# An Evolutionary Approach for Time Dependant Optimization

**Philippe Collard** **Cathy Escazut** **Alessio Gaspar**
University of Nice-Sophia Antipolis, Laboratory I3S — CNRS
250 Av. A. Einstein, Sophia Antipolis, 06560 Valbonne FRANCE
email: {pc,escazut}@unice.fr, gaspar@essi.fr

## Abstract

*Many real-world problems involve measures of objectives that may be dynamically optimized. The application of evolutionary algorithms in time dependent optimization is currently receiving growing interest as potential applications are numerous ranging from mobile robotics to real time process command. Moreover, constant evaluation functions skew results relative to natural evolution so that it has become a promising gap to combine effectiveness and diversity in a Genetic Algorithm.*

*This paper features both theoretical and empirical analysis of the behavior of Genetic Algorithms in such an environment. We present a comparison between the effectiveness of traditional Simple Genetic Algorithm (SGA) and the Dual Genetic Algorithm (DGA) which has revealed to be a particularly adaptive tool for optimizing a lot of diversified classes of functions. This comparison has been performed on a model of dynamical environment which characteristic are analyzed in order to establish the basis of a testbed for further experiments. At last, we discuss fundamental properties that explain the effectiveness of the dual paradigm to manage dynamical environments.*

## 1 Introduction

In this first part, we briefly present basics of genetic algorithms. The reader can refer to [4] or [8] for a more detailed presentation. Then, we give an overview of works in the area, and introduce the Dual Genetic Algorithm (DGA) as a new approach to dynamic optimization. At last, we define the concept of *Dynamical Environment.*

### 1.1 Genetic Algorithms

Genetic Algorithms (GAs) are evolutionary optimization methods inspired from natural selection and introduced in the sixties by J. Holland ([8]). According to the neo-darwinian evolution theory, they apply genetic like operators (such as crossover and mutation), and natural selection to a population of individuals coding in their(s) chromosome(s) (generally a $\lambda$ long binary string) a potential solution for the considered problem. Each individual receives a *Fitness* value which quantifies how much it is able to solve it. When optimizing with a GA, the reward is simply computed by applying a fitness function to the value encoded by the chromosome. If we look for a global optimum, higher the value of our chromosome when decoded and applied the function is, higher will be its fitness and lower will be the probability to be facing a local optimum.

This way, the GA evolves, over consecutive generations, a population where best individuals spread thanks to the exchange during crossover of genetic material belonging to selected parents. This genetic exploration of the search space is repeated until a halting criterion such as convergence ratio or time constraint is met. Figure 1 details the steps of the Simple Ge-

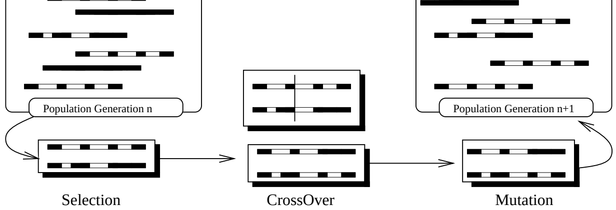


**Figure 1. Standard Genetic Algorithm**

netic Algorithm (SGA). At each generation, a *Selection* mechanism reproduces individuals according to their fitness, then parents are recombined by the mean of a *crossover* operator which produces sons that may be *Mutated*. As a result, the final population may have largely converged toward the best individuals found to time, and thus only contains instances of them.

Selective pressure and recombination of genetic material between individuals has lead, as expected by J. Holland, to a robust and adaptive evolutionary process. The following section presents some of the works already done in the area and underline the fact that dynamical optimization shares its problems and solutions with a lot of other research gap of the genetic algorithms community.

### 1.2 Evolutionary approaches to Dynamical Optimization

In this section, we consider two approaches aimed at applying a genetic algorithm to the optimization of dynamical functions. In fact, the problem considered here is more general that it seems at a first sight. Our objective is to preserve diversity; this is a fundamental property for a GA to successfully handle the optimization of functions such as multimodal or deceptive ones where finding out apparently good individuals may be respectively insufficient or misleading.

