A study of population uniformization in GAs

Philippe Preux*
Laboratoire d'Informatique du Littoral, Calais, France
and
LIFL, URA CNRS 369, Villeneuve d'Ascq, France

June 1994

Abstract

The work reported here aims at describing and understanding the behavior of GAs within a "reachable" future, rather than their asymptotic behavior. This paper presents a study of the population diversity when a GA evolves, that is, the population uniformization. The diversity is measured at the genotypic level. Extensive experiments have been performed to observe this uniformization on numerical functions and on some specifically tailored functions, and using various strategies. Among other things, it is shown that the uniformization/exploration phase is always rapid, whichever objective function is to optimize. We study the selection pressure and its crucial impact on the rapidity of the uniformization. Analytical developments of the notion of diversity is under work. Some analysis of the various steps of the GA on the diversity are given.

1 Introduction

Though lots of applications have already been performed showing the broad range of usefulness of GAs, their behavior are still far from being perfectly understood. Theoretical models have been proposed. They have been discussed and criticized, and their limits have been exhibited, though some predictive power has also been achieved with them (see section 5 for review of these works and critics).

We think that the mechanisms at work in GAs should be more thoroughly investigated. Crucial points influencing their behavior should be clearly exhibited and studied in order to make GAs more efficient and to gain know-how to use them. We do not seek an asymptotically true model of GAs but a description of the way GAs behave in the first few thousands, or tens of thousands generations. The idea behind this is that if GAs are used as heuristics to solve actual problems, we are interested in the result the GA will find "quite rapidly". It is nice to know that a GA can find the optimum if we can wait till the end of times. However, we also know that a simple enumerative procedure will do the same job in a finite

*Philippe.Preux@lil.univ-littoral.fr. All correspondance should be directed to the following address: Laboratoire d'Informatique du Littoral, BP 689, 62228 Calais Cedex, France. This paper was published in the proceedings of the ECAI'94 workshop on Applied Genetic and other Evolutionary Algorithms, A. Eiben and B. Manderick and S. Ruttkay (eds), Amsterdam 94

time. Furthermore, the halting problem is not simple for a genetic algorithm: when does it decide to stop ? How can it ever be sure that it found The optimum ? As it does not record the points it has yet sampled, it can not know that it has searched the whole research space. While we perfectly know that enumeration is actually impossible once the research space attains certain dimensions, the previous reflections prompted us with the question of studying the "short-term" behavior of GAs.

In the same time, we are interested in precising the possibilities as well as the limits of a robust optimization process of adaptation, based on the paradigms that encompasses all techniques known today as "evolutionary computation". In the long run, we expect to address question such as *When should we use such algorithms* ? and *to which extent does robustness go* ?

1.1 Evolutionary algorithms vs. "classical" algorithms

There is a fundamental difference between genetic algorithms[1] and classical algorithms[2]:

- classical algorithms are tailored to solve a problem in precise terms: given a set of data, we can, at least in theory, compute precisely the amount of space and time that is required to obtain the result (complexity of the algorithm). Furthermore, the result is always the same, at each execution of the algorithm, for the same set of input data (deterministic and reproducible algorithm). The behavior of the algorithm is in no part due to any randomness;

- a GA simulates a process, that is evolution of a population of organisms in natural environments. In principle, a GA can solve any kind of problem. The only parts that are application-dependent are the encoding of solutions in individuals and the objective function. Then, after initialization of the population, we let the whole system evolve according to its own laws of evolution. After a while, we get the best individual found in the population which, after decoding, gives us the solution found by the algorithm. The evolution process is stochastic. Thus, the GA (and all EA in general) is non deterministic and eventually non reproducible. Furthermore, the result is not guaranteed to be the global optimal existing solution.

These major differences in the nature of these two algorithm classes lead to major differences in their use. Major problems to solve to use a classical algorithm are:

- the design of the algorithm itself. Related to this design are the questions about its computability;

- its complexity in space and time. Here, the questions are: is the problem actually solvable ? Can we actually find the best solution to it ? Must we content ourselves with an approximation ?

[1] and this can be claimed for all kinds of EA, with some minor changes in each case

[2] by *classical algorithms* we wish to speak about those regular algorithms that are found in textbooks such as Knuth's or Aho, Hopcroft and Ullman's that are tightly based on structured programming ideas, notably according to the design methodology

The use of a GA does not raise the same questions. The design part is restricted to find an encoding of individuals and an objective function. Then, simulation parameters should be chosen. It should be pointed out that the complexity issue is no longer relevant here. The complexity in space is generally constant. The complexity in time is generally bounded in the simulation by a parameter (number of trials/generations to run). That might seem a fast treatment of the issue. However, we first point out that, contrary to most classical algorithms, a GA can be halted at any time and give a result (of course, the longer we wait, the more accurate). The second point is that we show in this paper that a certain convergence is rapidly attained w.r.t. the dimensions of the research space $\mathcal{E}$.

To sum up, classical algorithms give the result, but are neither robust nor always easy to design, nor always designable. GA may be regarded as easy to use (however, it may be a false impression to think that there is nothing to design in order to use GAs efficiently), they are robust[3], but they do not ensure that the optimal result will (actually) be found.

