INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Experiments on controlled regularity fitness landscapes

Evelyne LUTTON — Jacques LEVY VEHEL — Yann LANDRIN-SCHWEITZER

N° 5823

Février 2006

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5823--FR+ENG

INRIA
ROCQUENCOURT

Experiments on controlled regularity fitness landscapes

Evelyne LUTTON , Jacques LEVY VEHEL , Yann LANDRIN-SCHWEITZER

Thème COG — Systèmes cognitifs
Projet Complex

Rapport de recherche n° 5823 — Février 2006 —17 pages

Abstract: We present an experimental analysis of the influence of the local irregularity of the fitness function on the behavior of a simple version of an evolutionary algorithm (EA). Previous theoretical as well as experimental work on this subject suggest that the performance of EA strongly depends on the irregularity of the fitness function. Several irregularity measures have been derived, in order to numerically characterize this type of difficulty source for EA. These characterizations are mainly based on Hölder exponents. Previous studies used a global characterization of fitness regularity (namely the global Hölder exponent), with experimental validations being conducted on test functions with uniform irregularity. The present work refines the analysis by investigating the behavior of an EA on functions displaying variable local regularity. Our experiments confirm and quantify the intuition that performance decreases as irregularity increases. In addition, they suggest a way to modify the genetic topology to accommodate for variable regularity: More precisely, it appears that the mutation parameter, which controls the size of the neighbourhood of a point, should increase when regularity decreases. These results open the way to a theoretical analysis based on local Hölder exponents, and poses several questions with respect to on-line measurements and usage of regularity for fitness functions.

Key-words: Evolutionary algorithms, Hölder exponents, controlled regularity function.

Analyse expérimentale des paysages de fitness à régularité contrôlée

Résumé : Nous présentons une analyse expérimentale de l'influence de la régularité locale de la fonction de fitness sur le comportement d'un algorithme évolutionnaire (AE) simple. Des études théoriques et expérimentales antérieures suggèrent que les performances d'un AE dépendent fortement de l'irrégularité de la fonction de fitness. Plusieurs mesures de régularité ont été proposées dans le but de caractériser numériquement cette source de difficulté. Ces caractérisations sont fondées sur des exposants de Hölder. Les études antérieures considéraient une caractérisation globale de la régularité (en considérant un exposant de Hölder global), avec une validation expérimentale sur des fonctions à régularité uniforme. Les travaux présentés ici en sont un raffinement, par l'analyse du comportement d'un AE sur des fonctions ayant une régularité locale variable. Nos expérimentations confirment et quantifient l'intuition que les performance d'un AE chutent lorsque l'irrégularité des fonctions croît. En outre, les résultats obtenus suggèrent une métode pour modifier la topologie génétique afin de s'accomoder localement à l'irrégularité. Plus précisément, il apparaît que le paramètre de mutation qui contrôle la taille du voisinage d'un point devrait croître lorsque la régularité décroît. Ces résultats justifient l'intérêt d'une analyse théorique fondée sur les exposants de Hölder locaux, et posent plusieurs questions concernant l'estimation en ligne et l'usage des mesures de régularité.

Mots-clés : Algorithmes évolutionnaires, exposant de Hölder, fonctions à régularité prescrite.

1 Introduction

Irregularity has been experimentally and theoretically identified as a "difficulty" factor for EA [8, 5, 9]. More precisely a relationship between a measure of the fitness regularity (the global Hölder exponent of the fitness function) and a deception measure has been established. Experimental analyses, conducted on Weierstrass functions, have confirmed the theoretical findings. Weierstrass functions have a controlled regularity, which is uniform over its domain. This is a limitation in practice: "Real world" fitness function that one encounters in usual EA applications have variable regularity. It seems intuitive that the global results obtained previously should apply locally: More precisely, one expects that an EA should more easily locate a maximum lying in a smooth region than a maximum lying in an irregular neighbourhood. The aim of the present work is to confirm and quantify this intuition. In that purpose, we build below functions with controlled regularity and let simple versions of EA try to optimize them.

Of course local irregularity is but one aspect of difficulty for EA. Another identified source of "deception" for EA is epistasy [11]. The relationship between irregularity and epistasy has not yet been fully investigated. It seems however probable that these two sources are of different nature ("epistasy is not enough" [10]). Another factor is temporal noise [1, 2]. We do not consider in this paper temporal variations. Temporal irregularity is another complex and interesting aspect that remains to be investigated. All functions considered in this paper are fixed, and remain the same during the EA evolution. The question of interest here is to experiment the behaviour of an EA on a controlled but variable regularity test function, regularity variation being considered with respect to the spacial parameters.

Our work may be seen as a new contribution to the study of controlled fitness landscapes, which has been largely developed in the EA community (NK-landscapes and tuneable fitness landscapes [10], $(1, \lambda)$-ES on simple function [3]), in order to understand the behaviour of some simple EA engines in a controlled environment. It must be clear that fitness landscapes involves two main aspects: the fitness function itself, and the genetic engine characteristics, that set a specific topology on the definition domain. It is well known that for a same fitness function, two different EA engines (for example with or without crossover) may have very different behaviour. The term fitness landscape involves both the profile of the fitness function on its definition domain and the search paths produced by the genetic operators. Quantities related to EA difficulty must be measured with respect to this "genetic" topology.