In this paper we will only focus on dynamical aspects but it is worth seeing the potential interest of this work. Some attempts on dynamical environments optimization have directly used techniques originally designed for other classes of problems. It has provided interesting results underlining the link between these research areas. For example, Goldberg has shown that the use of *diploidy and dominance* ([5]) constitutes a relevant approach for managing non stationary fitness functions. It is worth noticing that diploidy has been early introduced to handle deceptive functions by preserving non-optimal chromosomes in a recessive form not expressed in the population until necessary. This preservation of diversity fulfills two different objectives; it brings an improved robustness against deceptivity and opens perspectives to dynamical fields.

More recently, J.J. Grefenstette has designed a new operator inspired from mutation and called *hypermutation* ([7]). This new genetic operator selects at each generation a constant amount of individuals and replace them by random ones. That way, diversity is preserved and allows the GA to find each optimum over time. We can notice that the GA's dynamics is almost perturbed and that it can no more reach a stationary state even when confronted to static environment. In order to experiment its method Grefenstette has chosen a dynamical environment featuring 14 tridimensional gaussians, going up and down at random regularly over a bidimensional search space. Our own testbed is inspired from it but we have decided to restrict to a basic version in order to present a deeper analysis of the behavior of both environment and GA. It should be inversely interesting to apply hypermutation to GA-hard reputed problems in order to establish its effectiveness and to compare its usefulness in both domains.

In the following, we propose a new approach to dynamic optimization relying on Dual Genetic Algorithms that have already featured enhanced effectiveness on a large variety of areas where GAs were facing theirs limits.

### 1.3 Dual Genetic Algorithms

Dual Genetic Algorithms offer an improved robustness against most intrinsic weaknesses of GAs ([1], [3], [2]). They are based upon the introduction of a *transliteration* phase during the evaluation of a chromosome. This one may be compared with the RNAm transcription in natural biology as it introduces a *gene expression* mechanism. Each chromosome is added a meta gene in front of it. For a value of '0', the meta gene is inactive and does not change the transcription of the following genes. But, when activated, we transcribe each of the following genes simply by considering its complementary allele. This expressed phenotype will be evaluated by the fitness function we want to optimize. For example, if we consider the chromosome [011] it can be tranliterated by both $[\dot{0}\ 011]$ or $[\dot{1}\ 100]$.

Chromosomes handled by a DGA are thus $\lambda + 1$ genes long but translate to $\lambda$ long strings for evaluation. This clearly involves two facts: at first we have doubled, for a binary allelic alphabet, the size of the search space. But this one has also become symmetrical and now the GA works on populations that may contain symmetrical strings, called *dual chromosomes*, featuring identical fitness (as they are transliterated to same phenotypes) while having complementary genotypes. DGAs feature interesting properties directly stemming from the use of such symmetrical structures combinated with the dynamics of a SGA. Among these, let us consider their ability to converge from a phenotypical point of view while keeping complementary chromosomes in their population. A special operator called *mirroring* that transforms any chromosome in its dual chromosome with a constant rate $\tau$, helps at keeping a minimal amount of diversity in the population. We will see later that this diversity can be used by the crossover operator to keep acting on dual pairs of chromosomes even when the GA has converged and so to introduce new individuals without the help of mutation.

Preserving diversity makes DGAs to be great candidates for managing in environment featuring a fitness function changing over time.

### 1.4 Dynamical Environments

S.A. Kaufman ([9]) has underlined that interest of the study of dynamical environments by modifying his $NK$ landscapes to allow them to suffer from variations of a given environment. His approach has consisted in affecting each site by $K$ others sites of the system itself and $W$ from the outside world. Thereby, changes happening in the environment can have drastical consequences on the modelled autonomous system. Our approach of dynamical environment will be more general, trying to consider what are the fundamental features of such environment and measure their correlation to a notion of difficulty for example.

For an optimization purpose, Dynamical Environments characterize fitness functions having theirs optima changing over time. They represent a particular class of functions that have not yet been deeply studied. A human being is not only adaptive through generations, but it is also adaptive over time at the human life scale. By confronting artificial systems to changing environments we take a chance to emphasize their adaptiveness at a life duration scale and so climb up a step toward the design of real biomimetic systems. We're at the very beginning of these investigations, so our first effort will mainly consist in a formal characterization of this kind of environment. Such a dynamical fitness function may be characterized in two ways.