1.2 Overview of the rest of the paper

In this paper, our goal is to understand how the population uniformizes during the evolution of a GA in order to enhance its behavior. These enhancements focus on the following two points which are closely related:

- have a more thorough and a more efficient exploratory phase
- avoid premature convergence

By the way, we would also like to figure out the consequences of the following crucial choices when using a GA:

- the encoding of chromosomes
- the objective function

as well as the role of the selection pressure. These three points greatly influence the behavior of GAs and thus deserve more studies.

In this paper, we present our preliminary results and conclusions. First, we present our methodology. We describe our measure of population diversity. Then, in order to have a look at what is going on, we describe some experiments. They are interpreted and we propose some explanations that are confirmed with more experiments. After reviewing some related works, we draw some conclusions about how these studies can be used and how we will continue our work.

Before going any further, we would like to make it clear that for the moment, we have only experimented GAs. However, we have not restricted ourselves to the canonical version of GAs. Hence, we use ranking reproduction strategy, various forms of crossover, elitism, self-adaptation of mutation rates, ... Though aiming at including these fields in our work in the future, we have studied neither genetic programming, nor evolutionary programming nor evolutionary strategies, nor hybrid GA using non binary representation.

[3]at least, apart from the encoding of chromosomes and the objective function

2 Our methodology for studying uniformization

According to the previously stated reflections, we want to study how the population evolves during the execution of a GA. It is obvious that starting with randomly chosen individuals, the population tends to uniformize during time. In order to understand this uniformization, we need some means to measure the degree of uniformity of the population.

In GAs, individuals can be regarded at two levels: genotypic, and phenotypic. These two levels are closely intertwined when the GA is acting. However, some mechanisms only happen at one level or the other. These mechanisms should be clearly distinguished in order to perform an accurate study of the process. Thus, we begin with precising these two levels of action.

2.1 Genotypic and phenotypic levels

The phenotypic level is the most apparent because it is the level where:

- the individuals are seen as data structures of interest,
- the objective function acts,
- the fitness and selection process act.

However, this level is only the mirror of what happens at the lower level of genotypes. At this level:

- the encoding of chromosomes is defined by its length (number of genes),
- genetic operators act,
- the whole evolutionary process acts, blindly, w/o r. t. how individuals are decoded or interpreted at an upper level.

Thus, we must observe the process at the genotypic level because this is the level at which the algorithm actually works. In particular, we must understand the evolution of the genotypic material. In the same time, it is clear that we must always keep an eye on the phenotypic level where the consequences of lower level processes are emerging. We must always try to figure out the consequences at the phenotypic level of what is going on at the genotypic level.

It should also be emphasized that there are consequently two research spaces, one at each level. The phenotypic research space is the space where the function to optimize is searched. For example, for numerical functions, this may be $\mathbb{R}^m$, or a subspace of it. The genotype research space is the space that the GA searches. For a GA processing chromosomes with binary alleles, it is $\{0,1\}^\lambda$. A proper mapping of the phenotypic space onto the genotypic space is crucial in order for the GA to be able to work. We will not go any further about this problem of mapping. Being one of our current research subject, it will be addressed in the last section of this paper. We simply indicate that each time we will speak of the research space $\mathcal{E}$, it should be understood as the genotypic space, except when explicitly stated otherwise.

2.2 Population diversity

We define the *genotypic diversity* as a measure of the diversity of the population chromosomes at the genotypic level. At this level, the distance between two chromosomes is the number of respective genes having different alleles in the two chromosomes. In this paper, we only consider individuals with binary alleles. Thus, the distance between two individuals is their Hamming distance. In order to measure the diversity in the whole population, we take the sum over all pairs (χ_i, χ_j) of individuals in the population P of their Hamming distance $\Delta_H(\chi_i, \chi_j)$

$$\Delta_H = \sum_{\{(\chi_i,\chi_j)\in P\times P, i>j\}} \Delta_H(\chi_i, \chi_j) \tag{1}$$

We will also write $\Delta_H^t(P)$ to qualify the genotypic diversity of population P at time t, or simply $\Delta_H(P)$ if t is unimportant.

2.3 Evolution of the genotypic diversity

Right now, we can examine how Δ_H evolves during the process. At each generation, the population P of ancestors is characterized by its $\Delta_H^t(P)$. Initially, at time $t = t_0$, a population of ν individuals with genotypes composed of λ binary genes is generated at random. Thus, its Δ_H is:

$$\Delta_H^{t_0} = \frac{\nu(\nu-1)\lambda}{4} \tag{2}$$

Then, the GA follows the steps:

reproduction $P \rightsquigarrow P'$. Using stochastic universal sampling, some individuals produce 2 offsprings, some produce 1 offspring and some do not yield any descendance. (On average, each chromosome is assigned $\varphi(i)$ offsprings, where $\varphi(i)$ is

$$\varphi(i) = \eta\text{max} - (\eta\text{max} - \eta\text{min})\frac{i-1}{\lambda-1}$$

with $\eta\text{max} = 2.0 - \eta\text{min}$ and $0.0 \leq \eta\text{min} \leq 1.0$. Of course, $\varphi(i)$ is stochastically rounded for all the individuals to produce an integer number of offsprings. The net effect is that diversity is reduced in the population P' since twin offsprings have their Δ_H equal to 0. Hence, reproduction lowers Δ_H in the descendant population: $\Delta_H(P') < \Delta_H(P)$

operator application

crossover Δ_H is invariant w.r.t. the crossover operator (whichever kind of crossover we use – 1-, 2-, m- points, uniform) because crossover simply exchanges alleles between two chromosomes. Hence, the distance between two respective alleles remains unchanged either swapped or not.