For regularity measurements the same holds: Irregularity characteristics must be measured with respect to an underlying measure based on the genetic operators effect. In other terms, the neighborhood system that serves as a basis for the calculation of Hölder exponents must ideally be linked with transition probabilities via the genetic operators. We should thus talk about fitness landscape irregularity, instead of fitness function irregularity. A first attempt has been done in this direction in [5] for discrete fitness landscapes. The present work deals with continuous functions.

The paper is organized as follows: section 2 recalls the basic definitions of Hölder global and local exponents. Section 3 presents the proposed test-functions. Section 4 presents the experimental analysis of several simple EA. Conclusions and future work are detailed in section 5.

2 Global and local regularity

Hölder regularity analysis is an important topic in various fields such as partial differential equations, fractal geometry and signal/image processing [7].

Hölder regularity allows to quantify in a precise way both local and global regularity. For our purposes, the following notions will be relevant. To simplify notations, we assume that our signals are nowhere differentiable. Generalization to other signals simply requires to introduce derivatives in the definitions.

Let $\alpha \in (0, 1)$, and $\Omega \subset \mathbf{R}$. One says that a function f defined on Ω belongs to $C_l^\alpha(\Omega)$ if:

$$\exists\, C : \forall x, y \in \Omega : \frac{|f(x) - f(y)|}{|x - y|^\alpha} \leq C$$

The supremum of the values α such that f belongs to $C_l^\alpha(\Omega)$ is called the global Hölder exponent of f in Ω. From the definition, it is clear that smaller values of α correspond to more irregular functions.

A local characterisation is derived by considering the same quantities in a restricted area of radius ρ centered on x_0, $B\,(x_0, \rho)$. Let now:

$$\alpha\,(f, x_0, \rho) = \sup\left\{\alpha : f \in C_l^\alpha\,(B\,(x_0, \rho))\right\}.$$

Clearly, $\alpha\,(f, x_0, \rho)$ is non increasing as a function of ρ. The *local Hölder exponent* of f at x_0 is the real number $\alpha\,(x_0) = \lim_{\rho \to 0} \alpha\,(f, x_0, \rho)$

Since $\alpha(x)$ is defined at each point, we may associate to f the function $x \to \alpha(x)$ which measures the evolution of its regularity.

This regularity characterization is widely used in fractal analysis because it has direct interpretations both mathematically and in applications. It has been shown for instance that α indeed corresponds to the auditive perception of smoothness for voice signals. Similarly, simply computing the Hölder exponent at each point of an image already gives a good idea of its structure, as for instance its edges [6]. More generally, in many applications, it is desirable to model, synthesize or process signals which are highly irregular, and for which the relevant information lies in the singularities more than in the amplitude. In such cases, the study of the Hölder functions is of obvious interest.

3 Test function with controlled local regularity

3.1 Weierstrass function

In order to precisely and finely investigate the impact of local regularity on the behavior of an EA, we need to construct test functions with prescribed Hölder exponent. In addition, we must make sure that no other factor will come into play and thus interfere with the analysis. An easy way to control $\alpha(x)$ is to consider generalized Weierstrass functions. Let us first recall basic facts on the usual Weierstrass function.

The Weierstrass function reads

$$W_{b,h}(x) = \sum_{i=1}^{\infty} b^{-ih} sin(b^i x) \quad \text{with } b \geq 2 \text{ and } 0 < h < 1$$

The parameter h controls the regularity: The global Hölder exponent of $W_{b,h}$ on, e.g., $[0, 1]$, is equal to h. In addition, $\alpha(x) = h$ for all x ([4]). Weierstrass functions are very irregular for small values of h, and become smoother as h tends to 1.

Generalized Weierstrass functions are defined as follows:

$$GW_{b,h}(x) = \sum_{i=1}^{\infty} b^{-ih(x)} sin(b^i x) \quad \text{with } b \geq 2 \text{ and } 0 < h(x) < 1$$

Provided h is differentiable, the local Hölder exponent of $GW_{b,h}$ is $h(x)$ at each x .

Figure 1 displays a generalized Weierstrass function with $h(x) = x$ on $(0, 1)$. One can clearly see the local regularity increasing along the graph. However, an additional feature is present: The local oscillation is large around 0, and decreases as x increases. It is important to note that the variation of the local oscillation is independent from the evolution of $\alpha(x)$. This particular behavior of $GW_{b,h}$ is a nuisance in our case: Since we want to focus on the sensitivity of the EA to local regularity, we need to get rid of other sources of variations, that would perturb our study. We will thus deal with a modified version of $GW_{b,h}$ where the local oscillations are normalized. This is explained in details below.

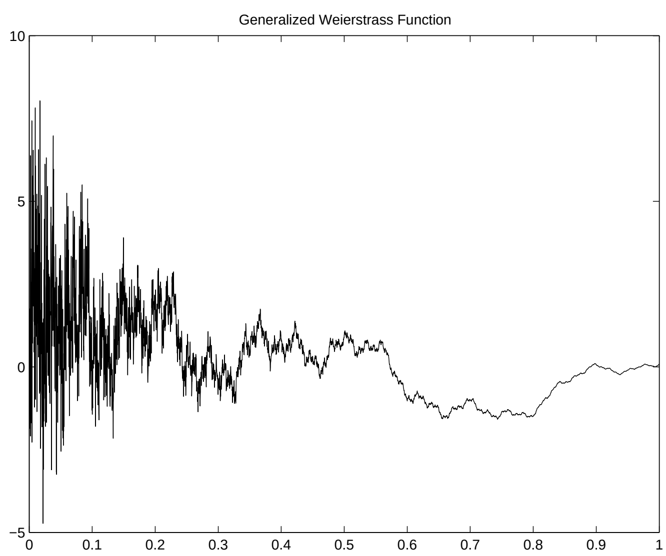

Figure 1: Generalized Weierstrass function with $h(x) = x$.