From a *static* point of view, the function is characterized by its features at some date (multimodality, deceptiveness, etc); in the following, we will refer to them as *static characteristics*. We can, in the other hand, characterize dynamical environments by the nature (frequency, range, synchronism, etc) of the changes applied to the function over time. Those features will be referred to as *Dynamic Characteristics*. While the static features has been intensively studied, the dynamic ones remain altmost unexplored to date. In the following, we propose to discuss the main dynamical parameters for such a function.

In order to only focus on the difficulty introduced by the dynamic nature of this function, we will keep its static properties constant and easy over time, thus using a unimodal and non-deceptive function. If we consider the function at a particular date, this leads to an easy challenge for genetic optimization. In this way, all troubles encountered should only be explained by the *transitions* over time. Concerning the dynamical features, the *transition speed* appears, at first sight, as an important parameter. It will be expressed by the number of generations during which the optimum remains unchanged. This obviously leads us to distinguish functions featuring a *uniform transition speed* from other, featuring a *non-uniform transition speed* that may reveal to be purely random or obeying particular laws. We will restrict to a function with a uniform and constant transition speed; each consecutive optimum will remain optimal for an identical amount of generations. Always for simplicity, we will consider each point of the search space $\Omega$ as optimal one after the other.

This choice reveals an other important dynamical characteristic; the *transition's difficulty* will be expressed by the hamming distance between two consecutive optima. In fact, our function will change regularly its optima over time by enumerating $\Omega$. The nature of this enumeration will entirely determine the difficulty of the transition; for example, if we order the search space according to a Gray code, hamming distance will keep constant between consecutive optima (equal to 1). In such a case, the mutation operator will allow a rapid and easy adaptation and, in order to design a relevant function, we should thus accelerate the transition speed to increase its difficulty. Our choice is to keep a reasonable transition speed while using a traditional binary coding for enumerating $\Omega$. With such an encoding, all transitions will be different (hamming distances varying from 1 to $\lambda$[1]) and will offer a variety of *transition difficulty* so that the GA will restart a genetic exploration for each transition in order to find the next optimum. An opportune random mutation will no more be sufficient to keep hanging on consecutive optima.

This choice implies that, in order to follow the environment's changes, a GA must incorporate a composite strategy mixing convergent and divergent dynamics. Results will bring us an empirical observation that DGAs feature such a dynamics.

## 2 A Minimal Dynamic Problem

This section provides the definition of a simple dynamical problem on which we have performed simulations of the behavior of both SGA and DGA in order to establish the opportunity of using a DGA. The early version of this work has been briefly presented in [2]; here, we summarize this result to stress the advantages brought by a DGA and then extend them in the following.

[1] $\lambda$ as the chromosome's length.

### 2.1 A simple model

Our Minimal Dynamic Problem will be defined over string of length $\lambda = 2$ as a minimal dynamical environment. The plotting 2 shows the function used in our simulations; each of the $2^\lambda$ possible chromosomes is rewarded by a fitness changing over time and ranging from 0 to 1 for the optima. As shown in figure 2, each individual will be consecutively the optimum over time. This property is simply induced by the *translation* of a gaussian-like fitness function over the search space $\Omega$ through time. By using only a few individuals we get a function with a good fitness differentiation between individuals. Thus, when a transition occurs, the previous optimum's fitness is decreased of a relevant amount, encouraging the GA to rapidly find its successor. We will see that when working with longer chromosomes leading to a finer covering of such a function, we can get into trouble when transitions only lightly affect the previous optimum's fitness. As we didn't use a gray code, hamming distance between consecutive optima won't be constant yielding to a non trivial problem for a genetic algorithm. This difficulty will range from 1 for simple transitions as $[00] \rightarrow [01]$ and $[10] \rightarrow [11]$, to 2 (i.e. $\lambda$) for the middle one ($[01] \rightarrow [10]$).