mutation Mutation is applied as a complementing of an allele value. As long as the generation of an allele 0 or 1 are equiprobable, on average Δ_H is also not modified by mutation. Statistical lower and upper bounds of $\Delta_H(\mu(P'))$ may be easily derived according to the mutation rate R_μ:

$$\Delta_H(P') - \frac{R_\mu\nu(\nu-1)\lambda}{2} \leq \Delta_H(\mu(P')) \leq \Delta_H(P') + \frac{R_\mu\nu(\nu-1)\lambda}{2} \tag{3}$$

where $\mu(P)$ denotes the application of the mutation operator to a population P.

inversion though not considered in this paper, we can point out that inversion also leaves the Δ_H of a population unchanged since alleles are not modified when applied

Hence, on average, the application of genetic operators does not modify Δ_H. That is, the diversity of a population before and after the application of genetic operators is the same at the genotype level.

selection as long as the generation gap γ is 1.0, the population of the next generation is composed of the population of offsprings on which genetic operators have been applied. Hence, we have:

$$\Delta_H^{t+1}(P) = \Delta_H^t(\mu(X_{\text{over}}(P'))) \tag{4}$$

where X_{over} denotes the crossover operator.

In general cases where γ is different from 1.0, we do not have yet a correct analytical estimation of Δ_H. Experiments with $\gamma \neq 1.0$ are reported further.

3 First experiments: gaining insight

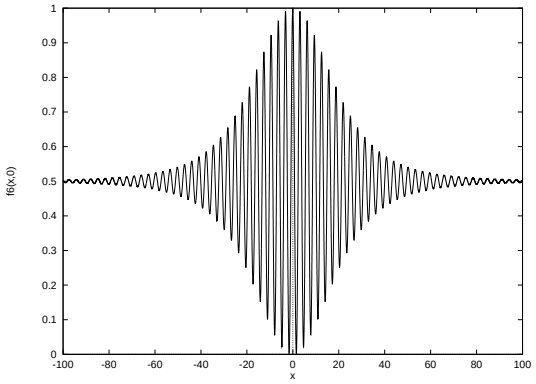

Figure 1: Shape of $f6(x, y)$ when y is set to 0

In order to get some feeling of how Δ_H evolves, we perform some initial experiments with a somewhat classical problem of numerical optimization. These early experiments are briefly reported in this section.

All our experiments are performed using our *Enhanced Grefenstette's Genesis* simulator [28], abbreviated herein as EGG, which is based on J. Grefenstette's *genesis* simulator [17] and has been extended along numerous ways and is still under extension.

Using standard parameters (population size $\nu = 50$, chromosome length $\lambda = 60$, crossover rate $R_\chi = 0.6$, mutation rate $R_\mu = 0.001$, generation gap $\gamma = 1.0$, stochastic universal sampling with minimum rank $\eta_{\min} = 0.75$ and maximum rank $\eta_{\max} = 2 - \eta_{\min}$), we study the function $f6$ used in [34]:

$$f6(x, y) = 0.5 - \frac{(\sin\sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$$

This function is symetrical around the z axis. The maximum is searched. It is obtained for $x = y = 0$, where $f6(0,0) = 1$.

A chromosome is composed of two parts, one part which encodes the value of x, the other part the value of y. Each value ranges from -5.12 to 5.12. It is binary encoded, the range being uniformly cut into $2^{\frac{\lambda}{2}}$ pieces. With λ set to 60, this leads to a resolution of around 10^{-9} for each value. Gray code is used.

The objective function is written straightforwardly: it simply returns the value $f6(\chi)$ where χ is a chromosome to evaluate. As can be seen on figure 1, the shape of $f6$ is characterized by a lot of local maxima where the algorithm may be easily trapped.

3.1 The uniformization

Whether or not the optimum is actually found is not the point here. Whatever the region of $\mathcal{E}$ the process converges towards, Δ_H evolves in the same manner. The figure 2 represents the value of Δ_H all along the generations. It represents the average Δ_H on 100 experiments, each performing 100000 trials.

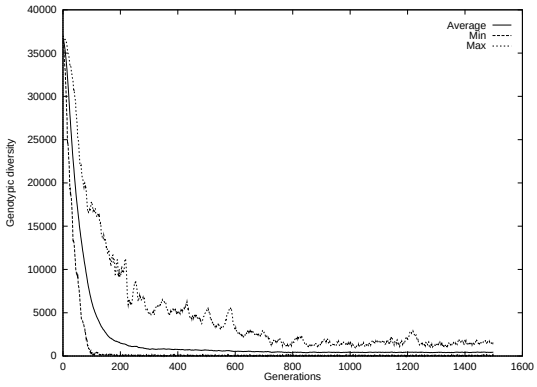

Figure 2: Evolution of Δ_H when solving $f6$. The three curves indicate the average on 100 experiments as well as the upper and lower bounds of Δ_H. Only the first 1500 generations are represented here which show the uniformization phase and the beginning of the steady-state phase. At steady-state, Δ_H remains around 420. The parameters of the simulation are those previously mentionned.