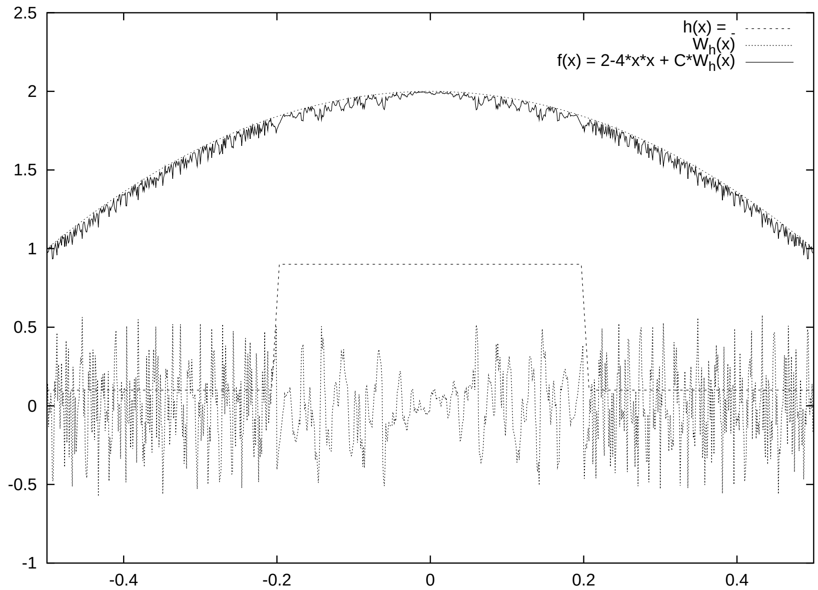

Figure 2: N(x): The “n” regularity profile function.

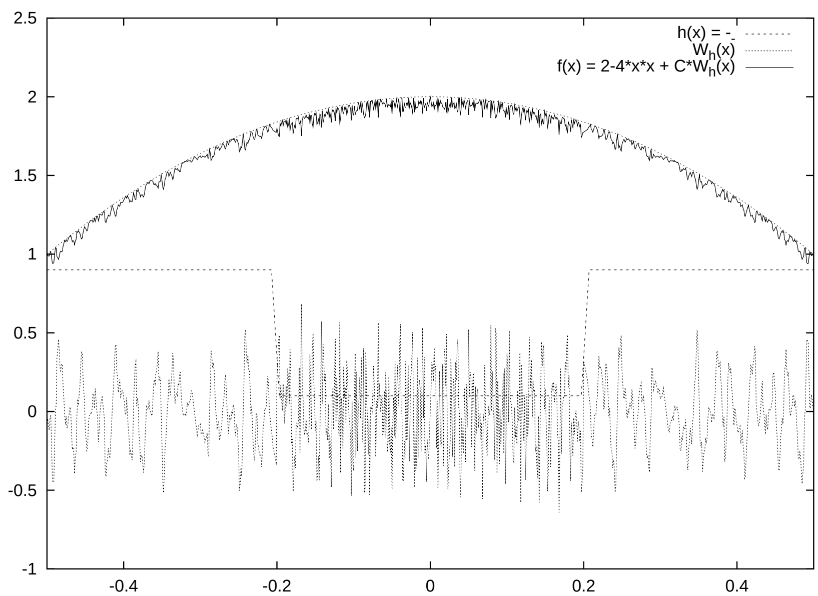

Figure 3: U(x): The “u” profile regularity function.

3.2 Test functions

We build two test functions with identical features except for the local regularity profile. For evident reasons, we have constrained the functions to have the same maximum fitness value located at the same point (0, the center of the domain), and a similar underlying smooth (quadratic) component. The irregularity is considered as a "noisy" local perturbation of limited amplitude.

The generalized Weierstrass function is oscillation-normalized as follows. The local mean value and maximal absolute increment are computed by sampling a neighborhood of width ϵ for each point x of the search space $[-0.5, 0.5]$:

$$\mu_\epsilon(x) = \frac{1}{2K}\sum_{k=-K}^{K} GW_{b,h}(x + k\frac{\epsilon}{K}) \quad (1)$$

$$Var_\epsilon(x) = Max_{k\in[-K,K]}|GW_{b,h}(x + k\frac{\epsilon}{K}) - \mu_\epsilon(x)| \quad (2)$$

The normalised generalized Weiertrass function is then (dotted curves on figures 2 and 3)

$$NW_{b,h}(x) = \frac{GW_{b,h}(x) - \mu_\epsilon(x)}{Var_\epsilon(x)}$$

The fitness function is finally defined as the smooth trend plus the noisy component with controlled irregularity. It has the following form:

$$f(x) = 2 - 4x^2 - |NW_{b,h}(x)|$$

The noisy component is included as a local perturbation (of small amplitude) that is *subtracted* to the smooth trend, in order to be sure to get se same global maximum at $x = 0$, with the same fitness target value (2)[1]. $NW_{b,h}(x)$ always equal 0 at $x = 0$, whatever h.