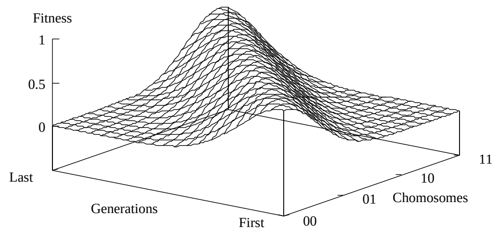


**Figure 2. Simple Dynamical Gaussian.**

This model is a simple one, but it introduces all relevant characteristics featured by Dynamical Environments and offers us an opportunity to study them in details.

### 2.2 Simulations

In the following, we present the results obtained by a simulation of the behavior of a DGA when facing the Minimal Dynamical Problem.

The simulation is based on equations ([11], [10]) describing the Dynamics of a GA but adapted to the dual approach. If we note $P_i$ as the proportion of individual $i$ at current generation, $q_i$ its proportion at next generation and, $x_i = \frac{F_i}{\overline{F}} P_i$ the effect for the selection mechanism, we obtain the following equational system:

$$\begin{aligned}
q_0 &= x_0 + p_c[x_3(x_1 + x_2 + x_3) + x_1(x_2 + x_3 + x_1) \\
&\quad - x_0(2x_2 + 2x_0 + x_1 + x_3)] \\
q_1 &= x_1 + p_c[x_2(x_0 + x_2 + x_3) + x_0(x_3 + x_2 + x_0) \\
&\quad - x_1(2x_1 + 2x_3 + x_2 + x_0)] \\
q_2 &= x_2 + p_c[x_1(x_0 + x_1 + x_3) + x_3(x_3 + x_0 + x_1) \\
&\quad - x_2(2x_2 + 2x_0 + x_3 + x_1)] \\
q_3 &= x_3 + p_c[x_0(x_0 + x_1 + x_2) + x_2(x_1 + x_0 + x_2) \\
&\quad - x_3(2x_1 + 2x_3 + x_0 + x_2)]
\end{aligned}$$

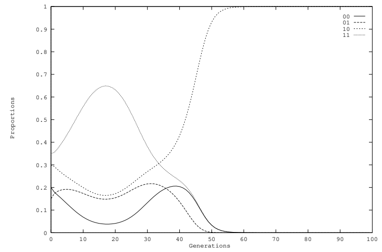


**Figure 3. Simulation of SGA.**

In fact we have redefined the reproduction operator to take into account the effects of the *mirroring operator* which allow the conservation of a certain redundancy because of the reproduction of dual chromosomes and so yields diversity in the population. Our new reproduction operator is : $x_i = (1-\tau) P_i^t \frac{F(i)}{\overline{F}} + \tau P_{\bar{i}}^t \frac{F(\bar{i})}{\overline{F}}$

Figures 3 and 4 present the proportions of each individual in the population over time. This criterion will allow us to observe that these proportions "follow" the successive optima. The DGA, contrary to the SGA, features an enhanced adaptiveness on this problem. According to the executable model, DGAs seem able to hang to each transition providing a new genetic exploration of the search space when needed. This exploration is enabled by the presence of complementary strings in the population (the dual chromosomes). When analyzing those results, we can observe that the proportion between dual chromosomes changes according to the transitions of the environment. DGAs feature, via crossover, a new way to introduce diversity in the population. It is easy to observe that crossover within a pair of dual chromosomes leads to a new pair of dual chromosomes but featuring a different expressed phenotype. Thus we can define an *Implicit Mutation Rate* quantifying the probability of performing such a crossover in the current generation : $imr(P) = p_c \cdot \Sigma_{\xi \in P \cap 0\Omega} \min(\nabla(\xi/P), \nabla(\overline{\xi}/P))$

Where $P$ stands for the current population, $p_c$ the crossover rate, and $\nabla(\xi/P)$ the number of occurrences