At first glance, we can easily distinguish two phases:

- a rapid decrease of Δ_H during the first 300 generations of evolution,
- an ever-lasting steady-state phase during which Δ_H remains in a constant range.

We performed lots of experiments on lots of objective functions (numerical or not) using various parameter sets (described further). It is striking that:

- the Δ_H evolution curve is always the same, with these two clearly distinguishable phases,
- the first phase is always very rapid. It ends before or around the 1000^{th} generation.

We will call the first phase *uniformization*, the second *steady-state*. These two phases correspond to the well-known exploration and exploitation phases of GA searches.

Uniformization acts at a very high rate: the genotypic diversity decreases at an exponential rate.

The steady-state is characterized by a *noise level*, the average value of Δ_H during this phase. This noise is discussed and explained in section 4.1. For the reported experiment, $\bar{\Delta_H}^{t>1000} = 420$ and it characterizes the diversity of the population which has been achieved. This is to be compared with $\Delta_H^{t_0} = 36750$; the diversity has been lowered by a factor close to 100.

3.2 Experimenting various strategies

We also performed experiments using the following strategies:

elitism EGG implements two kinds of elitism:

- usual elitism where the best individual is necessarily retained from the current generation to the next
- usual elitist strategy augmented with the retainment of the best individual binary complement. This strategy was proposed by J. Grefenstette [18] as an enhancement of the canonical GA in order to avoid deceptive traps

self-adaptation of mutation rates this strategy is inherited and adapted from evolution strategies. T. Bäck introduced it in [2] in the realm of GAs. For the sake of brevity, we describe only our implementation. The interested reader is referred to [2] and [3] for further explanations.

This strategy consists in having one mutation rate associated to each individual. Hence, there is no longer one mutation rate for all individuals. This mutation rate is used to mutate the genes of the individual it belongs to. Furthermore, the mutation rate mutates itself according to its own value. To sum up, mutation is applied according to the following scheme:

for each individual χ in the population

- decodes the part of the genotype that encodes the mutation rate of χ. This gives a mutation rate μ_1
- mutates the encoded mutation rate itself with rate μ_1
- decodes this newly mutated mutation rate. This gives μ_2
- mutates the genotype (apart from the mutation rate section) with rate μ_2

endfor

initially biased population EGG permits the initialization of the whole population with a copy of a single random genotype. This is useful to study the influence of the initial population bias on the behavior of the GA.

All the tests we performed using these strategies gave the same results. We also performed long running simulations (long with regard to the generation at which steady-state is typically

attained) which show that steady-state is stable and uniformization can be considered as finished. We also performed simulations varying the population size parameter ν as well as the generation gap γ. All these simulations gave the same behavior, that is the same noise level at steady-state. It is also interesting to point out that a biased initial population with $\Delta_H^{t_0} = 0$ first enters a diversification phase followed by steady-state similarly as when using unbiased initial population.

In the next section, we study these two phases.

4 Uniformization

Grounded on the previous remarks, we explain the shape of the Δ_H curve, design and perform some more experiments to confirm these hypothesis.

4.1 Noise level and selection pressure

When the steady-state phase is attained, a part of the genotype of all chromosomes has been fixed. These fixed bits constitute a schema σ which characterizes the region of $\mathcal{E}$ towards which the GA has converged. The other bits of the individuals may continue to flip; all the instances of σ broadly have the same fitness. Hence, all these combinations have the same odds to survive. Furthermore, the set $\mathcal{S}$ of the instances of σ, $\mathcal{S} = \{\chi \in \sigma\}$ is closed by crossover. That is,

$$\forall(\chi_i, \chi_j) \in \mathcal{S} \times \mathcal{S}, X_{\text{over}}(\chi_i, \chi_j) \in \mathcal{S} \times \mathcal{S}$$

The fact that $\mathcal{S}$ is closed by crossover is justified by the remark that if it was not closed, then the phase would not be steady.

At steady-state, the odds to generate better fitting chromosomes by mutation are reduced; otherwise, mutation would have acted yet, before the steady-state phase was entered, and already produced better fitting chromosomes. Hence, $\mathcal{S}$ is stable w.r.t. the crossover operator as well as w.r.t. the mutation operator. The number of bits that are allowed to flip without any consequences on the performance of the individuals will be called β_{lim}.

β_{lim} is then closely related to the steady-state noise level. Let β the number of bits that may change in the individuals of a population (this is the number of undefined bits of the schema σ previously discussed), Δ_H of this population is:

$$\Delta_H = \begin{cases} \frac{\nu(\nu-1)\beta}{4} & \text{if } 2^\beta > \nu \\ (1 - 2^{1-\beta})\nu^2 & \text{if } 2^\beta < \nu \end{cases} \tag{5}$$

β_{lim} is influenced by two factors:

- the thinness of the peak where the optimum towards which the population uniformizes lies
- the encoding of chromosomes, more precisely, the number of equally fitting individuals that a region of the phenotypic research space may hold

These two factors are closely related to each other. For a given resolution of chromosome encoding, we can define a minimum thinness of peaks that may be discovered by the GA.