In the experiments, we consider two profiles (the C source code is given in appendix):

1. **Favourable case : irregular areas of the function have a low fitness** (Figure2)

$$h(x) = 0.9 \quad \text{if} \quad x \in [-0.2, 0.2]$$
$$h(x) = 0.1 \quad \text{else}$$

2. **Unfavourable case : the most irregular points are located around the global maximum** (Figure3)

$$h(x) = 0.1 \quad \text{if} \quad x \in [-0.2, 0.2]$$
$$h(x) = 0.9 \quad \text{else}$$

Note that both h functions are not differentiable at ± 0.2. At all other points in $[-0.5, 0.5]$, however, h is smooth, and the local Hölder exponent of our fitness function is indeed equal to $h(x)$.

[1] Additionnally, each local maximum is located on the smooth trend $2 - 4x^2$.

4 Experimental analysis

In order to gain some knowledge on the dynamic of an EA on such landscapes, several experiments have been done with various EA engines. Statistics have been done on 20 runs for each function. In every experiment reported below, initialisations have been done in order to see how quickly the EA is able to find the global optimum (the top of the hill on $x = 0$): initial populations are located on the limits of the search space by a random shot of the two values $x = \pm 0.5$.

Convergence capabilities have been measured in two different ways:

- at a fixed number of generations, by the current best fitness value f^* (ideally 2) or equivalently the absolute value of the abscissa $|x^*|$
- by the number of generations T^* in order to get $|x^*| \leq 0.1$.

These characteristics have been tested with respect to the mutation parameter σ that tunes the "genetic" topology, i.e. the size of the neighborhood of a point by mutation. For this uniform mutation of fixed radius, the "genetic" topology is simple and corresponds directly to the euclidean distance that serves as a basis for the estimation of the local Hölder exponent. However, the σ parameter tunes a scale transformation, that represents the capability of the EA to explore the neighborhood of a given point.

In all figures below, the continuous curves corresponds to the "N" irregularity profile, i.e. the favourable one, while the dotted curves are for the "U" profile, i.e. the unfavourable case.

A first set of tests has been performed on the basic (1+1)-ES with uniform mutation of radius σ (figures 4 and 5). The behaviour of the (1+1)-ES on U and N profiles is different and exhibits a shorter convergence time (figure 4) and a better fitness in a given number of generations (figure 5 for 300 generations) for the "favourable" case.

The behaviour with respect of σ shows that a larger σ helps the algorithm to find his way toward the optimum. The "favourable" (continuous curve) case obviously needs a smaller σ to obtain a similar result in comparison to the "unfavourable case" (dotted curves).

A second set of test has been performed with population-based ES, namely a (10+10)-ES and and (50+50)-ES with uniform mutation of radius σ, see figures 6 to 9. The selection operator used is a tournament of size 2.

The difference of behaviour between U and N is less clear. Obviously the influence of local regularity vanishes with the "parallel" processing of the population that samples various areas of the search space (the larger the population, the most similar the dotted and continuous curves). This effect is especially evident with a low selection pressure algorithm (tournament of size 2). But the behaviour with respect to σ remains.

A third set of experiments has been performed in order to test the influence of a crossover operator. A (10+10) ES with uniform mutation and barycentric crossover ($pc = 0.5$, $pm = 1$) has thus been used, see figures 10 and 11. The convergence times are much shorter, but the difference of behaviour between the two functions is again in favour of the "favourable" irregularity profile, while the influence of the σ mutation parameter is much lower. These results suggest that on such landscapes the crossover operator is "dominant" (and very efficient), but sensitive to local irregularity.

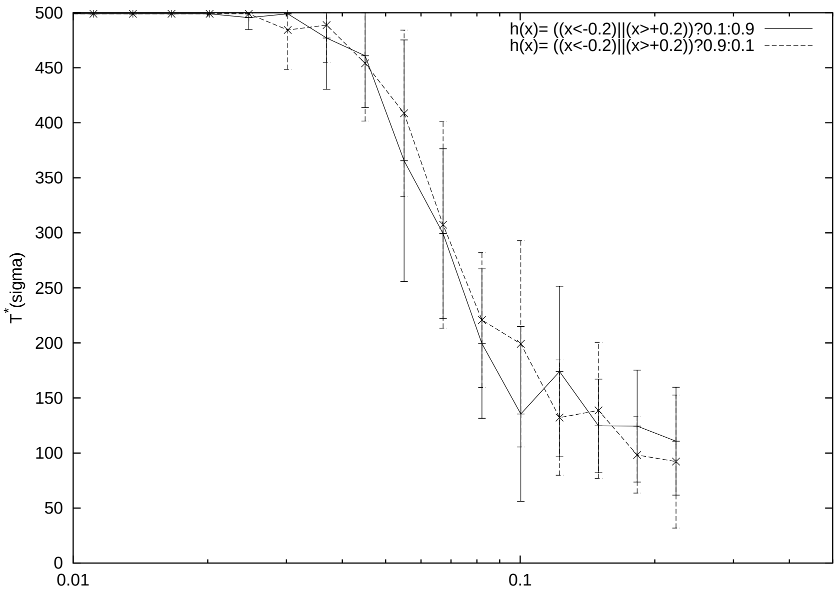

Figure 4: Average convergence time $T*$ (nb of generations) for a (1+1)-ES for the U and N functions as a function of σ (logarithmic scale). Vertical bars figure the standard deviations, statistics have been made on 20 runs for each parameter setting.