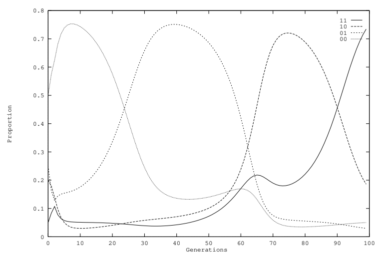

**Figure 4. Simulation of DGA.**

of the individual $\xi$ in this population. According to the dualization of the search space, we consider that the *mirror space*, $< \Omega >= \{0,1\} \times \Omega$, build over the basical space $\Omega$, is composed of two subspaces having respectively a meta bit at '0' or '1' and denoted by $0\Omega$ and $1\Omega$. The IMR rate evolves dynamically according to the ratio of each kind of dual chromosome and, during theses experiments, it has been observed to increase for each translations (generations 24, 62 and 90), thus providing an enhanced exploration of $\Omega$ whenever needed by the GA to fit its environment.

We have presented, in this section, a theoretical investigation of the DGA's ability to handle dynamical function optimization. In the following, we will extend those results to more realistic problems involving longer chromosomes and multidimensional functions.

## 3 Empirical confrontation

This section is aimed at presenting an empirical evaluation of DGA's effectiveness when confronted to more relevant dynamical environments based on 64 genes long chromosomes. In order to achieve this goal, we present the experimental conditions under which our evaluation was done. Then, we detail and explain empirical results describing the behavior of both DGA and SGA and finally, we underline the benefits of the dual approach in such a context.

### 3.1 Dynamical Gaussian

As in previous section, our model for investigating dynamical environments will rely on a simple Gaussian-like fitness function. From a static point of view this function is unimodal and non-deceptive. This Gaussian will translate over $\Omega$ in order to center its top on each consecutive individual of the search space.

We generalize this kind of function that will be formally defined over $n$ dimensions by :

$$\frac{1}{n}\sum_{i=1}^{n} e^{-\pi*\left(\frac{x_i}{2^{\lambda/2}}-t\right)^2}$$

In this formula, $t$ stands for the number of generations (time parameter), $\lambda$ represents the chromosome's length and $\pi$ a *Peakness* coefficient parameterizing the gaussian top's shape. This parameter has been introduced to avoid a "softer" top that induces small fitness differences between individuals belonging to the optimum's neighboring. With such a function, the GA will be allowed to keep the same individual as optimum during consecutive transitions without suffering from a relevant drawback as the fitness difference between optima is not significant. In this case our analysis criterion consisting in plotting, for each generation, the fitness of the best individual in population should not reflect the fact that the GA hanged to the new optimum. Thus, we decided to use a Gaussian featuring a sharper peak and so we used a relatively high peakness coefficient.

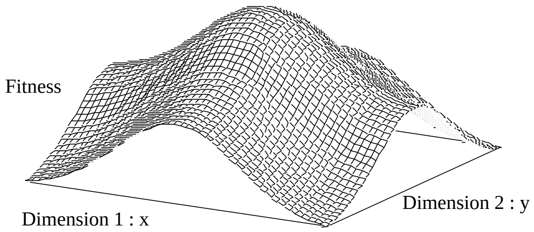


**Figure 5. General Shape for Dynamical Gaussian.**

Figure 5 illustrates our function at date $t$ in two dimensions. We chose to experiment with 8 dimensions each coded by a 8 genes long segment in the chromosome for a total of 64 genes. This choice has been dictated by an other problem that may occur when optimizing in changing environments. As working with binary coded integers, the GA may in case of a too high transition speed, only optimize the most significant genes according to the decoding algorithm. For example when handling 20 genes long chromosomes, the optimization of the first 5 bits induces a fitness above 95 % of the maximal one. This leads to final populations only consisting in partially adapted individuals. To avoid such a behavior, we have decided to use chromosomes long enough to implement a relevant search space but coding for different variables. In this context, a multidimensional function appeared to be a good choice. The transition will consist in a translation of the gaussian's top over $2^8$ individuals each composed of 8 identical sub-segments. For two dimensions, as shown in figure 5, this corresponds to simply following of $\Omega$ plan's diagonal. As expected, this motion will introduce variable hamming distances between two consecutive optima.

## 3.2 Experiments

In this section, we precisely define experimental conditions and present results obtained by SGAs and DGAs on this dynamical function. Finally, results are analyzed in order to underline DGA's specific capabilities in dynamical environments.