We have observed that a peak that may hold less than $\frac{3}{2^{30}}$ genotypic different chromosomes is hard to find. Works are being done to precise this limit.

There is an other factor which has not yet been pointed out but has a crucial effect on the uniformization, namely the selection pressure. High selection pressure implies big differences between the fitness of the best individuals and the one of the worst. Hence, more offsprings are allocated to the best, leading to a faster uniformization phase. Using stochastic universal sampling, we define the selection pressure ρ :

$$\rho = \varphi\max - \varphi_{\min} = \eta\max - \eta_{\min} \tag{6}$$

4.2 Experiments

To further investigate, we perform tests using numerical functions in order to check the results we obtained with $f6$, and tailored functions to confirm our hypothesis.

4.2.1 Test functions

De Jong's functions [9], Corona's function [6], Rosen's nowhere differentiable function [33] give the same kind of results as $f6$[4]. We will not spend more time considering them.

In order to investigate the relation between the thinness of the optimum peak and the encoding of chromosomes as well as the effect of the selection pressure, we performed tests with the following classes of objective functions:

- a "hole function" which shape is very simple: a deep hole in an otherwise flat or slightly inclined plateau. We perform experiments with various hole diameters ranging from a big hole to a little one, to the limits of chromosome resolution
- a "flat function" which shape is completely flat. The objective function simply returns the same constant value for all sampled points. These experiments were performed in order to observe the behavior of the GA when absolutly no information is given by the objective function according to which region of $\mathcal{E}$ contains the optimum
- "random functions" which have no definite shape. The objective function returns a value computed at random for each point. The random values are uniformly distributed in the range $[0, 1[$. This function was included in order to "fool the GA" with an objective function that does not return any valid information about the location of the optimum.

4.2.2 Sketch of results and discussion

We sketch the results we obtained with the tailored functions and discuss the extent to which they conform with previously stated explanations.

hole functions aimed at testing whether the noise is due to the action of crossover on a class of equally fitting chromosomes. The noise level is hence directly related to the cardinality of $\mathcal{S}$. The results are displayed in table 1. They clearly show a relation between the noise level and the number of chromosomes that the hole may hold (actually, the noise level is linearly related to the logarithm of the number of chromosomes).

[4] see appendix for Rosen's and Corona's functions

Hole radius r	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	∞
Number of different genotypes that may fall in the hole N_g	3	21	210	2100	20993	209921	2099202	20992020	2^{30}
Cardinality of $\mathcal{S} : \mathcal{C}$	1	8	128	2048	16384	131072	2097152	16777216	2^{30}
$\beta_{\text{lim}} = \log_2 N_g$	1.35	4.40	7.71	11.03	14.36	17.68	21.00	24.32	30
Noise level N_l^{th}	827	2695	4722	6756	8795	10829	12862	14896	18375
Noise level N_l^{exp}	462	1164	2020	2646	3593	4371	4982	5816	8100
$\beta = \log_2 N_l^{exp}$	0.75	1.90	3.30	4.32	5.87	7.14	8.13	9.50	12.98
Ratio $\frac{\beta_{\text{lim}}}{\beta}$	1.80	1.97	2.38	2.59	2.47	2.50	2.60	2.58	2.31

Table 1: Theoretical (β_{lim}, $\mathcal{C}$, N_l^{th} rows) and experimental (N_l^{exp}, β rows) noise level for the hole functions. Given a hole radius r, the number of different genotypes that correspond to values situated in the hole N_g is deduced as well as β_{lim}, the cardinality of $\mathcal{S}$, $\mathcal{C}$, and the theoretical noise level N_l^{th}. N_g depends on the phenotypical value range and on λ. $\mathcal{S}$ is the sub-set of these N_g chromosomes that is closed by crossover. β_{lim} is the logarithm of N_g. For the experimental results, the noise level N_l^{exp} is measured, from which β is deduced. The right-most column ∞ holds the data for the flat function which can be considered as a hole of infinite diameter. In these experiments, the chromosome phenotypical value ranges from -5.12 to 5.12, $\lambda = 30$, $\eta_{\min} = 0.75$ and $\nu = 50$.

flat function is an extreme case where all points of $\mathcal{E}$ are equally fitting. With this regard, it may be considered as a hole function with an infinite diameter. As such, the flat function is included in table 1 in the column with an ∞ diameter.

The first remark is that even in this case, the GA population uniformizes. According to the viewpoint we adopt, that may be embarrassing: even if there is nothing to find in the research space, the GA "finds" something. However, it should be pointed out that the noise level is very high. That may give a clue to detect that the GA did not converge towards a precise region of $\mathcal{E}$.

The second point concerns the effect of the selection pressure on the uniformization. Figure 3 shows the variation of the steady-state noise level with regard to $\eta_{\min}$. When the selection pressure is high ($\eta_{\min} \to 0.0$), the noise level, hence the diversity is low, and conversely. Again, we observe that the flat function gives an upper bound of the noise level for hole functions, and for any other function as well, due to its flatness.

random function is an other kind of extreme case in that it has no continuous landscape. Hence, the information gathering performed by the GA during its search can not properly act: if a sampled individual performs well, there is no information that can be deduced for surrounding points. We observe that the population uniformizes rapidly around a highly fitting individual.