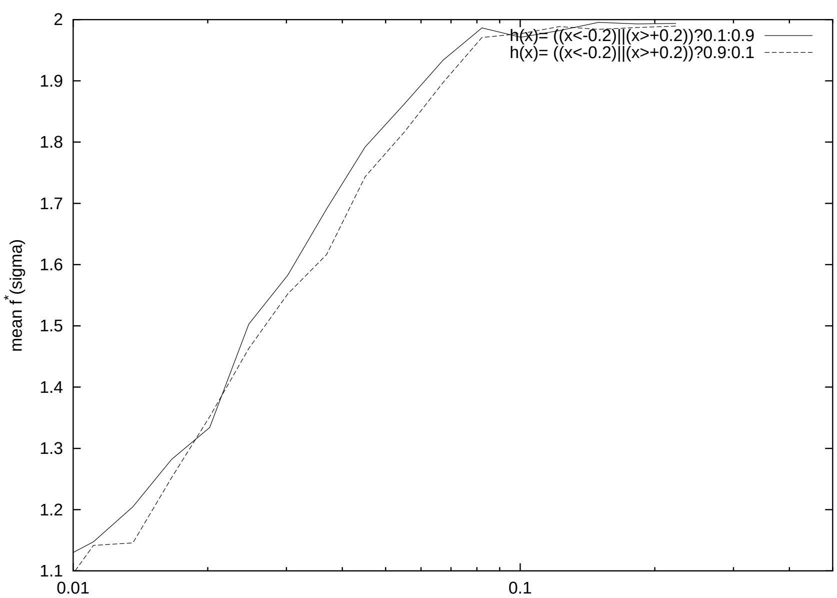

Figure 5: Average best fitness in 300 generations of a (1+1)-ES for the U and N functions as a function of σ

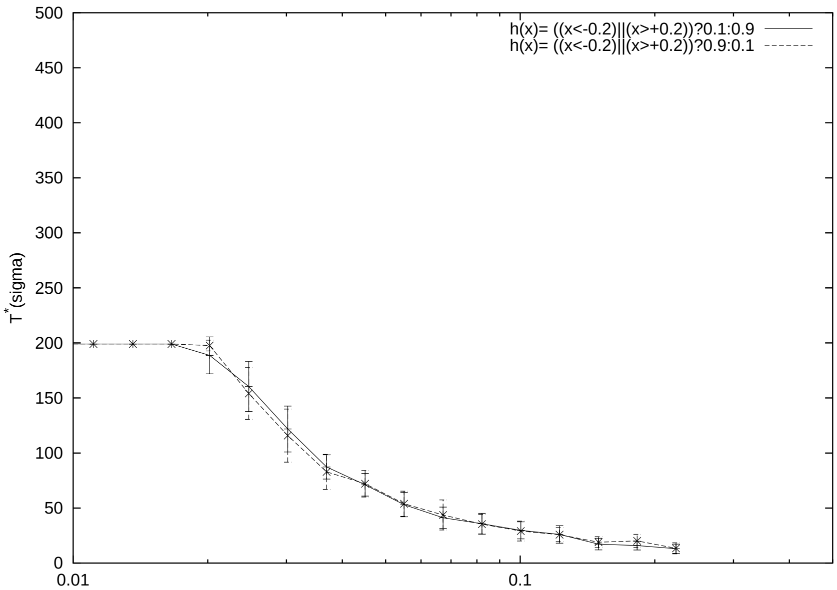

Figure 6: Average convergence time $T*$ (nb of generations) for a (10+10)-ES for the U and N functions as a function of σ

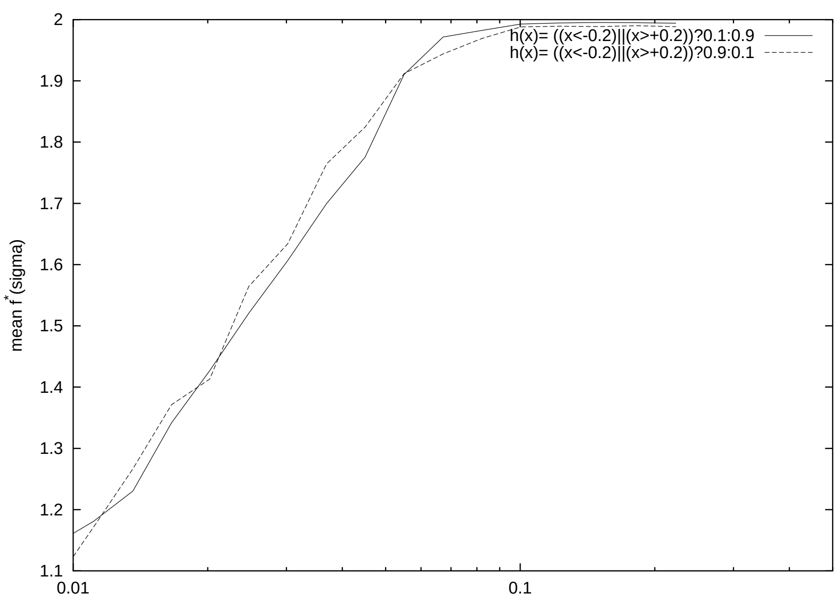

Figure 7: Average best fitness in 50 generations of a (10+10)-ES for the U and N functions as a function of σ