### 3.2.1 Experimental conditions

In order to reduce the influence of the initial population, each result presented hereafter stands for an average over 10 experiments. Each experiment has been realized using 64 bits length chromosomes encoding 8 integer values each 8 genes long. The function used is the Gaussian previously described, generalized at 8 dimensions. The Genetic Algorithm has been run over 5000 generations leading to a relevant translation speed. During this transition, each of the 8 chromosome's segments has taken an identical value upon the $2^8$ possible ones, according to an enumeration corresponding to a simple binary coding. For these experiments, the Peakness coefficient will be settled to $\pi = 3$ in order to avoid the "needle in the haystack" that would lead to a function too difficult to optimize whatever optimization method we choose.

All other "classical" parameters will be settled with traditional values: crossover rate at 70%, mutation rate at 0.1% and mirroring rate at 1%. The population will be sized to 100 individuals both for SGA and DGA. Results obtained with the latter should be interpreted as lower bound for its ability to deal with dynamical environments because the search space size has been increased by the use of meta genes. The crossover method is one point crossover and selection is assumed by a stochastic sampling method without elitism.

Concerning the meta-genes, we used one meta bit for each dimension. This choice has yield to multiply the search space dimension by a factor of $2^n$ where $n$ stands for the number of dimensions. Chromosomes used in the following experiments will then be 72 genes long for the DGA and 64 gene long for the SGA. Each meta gene will be located at the beginning of the segment coding for its associated dimension variable.

This choice has allowed us to generalize the concept of DGA, which has been previously defined in a unidimensional context, to a multidimensional environment.

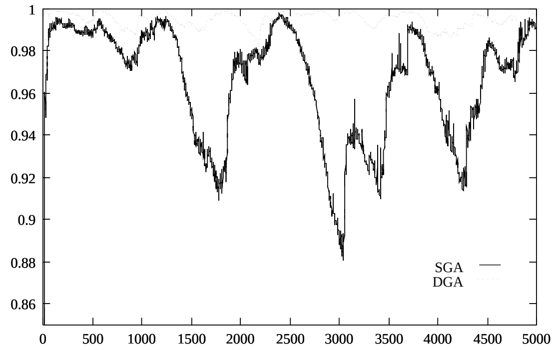


**Figure 6. Best Fitness in generation versus Time.**

It is worth to notice that following results has been obtained without increasing population size for DGA thus representing a lower bound for their abilities.

### 3.2.2 Empirical Results

Figure 6 plots the fitness of the best individual for each generation. This is a relevant criterion for analysis according to the fact that our function features consecutive optima having the same fitness (1.0). An ideal plot, for it, consists in an horizontal line at ordinate 1.0. This should involve that the GA has found the optima at each generation, thus providing an exeptionnal adaptiveness. Experimentaly, plotting this criterion allows us to observe difficulties encountered by the GA during its adaptation. The figure clearly shows that using a DGA leads to better results than those obtained with a SGA. The average value, over all generations, of the best fitness value thus giving us a numerical measure of this advantage (0.965035 for SGA versus 0.994264 for DGA).

This clearly advocates that the DGA keeps nearer of the consecutive optima than the SGA thus featuring an improved adaptation over dynamical changes to the fitness function shape. If we look at the plotting, we can see that the difficulties encountered by both GAs are more important at regular locations. In fact, the hardness seems proportional to the hamming distance between two consecutive optima. For example, the SGA drops off mainly at the middle and quarter transitions precisely where hamming distances are respectively of $\lambda$ and $\lambda/2$.

For the DGA the situation is almost inverse; during the more difficult transition, the DGA keeps hanging on thus offering a more adaptive behavior. And what's more, this property seems proportional to difficulties encountered. Furthermore, the DGA keeps very close to the SGA for easier transitions. Results presented are thus encouraging for the dual approach as this one revealed to increase performances proportionally to transition's difficulty. Now, we will analyze this fundamental property.

