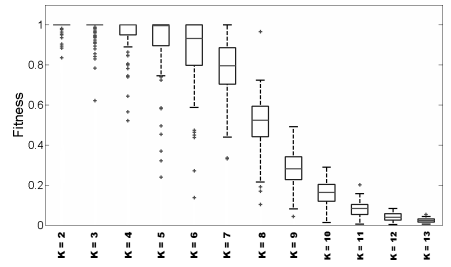


Figure 14: Fitness of the best individual in the population at the last generation.

when K increases. Finally, for high values of K the fitness still decreases but at a slower rate, since the fitness values approach 0, the expected fitness of a randomly generated GP tree.

We have also performed a t-test over the best fitnesses registered at the last studied generation for the performed 100 runs. The null hypothesis is that two means are identical. In Table 1 the p-values of the t-test are reported for all the different possible combinations of K. The combinations where the p-value is larger than 0.05 are written in italic. The results show that the differences between the recorded fitness values is almost always statistically significant for different values of K.

In conclusion, the presented experiments show that GP is able to find a global optimum in few generations for low values of K; increasing K, the best normalized fitness found by GP gets worse until, for large values of K, the behavior of GP becomes comparable to the one of random search. Interestingly, the size of the individuals inside the population increases with K. This allows us to conclude that, for large values of K, we have a progressive code growth without a corresponding improvement in fitness. This is exactly the definition of bloat, as presented, for instance, in [13]. This means that the difficulty of the proposed problem can be effectively tuned by changing the value of K: increasing K we create problems in which GP is more and more unable to optimize and in which GP is more and more affected by bloat.

6. CONCLUSIONS AND FUTURE WORK

An extension of the NK landscapes to tree based GP, simply called K landscapes, has been presented in this paper. In this benchmark, the epistatic interaction is quantified by the mutual influence on fitness of larger and larger structures in a tree as the value of the K parameter increases. The fact that the hardness of the problem increases with K has been experimentally shown. Furthermore, we have shown that GP produces more bloat as K increases.

A formal proof of the fact that the number of local optima increases with K is needed in the future and it is one of the main subjects of our current research. Furthermore, we plan to investigate the use of many landscape indicators, like fitness distance

k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10	k=11	k=12	k=13	
.0193	.0001	.0000	*.1713*	.0000	.0000	.0000	.0000	.0000	.0000	.0000	**k=2**
	.0301	.0002	*.2728*	.0000	.0000	.0000	.0000	.0000	.0000	.0000	**k=3**
		.0365	*.5148*	.0000	.0000	.0000	.0000	.0000	.0000	.0000	**k=4**
			.9650	.0000	.0000	.0000	.0000	.0000	.0000	.0000	**k=5**
				.0115	.0000	.0000	.0000	.0000	.0000	.0000	**k=6**
					.0000	.0000	.0000	.0000	.0000	.0000	**k=7**
						.0000	.0000	.0000	.0000	.0000	**k=8**
							.0000	.0000	.0009	.0001	**k=9**
								.0000	*.2293*	*.0845*	**k=10**
									.7669	*.9045*	**k=11**
										.7645	**k=12**

Table 1: The p-values given by the t-test.

correlation, auto-correlation, density of states and many others, on the proposed benchmark, in order to further corroborate the hypothesis that a more and more rugged fitness landscape is induced by increasing K. Finally, we plan to extend the proposed benchmark, overcoming some of its major limitations, namely: (1) the current model does exhibit epistatic behavior under point mutation, but this is no longer true when considering subtree crossover; (2) the current benchmark cannot model dead code: every subtree contributes to the fitness; (3) an ideal solution is very repetitive in terms of used subtrees and this is very far from real world, where it is rarely the case that an ideal solution can be built from cloned and systematically arranged subtrees.

7. REFERENCES

[1] L. Altenberg. B2.7.2 NK fitness landscapes. In T. Baeck, et al., editors, *Handbook of evolutionary computation*. New York: Oxford University Press, 1997.

[2] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[3] J. M. Daida, R. Bertram, S. Stanhope, J. Khoo, S. Chaudhary, and O. Chaudhary. What makes a problem GP-hard? analysis of a tunably difficult problem in genetic programming. *Genetic Programming and Evolvable Machines*, 2:165–191, 2001.

[4] 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.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[6] S. Gustafson, E. Burke, and N. Krasnogor. The tree-string problem: An artificial domain for structure and content search. In M. Keijzer, et al., editors, *EuroGP 2005*, pages 215–226. Springer, 2005.

[7] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan, 1975.

[8] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *J. Theoret. Biol.*, 128(1):11–45, 1987.

[9] S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.

[10] J. Kingman. *Mathematics of genetic diversity*. Number 34 in CBMS-NSF regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 2. druck edition, 1980.

[11] J. R. Koza. *Genetic Programming*. The MIT Press, Cambridge, Massachusetts, 1992.

[12] M. Mitchell, S. Forrest, and J. Holland. The royal road for genetic algorithms: fitness landscapes and ga performance. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems, Proc. of the First European Conf. on Artif. Life*, pages 245–254. The MIT Press, 1992.

[13] M. O'Neill, L. Vanneschi, S. Gustafson, and W. Banzhaf. Open issues in genetic programming. *Genetic Programming and Evolvable Machines*, 11(3-4):339–363, 2010.

[14] A. Orfila, J. M. Estevez-Tapiador, and A. Ribagorda. Evolving high-speed, easy-to-understand network intrusion detection rules with genetic programming. In M. Giacobini, et al., editors, *App. of Evolutionary Computing, EvoWorkshops2009*, LNCS. Springer Verlag, 2009.

[15] 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).

[16] 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.

[17] D. Song, M. I. Heywood, and A. N. Zincir-Heywood. A linear genetic programming approach to intrusion detection. In E. Cantú-Paz, et al., editors, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 2325–2336, Chicago, 12-16 July 2003. Springer-Verlag.

[18] W. A. Tackett. *Recombination, Selection, and the Genetic Construction of Computer Programs*. PhD thesis, University of Southern California, Department of Electrical Engineering Systems, USA, 1994.

[19] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue. A study of fitness distance correlation as a difficulty measure in genetic programming. *Evolutionary Computation*, 13(2):213–239, Summer 2005.

[20] L. Vanneschi. *Theory and Practice for Efficient Genetic Programming*. Ph.D. thesis, Faculty of Science, University of Lausanne, Switzerland, 2004. Downlodable version at: http://www.disco.unimib.it/vanneschi.

[21] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In D. F. Jones, editor, *Proceedings og the Sixth International Congress on Genetics*, volume 1, pages 356–366, 1932.