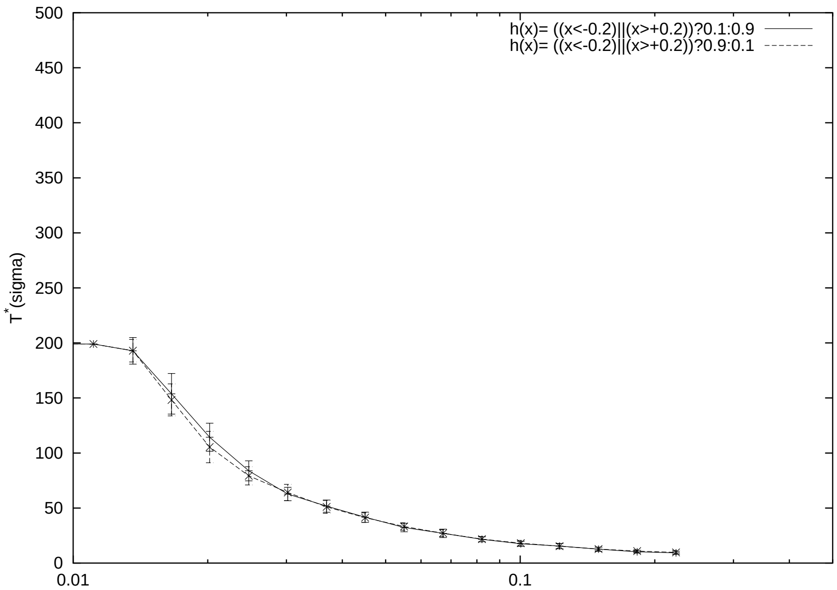

Figure 8: Average convergence time $T*$ (nb of generations) for a (50+50)-ES for the U and N functions as a function of σ

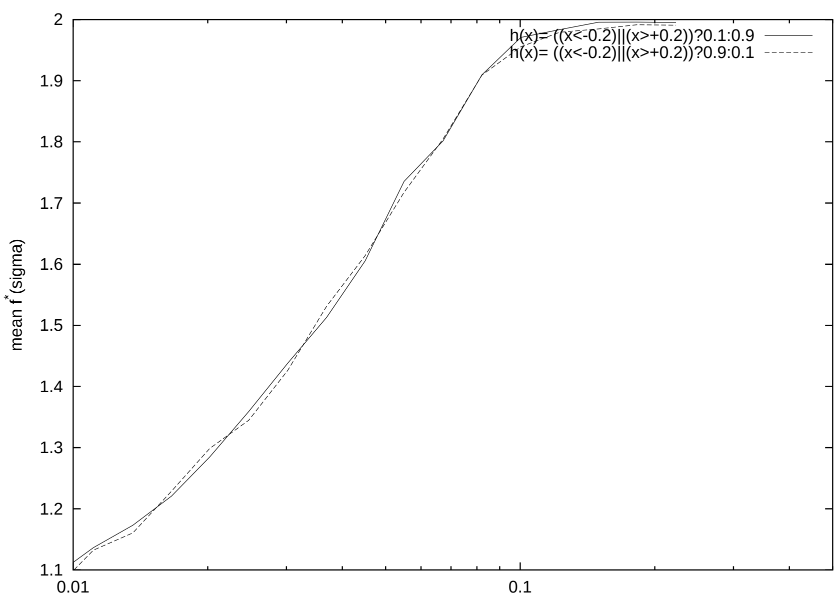

Figure 9: Average best fitness in 20 generations of a (50+50)-ES for the U and N functions as a function of σ

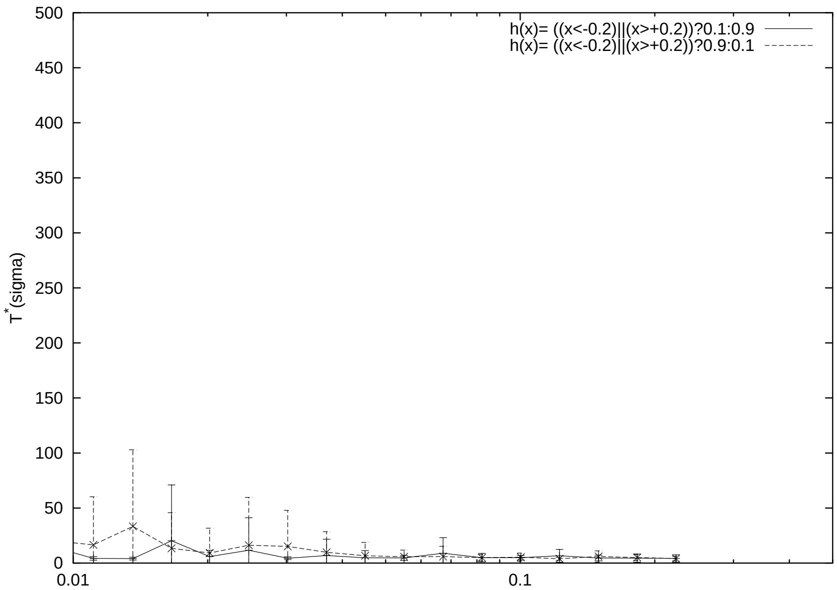

Figure 10: Average convergence time (nb of generations) $T*$ for a (10+10)-ES with barycentric crossover for the U and N functions as a function of σ