### 3.2.3 Discussion of results

All previously presented results can be entirely explained by the intrinsic nature of dual genetic algorithms. The transition difficulty is directly proportional to the hamming distance between two consecutive optima. This difficulty measure was only effective according to the hypothesis that the GA's search space exploration was only relying on the concept of hamming distance. In the case of a DGA, there exists an other metric that can be considered during the search process. By its ability to deal with complementary chromosomes (dual chromosomes), the DGA is able to keep, even while converging from a phenotypical point of view, a genotypical diversity by the coexistence of both dual chromosomes which are decoding via transliteration to the same current solution.

Let us consider the hardest transition for a SGA; it only consists of taking the new optimum as a chromosome separated by a $\lambda$ hamming distance from the previous one (refer to figure 7. This clearly involves to take the complementary string. In a DGA the genetic material needed to obtain such a string by application of crossover operator only is directly available, thus the distance between the two optima is one (mutation of the meta bit in one of the dual chromosomes for example). In fact, we can call this distance, defined over $\Omega$, a *dual distance* that shows how a DGA can reach any point of the search space by taking the complementary string of its current location in a single step. If $\delta_H(X,Y)$ stands for the hamming distance, the dual distance is : $\delta_D(X,Y) = \delta_H(\overline{X},Y)$

In the case of our sub-transitions, the DGA features results similar to the SGA's ones. Our previous arguments have advocated for the capacity of a DGA to use the dual distance. But, when facing less important transitions in terms of hamming distances between consecutive optima, the DGA seems to feature results similar to SGA's ones. This experimental observation leads us to consider the conjecture that a DGA can dynamically select which distance is relevant during its search. While exploring the search space, the DGA is supposed to use one of the previously defined distances according to its needs. We can call this an *optimal distance* according to the exploration process and write it $\delta_{Opt}(X,Y) = Min(\delta_H(X,Y), \delta_D(X,Y))$

The figures 7 and 8 summarize all this stuff according to Hamming distance for SGA and to the optimal distance for DGA. This enables us to see that the GA

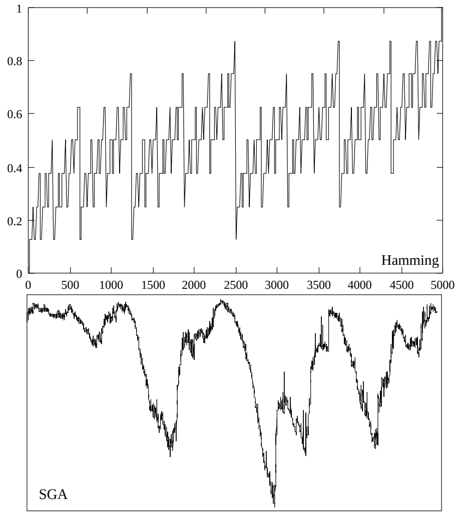


**Figure 7. Distances between two consecutive optima (SGA).**

generally hangs out when there is a recess in the distance function. The hamming distance is sufficient for explaining the SGA's behavior. For the DGA, we have plotted Optimal distance which is pertinent to explain this new dynamics.

This property of dynamically choosing a good metric over the search space $\Omega$ explains the behavior featured during experiments and the results presented in the previous section. This also explains the superiority of the dual approach on this new field and opens to DGA perspectives closed to SGA. Choosing dynamically a metric on their search space appears to be one of the fundamental DGA's properties. Furthermore, this represents a new step toward the design of adaptive autonomous entities and its study may be a relevant part of the future work concerning the dual approach.

## 4 Discussion and further work

Dynamical environments constitute an important challenge for any optimization method devoted to real time applications and moreover to all optimization methods needing to keep their adaptiveness after reaching a potential solution. In this paper, we have proposed a simple model of such an environment and underlined its main characteristics. This part of our work should be extended with other models in order to constitute a basis for a complete testbed devoted to dynamical environments. The second part of our work

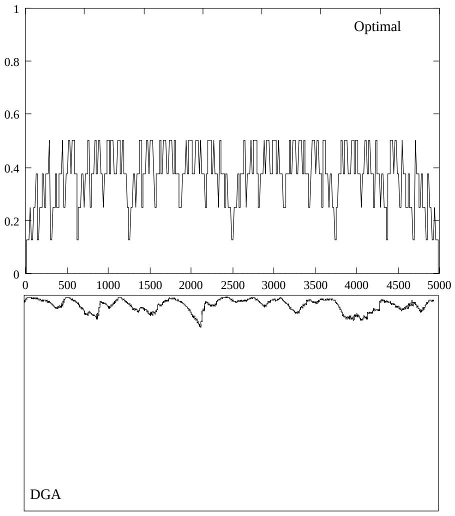


**Figure 8. Distances between two consecutive optima (DGA).**

has consisted in comparing both SGA and DGA effectiveness on this problem considering consecutively their theoretical behavior on a minimal model and then their real behavior on a 64 genes long chromosomes problem.