5 Other approaches

We briefly review some works aiming either at understanding how GAs work, or at enhancing the exploratory phase. For those works falling in the first category, we essentially show why they are inadequate according to our goal.

Though the oldest model and still widely accepted, it becomes quite clear that the schema theorem, introduced by J. Holland [19], is inadequate to explain and predict the behavior of

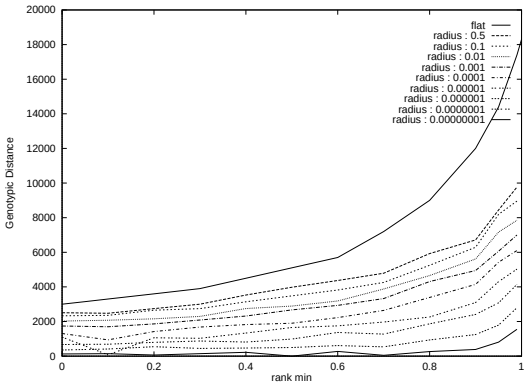

Figure 3: Variation of the genotypic distance vs. $\eta_{\min}$. The impact of varying $\eta_{\min}$ on hole functions and on the flat function is displayed here. When $\eta_{\min}$ gets bigger, that is, when the selection pressure decreases, the diversity in the population raises, and conversely. Using our definition of selection pressure $\rho = \eta_{\max} - \eta_{\min}$, $\rho = 2$ for $\eta_{\min} = 0.0$, $\rho = 0.0$ for $\eta_{\min} = 1.0$, with a linear decrease between these bounds.

GAs. As pointed out in [18, 25, 16], the schema theorem relies on the static building-block hypothesis which does not take the dynamic of the process into account. Though extensions of the schema theorem have been performed either for non binary encoded chromosomes (e.g., see [1]) or for other structures than schemata (predicates [39], formae [30, 31], interval for real coded individuals [13]). Other works on the behavior of GAs focus on giving a proof of convergence of the algorithm, either by enhancing the schema theorem [43][41, 26, 40], or by modeling the behavior of the GA with Markov's chain [11] [29] [8]. A very interesting review of all these works is available in [27].

Though interesting, it should be noted that all these works mostly tend to describe the process rather than explaining the way it works. In this sense, it is of too high level to fully understand the miscellaneous effects of operators, or various strategies or biases. Furthermore, these works generally prove convergence when the number of generations tends toward infinity, which is not of practical interest. It seems quite paradoxical to us to regard GAs as potential heuristics from which results are expected to be achieved rapidly and studying their behavior when $t \to \infty$ and content ourselves with a proof that at that "time", we are sure that the optimum will have been found.

For their part, works around enhancing the exploratory phase have mostly focussed either on avoiding all chromosomes to rush towards one optimum (multi-niched GAs), or on iteratively letting the GA converge and trying to make it converge towards different points each time [21], [12], [24], [4]. The first path has been treated for a long time by preventing too similar individuals to recombine (incest prevention) or to avoid too much individuals of being similar (sharing techniques). The interested reader is referred to [37] which have recently reviewed these techniques.

However, it is crucial that already found information are not lost when the evolution process is restarted. Otherwise, the restart is simply a new run, independant from the previous evolution(s) and forgetting already gleaned information.

6 Conclusion and Future work

In this paper, we have reported some on-going works aiming at understanding how a GA population uniformizes. We aim at obtaining practical results about the behavior of GAs, rather than theoretical asymptotical results. The goal is to derive a better know-how to use GAs to solve problems efficiently and precisely (in order to find the global optimum and avoid the trap of premature convergence), that is, to gain insight on how to use GAs as heuristics.

An important experimental result we have found is this very fast uniformization of the population. Its immediate consequence is that if the region[5] where the optimum lies is not found very quickly (after the first few hundred generations for the population size we used), it becomes quite improbable that it will be found later. Once uniformization is obtained, the process turns out to be an exploitation.

The uniformization has been linked to the selection pressure, while the remnant noise level has been related to the action of crossover on a set of genotypes of equally fitting individuals closed by crossover and insensible to mutation.

Exploration may be enhanced either by trying to make it more rapid to find the interesting region, or by trying to make it last longer in order to search the research space more thoroughly.

The first possibility leads either to seek the best GA parameters (dynamically adapted operator rates, γ, ν, ... reproduction strategy, selection strategy, elitism, ...), or to tune the objective function as well as the encoding of chromosomes. The second possibility is closely related to the selection pressure. Our work currently focusses on these points and we describe now briefly these directions:

encoding of chromosomes , that is the mapping of the phenotypic space onto the genotypic space. It is now known that proper encoding may turn any function optimization to be as easy as the 1's counting problem [42]. When the binary encoding is assumed, the classical debate is between pure binary encoding vs. Gray encoding. In both cases, the weigth of the bits leads to have dramatically different consequences whether it acts on a part or on an other of the genotypes. Though mutation strikes genes at random, we obtained surprising results when we used Gray coding with shuffled bits w.r.t. the noise level. It was measured at a 10 times higher level than for non-shuffled bits. This fact deserves more studies and it is one point of our current studies. We are also performing studies concerning other non binary encodings of chromosomes.

objective function A better design of the objective function may be an other way of tackling the problem. Only nearly obvious objective functions have been used so far in the case of the optimization of numerical problems where a direct use of the function itself is usual. This straightforward use of the function being not so clear when tackling symbolic problems (see for exmple Whitley's treatment of the Traveling Salesman Problem in [45] or De Jong and Spears' study of NP-complete problems [10]), more diverse functions have been studied, coupled with more subtile encodings of chromosomes.