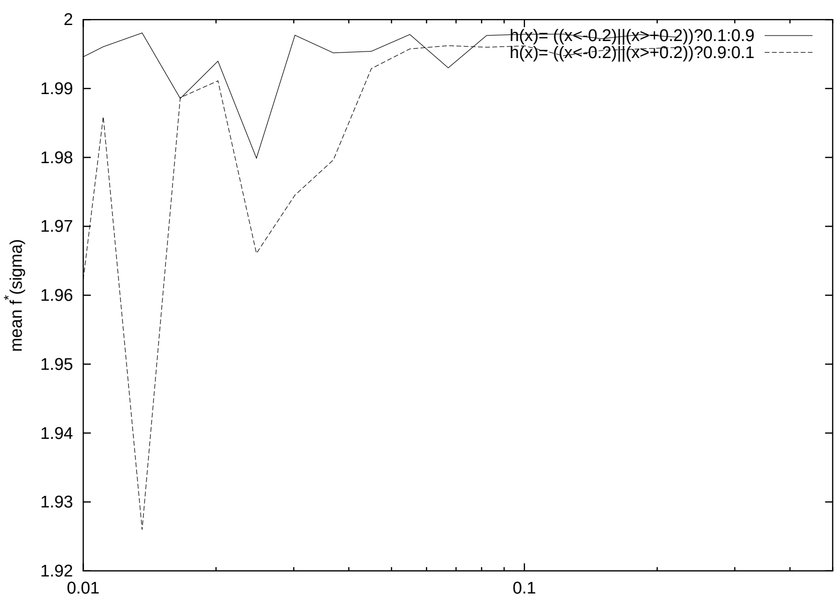

Figure 11: Average best fitness in 50 generations of a (10+10)-ES with barycentric crossover for the U and N functions as a function of σ

However for further studies of crossover operators, a more precise definition of regularity is needed, with respect to the ”genetic” topology of this operator.

5 Conclusion and future work

Hölder regularity is difficult to measure, as it requires the knowledge of the whole fitness function over its definition domain (or at least a sampling at various scales). As they only consider local quantities, local Hölder exponents seem more adapted to the modelling of EA evolution in an unknown landscape.

Moreover, the dynamics of EAs can be modeled as an irregular sampling of its search space, biaised towards high fitness areas.

These two reasons advocate for on-line estimations of the local irregularity measures. And as this must ideally be done on the basis of the topology provided by genetic operators, an offspring tracing (genealogic analysis !) procedure may be considered. If done so, a convenient (and computationally economic) way of gaining irregularity informations could be designed. The question of efficiently using these informations still remains, for the moment.

The presented experimental analysis has proved that a mutation parameter σ conversely proportional to the local Hölder exponent $\alpha(x)$ has to be considered as an adaptive mutation scheme. However, further theoretical analyses must be done to better understand the influence of the parameter tuning on fitness landscapes irregularity, and to be able to derive efficient adaptive tuning stategies.

References

[1] D. V. Arnold and H.-G. Beyer. Efficiency and mutation strength adaptation of the (μ, μ_i, λ)-es in a noisy environment. In M. Schoenauer, K. Deb, G. Rudolf, X. Yao, E. Lutton, Merelo. J.J., and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, Paris, France, September 16-20 2000. Springer Verlag. LNCS 1917.

[2] H.-G. Beyer. Evolutionary Algorithms in Noisy Environments: Theoretical Issues and Guidelines for Practice. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4):239–267, 2000.

[3] H.-G. Beyer. On the Performance of $(1, \lambda)$-Evolution Strategies for the Ridge Function Class. *IEEE Transactions on Evolutionary Computation*, 5(3):218–235, 2001.

[4] Kenneth Falconer. *Fractal geometry*. John Wiley & Sons Ltd., Chichester, 1990. Mathematical foundations and applications.

[5] Benoit Leblanc and Evelyne Lutton. Bitwise regularity and ga-hardness. In *ICEC 98, May 5-9, Anchorage, Alaska*, 1998.

[6] J. Lévy Véhel. Fractal approaches in signal processing. In H.O. Peitgen C.J.G. Evertsz and R.F. Voss, editors, *Fractal Geometry and Analysis*. World Scientific, 1996.

[7] Jacques Lévy Véhel and Evelyne Lutton. Evolutionary signal enhancement based on hölder regularity analysis. In *EVOIASP2001 Workshop, Como Lake, Italy, Springer Verlag, LNCS 2038*, 2001.

[8] E. Lutton and J. Lévy Véhel. Hölder functions and deception of genetic algorithms. *IEEE transactions on Evolutionary computation*, 2(2):56–72, July 1998.

[9] Evelyne Lutton. *Evolutionary Algorithms in Engineering and Computer Science*, chapter Genetic Algorithms and Fractals. John Wiley and Sons, 1999.

[10] C. R. Reeves. Experiments with tuneable fitness landscapes. In M. Schoenauer, K. Deb, G. Rudolf, X. Yao, E. Lutton, Merelo. J.J., and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, Paris, France, September 16-20 2000. Springer Verlag. LNCS 1917.

[11] S. Rochet, G.Venturini, M. Slimane, and E.M. El Kharoubi. *Artificial Evolution, European Conference, AE 97, Nimes, France, October 1997, Selected papers*, volume Lecture Notes in Computer Science, chapter A critical and empirical study of epistasis measures for predicting GA performances : a summary. Springer Verlag, 1997.