The empirical study has confirmed theoretical simulations; we have thus provided two criterion analysis for observing the ability of a GA to deal with dynamical environments. The first one consists in following the proportions of each individual to observe the consecutive convergences. It is particularly well suited for small problems or simulations. For the second, we plot the best fitness for each generation giving us an approximation on how close to the optima the GA keeps according to the fact that we know the values for the optimum's fitness for each generation. Over experiments, the DGA has revealed to be better than the SGA, this effectiveness has been shown to come directly from their ability to deal with transitions featuring high hamming distances between two consecutive optima. This measure has also revealed to be a relevant measure of difficulty for these environments and suggests further work to clearly establish its usefulness to evaluate function difficulty from a dynamic point of view. The introduction of a transliteration phase has explained DGA's enhanced adaptiveness, allowing them to handle dual chromosomes and thus converging to an uniform population (from a phenotypical point of view) while keeping various genotypes. This diversity is maintained by the application of the mirroring operator which enforces the preservation of similar proportions of each member of a dual chromosomes pair. Such a behavior leads us to observe that the mirroring operator builds symmetry (thus yielding to mirror populations), while the natural dynamics of the GA breaks these attractors in the populations space. Both antagonist dynamics are present but acting at different levels; diversity for genotypes handled and uniformity for the phenotypes expressed. This property seems to be the key of an efficient dynamical function optimization.

Beyond this problem, we can notice that preserving diversity is also a promising gap for solving numerous problems GAs are confronted with. Thus, our work should be extended to investigate methods for preserving genotypical diversity and studying their effectiveness not only on dynamical functions but also on deceptive problems for example. We should also notice that they should cover domains like multicriterion optimization, multimodal optimization where diversity is the *sine qua non* condition for finding out a majority of niches for example. This work underlines a fundamental problem for GAs and investigating how to build GA reacting faster to their environment yields to increasing their main characteristic : adaptiveness.

## References

[1] P. Collard and J. Aurand. Dga: An efficient genetic algorithm. *ECAI'94: European Conference on Artificial Intelligence*, 1994.

[2] P. Collard and C. Escazut. Genetic operators in a dual genetic algorithm. In *Tai'95*, 1995.

[3] P. Collard and C. Escazut. Relational schemata: A way to improve expressiveness of classifiers. *ICGA'95: Proceedings of the sixth International Conference on Genetic Algorihtms*, pages 397–404, 1995.

[4] D. Goldberg. *Genetic Learning in optimization, search and machine learning*. Addisson Wesley, 1994.

[5] D. Goldberg and R. Smith. Nonstationnary function optimization using genetic algorithms with dominance and diploidy. In *ICGA-2*, 1987.

[6] D. Golderg. Diploidy and dominance in artificial genetic search. *Complex Systems*, 6(3):251–285, 1992.

[7] J. Grefenstette. Genetic algorithms for changing environments. In R. M. abd B. Manderick, editor, *Parallel Problem Solving from Nature 2*. Elsevier Science Publishers B.V., 1992.

[8] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 2nd edition, 1992.

[9] S. Kauffman. *The Origins of Order : Self-Organisation and Selection in Evolution*. Oxford University Press, 1993.

[10] M. Vose. Modeling simple genetic algorithms. *Fundations of Genetic Algorithms*, 1991.

[11] D. Whitley. An executable model of a simple genetic algorithm. *Foundations of Genetic Algorithms*, 1991.

[12] D. Whitley. A genetic algorithm tutorial. Technical Report CS-93-103, University of Colorado, 1993.