# Time Dependent Optimization with a Folding Genetic Algorithm

**Alessio Gaspar** **Philippe Collard**
University of Nice Sophia-Antipolis, I3S Laboratory — CNRS
250 av. A. Einstein, Sophia Antipolis, 06560 Valbonne FRANCE
Fax : (33-1) 04-92-94-28-98
e-mail : {gaspar, pc}@alto.unice.fr
Web : http://wwwi3s.unice.fr/mage.html

**Abstract**

*Time dependent optimization has revealed to be a promising gap for the entire Genetic Algorithms community since it has numerous applications. This paper extends previous work related to the use of metagenes ([5]) in the so-called Dual Genetic Algorithms (DGAs). A more generic framework, involving a variable number of genes, is introduced. Folding Genetic Algorithms are thus proposed as a new class of genetic algorithms which effectiveness is investigated on two well known models of dynamical environments and compared to Simple Genetic Algorithms and DGAs.*

*Eventually, further analysis of these results enlightens the ability of FGAs to evolve a metrics over the search space (i.e. a kind of encoding scheme) along with potential solutions. These particularly encouraging results open us interesting perspectives as FGAs should be applied to to other fundamental problems investigated by the GA community in order to measure the benefits of this really meta level of evolution.*

## 1 Evolutionist Approach

This paper considers time dependent optimization by Genetic Algorithms. At first, we precisely define what both of these terms correspond to providing a characterization of dynamical environments difficulty. We also link this particular issue to most fundamental ones addressed by the Evolutionary Community.

### 1.1 Genetic Algorithms

Genetic Algorithms ([9, 11]) have been introduced in the sixties by professor John Holland at university of Michigan as models of an Artificial Evolution. In the thirty past years, they have been successfully applied to a wide range of problems such as Natural Systems Modeling (e.g. Artificial Life environments [11], immune system modeling [7]), Machine Learning by the way of classifier systems, and optimization.

GAs basically handle a population of chromosomes (Individuals) often modeled by $\lambda$ long vector of binary genes. Each one encodes a potential solution to the considered problem and is rewarded by a so-called *Fitness value* which is directly correlated to how good it is to solve the problem. Starting with a random initial population, we iteratively apply a so-called *Genetic Transformation* which consists of three genetic-like operators such as *crossover*, *mutation* and *selection*each having the following specific role ;

- *Selection* exponentially spreads, over generations, above average fitted chromosomes by allowing them more offsprings in next generation.
- *Crossover* exploits genetic material of the current populations by breeding most interesting strings.
- *Mutation* introduces diversity in population by flipping randomly chosen genes.

As a result, this overall non linear dynamics leads to an homogeneous final population only featuring instances of the best chromosome found so far.

Simple Genetic Algorithms (SGA) have shown a robustness in fitness landscapes featuring local optima, being able most of the time to find the global optimum. Nevertheless, *deceptive functions* or *Dynamical* ones revealed theirs limitations.

### 1.2 On Dynamical Environments

Fitness values changing over time define what we should call *Fitness Dynamics*. From this generic definition, we can distinguish between fitness values changing due to the Genetic Algorithm own dynamics (endogenous causes) or due to environmental causes (exogenous). The first case concerns average schemata values during hyperplane sampling as well as chromosomes fitnesses when using *scaling* or *sharing* methods.

The second defines *Dynamical Environments* that is functions fitness changing over time.

In both cases, the Genetic Algorithms are expected to be able to track the optimum over time after it has changed during a *Transition* separating two *Periods* of static behavior. By addressing the time dependent optimization issues, one obviously also addresses most of intrinsic weaknesses of Genetic Algorithms. Clearly, adapting a GA to such a context is nothing else than improving its exploration-exploitation tradeoff in order to enhance *diversity* management. Indeed, in natural as artificial ecosystems, *Polymorphic Populations* are a key to reactiveness as well as robustness to perturbations for species. In a Genetic search, maintaining diversity is central in order to be able to react to sudden environmental change as well as to escape local misleading optima leading to premature convergences.

This paper will focus on time dependent optimization in order to address both classes of issues. Without entering into details, a dynamical environment, at least in the unimodal case, may be fully characterized by its transitions parameters. Among all parameters that can be measured, this paper will focus on the *Transition Distance* ($\delta_d$) and the *Period.* The first measure is defined as the hamming distance separating two consecutive optima in an unimodal context. The second measure, noticed $\delta_p$, stands for the length, expressed in generations, of periods during which fitness is stationary (*Transition Speed*).

Both parameters have already been proved to be significative of a dynamical environment's difficulties.

### 1.3 The Pattern-Tracking Model

The first model of Dynamical Environments consists in experimenting with fixed values for $\delta_d$ and $\delta_p$. The fitness value $\mathcal{F}_x$ is the hamming distance between $x$ and an arbitrary chosen optimum. At each transition, optimum is changed to a random one located at distance $\delta_d$ from it. Such a model has been first experimented by Pettit and Swigger ([15]) and then Vavak and Fogarty ([17]). We will work on 40 genes long chromosomes for 250 generations with $\delta_s = 50$ and 3 different values for $\delta_d$ (5, 20, 35). The second model we used is more complex as it features a greater diversity of transitions within the same experiment.