Appendix: C Source code for Generalized Weierstrass, U and N functions

```
/*______________________________________________________

Tests on functions with prescribed local regularity

Generalized Weierstrass functions

______________________________________________________*/

#include <stdlib.h>
#include <math.h>

#define X_MIN -0.5
#define X_MAX 0.5
#define Abs(x) ((x) < 0 ? -(x) : (x))
#define MAX(x,y) ((x)>(y)?(x):(y))
#define MIN(x,y) ((x)<(y)?(x):(y))


// Generalized Weierstrass where the Holder exponent h(x) = Abs(x)
double WeierX(double);

// "Unfavourable" Holder profile
double ProfilU(double);
// Generalized Weierstrass where the Holder exponent h(x) = U(x)
double WeierU(double);
```

```
// Oscillation-normalised generalized Weierstrass where h(x) = U(x)
double FonctionU(double);

// "Favourable" Holder profile
double ProfilN(double);
// Generalized Weierstrass where the Holder exponent h(x) = N(x)
double WeierN(double);
// Oscillation-normalised generalized Weierstrass where h(x) = N(x)
double FonctionN(double);


// The number of steps we actually compute in infinite sums
#define ITER 50




//------------------------------------------------------------
// Generalized Weierstrass where the Holder exponent h(x) = Abs(x)
// W(x) is defined for x in [-0.5, 0.5]

double WeierX(double x) {
  double val = 0;
  double b = 2.;
  double h=0;
  int k;

// Compute the desired local Holder exponent
  h = Abs(x);

// Compute the infinite sum of 2^{-k.h(x)}.sin(2^k . x)
  for (k=0;k<ITER;k++)
    val += pow(b,-(double)k*h) * sin(pow(b,(double)k)*x);

  return val ;
}




//------------------------------------------------------------
// "Unfavourable" Holder profile
// U(x) is 0.1 close to 0, 0.9 elsewhere

double ProfilU(double x) {
  if (Abs(x)<0.2) return 0.1 ;
  else return 0.9 ;
}


//------------------------------------------------------------
// Generalized Weierstrass where the Holder exponent h(x) = U(x)
// Wu(x) is defined for x in [-0.5, 0.5]

double WeierU(double x) {
  double val= 0;
  double b = 2.;
  double h=0.;

  int k;

// Compute the desired local Holder exponent
  h =  ProfilU(x);
```

```
// Compute the infinite sum of 2^{-k.h(x)}.sin(2^k . x)
  for (k=0;k<ITER;k++)
    val += pow(b,-(double)k*h) * sin(pow(b,(double)k)*x);

  return val ;
}


//----------------------------------------------------------
// Oscillation-normalised generalized Weierstrass
// Fu(x) is defined for x in [-0.5, 0.5], and also has h(x) = U(x)

double FonctionU(double x) {
// Diameter of the neighborhood used for the normalisation
  double epsilon = 0.05;
// Number of samples in the neighborhood
  int Kmax = 10;

// Mean on the neighborhood
  double mean = 0.;
// Max variation on the neighborhood
  double varia = 0.;

  double res =0;
  int k;

// Computing the function's mean on the neighborhood
  for (k=-Kmax;k<Kmax;k++) mean += WeierU(x+(k*epsilon)/Kmax);
  mean = mean/(2.*Kmax);
// Computing the function's maximum variation on the neighborhood
  for (k=-Kmax;k<Kmax;k++) varia = MAX(varia,WeierU(x+ k*epsilon/Kmax) - mean);

// Normalising the Weierstrass function at x, and adding the global quadratic trend
  if (varia)  res = 2-4*x*x - 0.1*Abs((WeierU(x) - mean)/varia);

  return res ;
}



//----------------------------------------------------------
// "Favourable" Holder profile
// N(x) is 0.9 close to 0, and 0.1 elsewhere

double ProfilN(double x) {
  if (Abs(x)<0.2) return 0.9 ;
  else return 0.1 ;
}


//----------------------------------------------------------
// Generalized Weierstrass where the Holder exponent h(x) = N(x)
// Wn(x) is defined for x in [-0.5, 0.5]

double WeierN(double x) {
  double val= 0;
  double b = 2.;
  double h=0.;

  int k;
```

```
// Compute the desired local Holder exponent
  h =  ProfilN(x);

// Compute the infinite sum of 2^{-k.h(x)}.sin(2^k . x)
  for (k=0;k<ITER;k++)
    val += pow(b,-(double)k*h) * sin(pow(b,(double)k)*x);

  return val ;
}


//----------------------------------------------------------
// Oscillation-normalised generalized Weierstrass
// Fn(x) is defined for x in [-0.5, 0.5], and also has h(x) = N(x)

double FonctionN(double x) {
// Diameter of the neighborhood used for the normalisation
  double epsilon = 0.05;
// Number of samples in the neighborhood
  int Kmax = 10;

// Mean on the neighborhood
  double mean = 0.;
// Max variation on the neighborhood
  double varia = 0.;

  double res =0;
  int k;

// Computing the function's mean on the neighborhood
  for (k=-Kmax;k<Kmax;k++) mean += WeierN(x+(k*epsilon)/Kmax);
  mean = mean/(2.*Kmax);
// Computing the function's maximum variation on the neighborhood
  for (k=-Kmax;k<Kmax;k++) varia = MAX(varia,WeierN(x+ k*epsilon/Kmax) - mean);

// Normalising the Weierstrass function at x, and adding the global quadratic trend
  if (varia)  res = 2-4*x*x - 0.1*Abs((WeierN(x) - mean)/varia);

  return res ;
}
```

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399