[5] in fact, a region is a neighborhood (in the sense of operator action) of the optimum from where the optimum is directly achievable. The region of the optimum is something like the peak where it lies. The peak has to be detectable, leading to the problems of chromosome encoding and its "resolution"

selection pressure We think that since the selection pressure is of crucial importance for the GA based evolutionary process itself, independently from the particular function it optimizes, it should be much more studied, understood and mastered. A proper balance of the selection pressure is needed: too high selection pressure leads to premature convergence, the GA having not enough time to explore; too low selection pressure transforms the GA into an enumerative procedure. We think that varying selection pressure into the course of evolution may lead to interesting results.

It should be pointed out that works focussing on topological properties of fitness function landscape are of major interest (see e.g. [23], [22], [20], [7]). Understanding how this landscape helps or traps a GA should give us clues w.r.t. how GAs work and GA capabilities to discover the optimum. Following others, we have engaged ourselves in the study of topologically simple functions[6]. However, it is clear that the predictive power of this kind of modeling would remain weak for fitness function having a landscape which is not topologically knowable, neither would it help us to choose the best (or, at least, good ones) objective function for a given problem.

To sum up these points, we strongly think that on behalf of the understanding of the process itself which is still not clear, the application-dependent parts deserve much more work. Among them, the roles of chromosome encoding and of the objective function are rarely emphasized and their understanding is still very weak.

Besides the points already raised, we will extend our work by performing some experiments on symbolic problems. Our very first experiments show a similar GA behavior on satisfiability problems. However, they have to be more thoroughly performed before being able to draw any valid conclusion. We will also continue our effort w.r.t. the theoretical developments of our model. Our first objective is to complete an analytical model of the flat function. Furthermore, as soon as the behavior of a single, canonical-or-so GA will have gained in clarity, we aim at studying multi-niched GA and parallel GA, as well as hybrid GA using non binary representation and dedicated operators and more generally other kind of EA (evolution strategies, evolutionary programming, genetic proramming, ...).

References

[1] H. J. Antonisse. A new interpretation of the schema notation that overturns the binary encoding constraint. In *[35]*, pages 86–91, 1989.

[2] Thomas Bäck. Self-adaptation in genetic algorithms. In *[38]*, pages 263–271. MIT Press, Cambridge, MA, USA, 1991.

[3] Thomas Bäck. Optimal mutation rates in genetic search. In *[15]*, pages 2–8, 1993.

[4] David Beasley, David R. Bull, and Ralph R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.

[5] Richard K. Belew and Lashon B. Booker, editors. *Proc. of the Fourth International Conference on Genetic Algorithms*, La Jolla, CA, USA, July 1991. Morgan Kaufmann, San Mateo, CA, USA.

[6] A. Corona, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm. *ACM Transactions on Mathematical Software*, 13(3):262–280, September 1987.

[7] Joseph C. Culberson. Mutation-crossover isomorphisms and the construction of discriminating functions. *Evolutionary Computation*, 2(3), 1994.

[6] simple for human's eyes at least

[8] Thomas E. Davis and Jose C. Principe. A simulated annealing like convergence theory for the simple genetic algorithm. In *[5]*, pages 174–183, 1991.

[9] Kenneth A. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975. *Dissertation Abstracts International* 36(10), 5140B (University Microfilms 76-9381).

[10] Kenneth A. De Jong and William Spears. Using genetic algorithms to solve NP-complete problems. In *[35]*, pages 124–132, 1989.

[11] A. E. Eiben, E. H. L. Aarts, and K. M. V. Hee. Global convergence of genetic algorithms: a markov chain analysis. In *[36]*, pages 1–12, October 1991.

[12] Larry J. Eshelman. The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In *[32]*, pages 265–283, 1991.

[13] Larry J. Eshelman and J. David Schaffer. Real-coded genetic algorithms and interval schemata. In *[44]*, pages 187–202, 1992.

[14] David B. Fogel, editor. *Proc. of the* 3^{rd} *Annual Conf. on Evolutionary Programming*, San Diego, CA, February 1994. Evolutionary Programming Society.

[15] Stephanie Forrest, editor. *Proc. of the Fifth International Conference on Genetic Algorithms*, Urbana-Champaign, IL, USA, July 1993. Morgan Kaufmann, San Mateo, CA, USA.

[16] Stephanie Forrest and Melanie Mitchell. Relative building-block fitness and the building-block hypothesis. In *[44]*, pages 109–126, 1993.

[17] John J. Grefenstette. *A User's guide to Genesis Version 5.0*, October 1990. The *genesis* package is available via anonymous ftp on `ftp.aic.nrl.navy.mil:pub/galist/src/ga/genesis.tar.Z`.

[18] John J. Grefenstette. Deception considered harmful. In *[44]*, pages 75–91, 1993.

[19] John H. Holland. *Adaptation in Natural and Artificial Systems.* Michigan Press University, Ann Arbor, MI, 1975.

[20] Terry Jones. A model of landscape, 1994.

[21] K. Krishnakumar. Micro-genetic algorithms for stationary and non-stationary function optimization. In *SPIE's Intelligent Control and Adaptive Systems Conf.*, 1989.

[22] Marc Lipsitch. Adaptation on rugged landscapes generated by iterated local interactions of neighboring genes. In *[5]*, pages 128–135, 1989.

[23] Bernard Manderick, Mark de Weger, and Piet Spiessens. The genetic algorithm and the fitness landscape. In *[5]*, pages 143–150, 1989.

[24] Keith Mathias and Darrell Whitley. Remapping hyperspace during genetic search: Canonical delta folding. In *[44]*, pages 167–186, 1993.

[25] Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscape and GA performance. In *[38]*, pages 245–254, 1991.

[26] Allen E. Nix and Michael D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and AI*, 5:79–88, 1992.

[27] Charles C. Peck and Atam P. Dhawan. A review and critique of genetic algorithm theories. Technical Report TR-153/6/93/ECE, Department of Electrical and Computer Engineering, University of Cincinnati, Cincinnati, OH 45221, USA, June 1993.

[28] Philippe Preux. *Description of Special Features of Enhanced Grefenstette's Genesis.* Laboratoire d'Informatique Fondamentale de Lille, Université de Lille I, 59655 Villeneuve d'Ascq Cedex, France, April 1994. (working paper).

[29] Y. Rabinovich and A. Wigderson. An analysis of a simple genetic algorithm. In *[5]*, pages 215–221, 1991.

[30] Nicholas J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–205, 1991.

[31] Nicholas J. Radcliffe. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 1994.

[32] Gregory J. E. Rawlins, editor. *Workshop on the Foundations of Genetic Algorithm and Classifier*, Bloomington, IN, USA, 1991. Morgan Kaufmann, San Mateo, CA, USA.

[33] Bruce Rosen. Function optimization based on advanced simulated annealing. In *Proc. of Workshop on Physics and Computation, PhysComp'92*, Dallas, TX, 1992.

[34] J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *[35]*, pages 51–60, 1989.

[35] J.D. Schaffer, editor. *Proc. of the Third International Conference on Genetic Algorithms*, Bloomington, IN, USA, 1989. Morgan Kaufmann, San Mateo, CA, USA.

[36] H-P. Schwefel and R. Männer, editors. *Proc. of the First Parallel Problem Solving in Nature*. Springer-Verlag, Berlin, 1991.

[37] William M. Spears. Simple subpopulation schemes. In *[14]*, 1994.

[38] Francesco Varela and Paul Bourgine, editors. *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. MIT Press, Cambridge, MA, USA, December 1992.

[39] Michael D. Vose. Generalizing the notion of schema in genetic algorithms. *Artificial Intelligence*, 50:385–396, 1991.

[40] Michael D. Vose. Modeling simple genetic algorithms. In *[44]*, pages 63–73, 1993.

[41] Michael D. Vose and Gunar E. Liepins. Punctuated equilibra in genetic search. *Complex Systems*, 5:31–44, 1991.

[42] Michael D. Vose and Gunar E. Liepins. Schema disruption. In *[5]*, pages 237–242, July 1991.

[43] Darrell Whitley. An executable model of a simple genetic algorithm. In *[44]*, pages 45–62, 1993.

[44] Darrell Whitley, editor. *Proc. of the Workshop on Foundations of Genetic Algorithms*, Vail, CO, USA, 1993. Morgan Kaufmann, San Mateo, CA, USA.

[45] Darrell Whitley, Thimothy Starkweather, and D'Ann Fuquay. Scheduling problems and traveling salesman: The genetic edge recombination operator. In *[35]*, pages 133–140, 1989.

Appendix

Rosen's function

$$f(X) = \prod_{k=1}^{D} (1 + k \sum_{n=0}^{\beta} \frac{\left| 2^n x_i - \lfloor 2^i x_i \rfloor \right|}{2^n})$$

where $D = 4$, $\beta = 60$ (normally, $\beta = \infty$) and $-1000 \leq x_i \leq 1000$.
When β tends towards ∞, $f(x)$ tends towards $|x|$. This function is nowhere differentiable.

Corona's function

$$f(X) = \sum_{i=1}^{n} \begin{cases} (t_i sgn(z_i) + z_i)^2 c d_i \text{ if } |x_i - z_i| < |t_i| \\ d_i x_i^2 \text{ otherwise} \end{cases}$$

where $z_i = \lfloor \left| \frac{x_i}{S_i} \right| + 0.49999 \rfloor sgn(x_i) S_i$, $S_i = 0.2$, $t_i = 0.05$, $1 \leq i \leq n$, $d_i = \{1.0, 1000.0, 10.0, 100.0\}$, $c = 0.15$, $-1000 \leq x_i \leq 1000$, $n = 4$.
This function has $10^{5n} - 1$ local minima.