### 1.4 The MD-Gaussian Model

This Dynamical Environment relies on a multi-dimensional gaussian-like function defined by :

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

Where $t$ locates the optimum, $\lambda$ represents the chromosome's length and $\pi$ a *Peakness* coefficient parameterizing the gaussian top's shape (here $\pi = 3$). By increasing $t$ over time the gaussian slides on the search space's diagonal, each of its $n$ variables changing uniformly. We are going to encode 8 dimensions variables each 8 genes long. As a result, this 64 gene long chromosome will take $2^8$ different values during the experiment. Each run will be 5000 generations long, this leads to a transition speed $\delta_s = 5000/2^8 = 19.53$. We use a binary coding to enumerate the search space, sliding on it will lead to every possible $\delta_d$ ranging from 1 when passing from $< 00000000 >$ to $< 00000001 >$ for example, to $\lambda$ for the *middle transition* between $< 01111111 >$ and $< 10000000 >$. It's worth noticing that, in the case of DGAs and FGAs, population size was not increased despite of the growth of the search space[1]. Following results may thus be interpreted as a lower bound for the abilities of these meta approaches.

Moreover, a Multidimensional function avoid the Genetic Algorithm to only optimize most significant genes when processing long chromosomes with binary encoding (i.e leftmost ones traditionally). When considering Dynamical Environments, this would leads us to ignore transitions affecting low weighted genes.

### 1.5 GAs in dynamical environments

Our previous definition of a Simple Genetic Algorithm straightforwardly implies that it is not able, once having converged to a uniform population, to converge to a new optimum if fitness function changes. This is due to the fact that crossover and selection can not provide new chromosomes when applied to a uniform population. The SGA thus only relies on mutation to provide the required diversity to converge toward the new optimum. The issue thus becomes how to adapt GAs for time dependent optimization ?

A brief state of the art of Genetic approaches in this area enlightens some relevant entry points of the evolutionist paradigmto enhance significantly GA's adaptiveness to environmental changes. For instance, we classify them in 3 main classes;

- *Operational Diversity* i.e. adapting the mutation rate by increasing it when a change is detected in the fitness landscape ([10, 1]) or by making it emerge by the use of an implicit mutation rate (see Dual Genetic Algorithms below for an illustration of this principle).
- *Fitness Evaluation* : By changing the way to evaluate individuals by the use of sharing methods for example, polymorphic population can emerge from the GA dynamics. Therefore diversity can

[1] which is doubled for the DGAs and multiplied by a factor of $2^{\lambda}$ for the FGAs.

be enhanced by this method providing relevant enhancement of the diversity management.

- *Using Introns* : under this generic term we group all attempts to use genes non directly coding for a given solution. These genes are either ignored ([12, 13]) or used to influence other ones during a *transcription* like process.

Now that we have defined what is the problem we want to address, we describe, next section, the several steps leading to the approach we propose to address the intrinsic limit of SGAs through the third entry point previously presented.

## 2 Folding Genetic Algorithms

At first, we summarize previous work on the introduction of a meta-gene in chromosomes. Next, we extend this primary model and generalize it to the use of an arbitrary number of meta-genes. In order to valid this approach we eventually lead an empirical study on dynamical environments.

### 2.1 Dualism as a first step

*Dual Genetic Algorithms* (DGA) ([2, 6]) have been proved to be particularly efficient genetic algorithms which offer an improved robustness against most intrinsic weaknesses of SGA. DGAs work on chromosomes with a head-bit *meta-gene* in front of them. The search space $\Omega$ is thus doubled as the DGA considers $\lambda+1$ gene long chromosomes. So,this new search space $< \Omega >$ contains all possible $\lambda + 1$ gene long strings (*genotypes*). As a result, the basic space $\Omega$ contains *phenotypes*, i.e. normal $\lambda$ genes long chromosomes directly evaluated by user defined fitness function.

Thus, in order to evaluate a genotype, the DGA to maps it back to the phenotypic space through a so-called *Transcription* step[2] which is stated as follow;

$$\begin{array}{rcl} T: \; <\Omega> & \rightarrow & \Omega \\ T(\dot{0}\omega) & = & \omega \\ T(\dot{1}\omega) & = & \overline{\omega} \end{array}$$

In other words, if the meta-gene is active ($\dot{1}$) the phenotype is taken as the complement of the following $\lambda$ genes of the genotype. Otherwise, the phenotype is the remaining genes themselves. This mechanism clearly leads complementary genotypes to encode for similar phenotypes, $< \Omega >$ wcan be therefore viewed as a doubled, symmetrically unfolded version of $\Omega$. For instance, the phenotype $[011] \in \Omega$; can be transcripted from both $[\dot{0}\ 011]$ or $[\dot{1}\ 100] \in < \Omega >$.

[2]In analogy to tnatural genetics mRNA transcription phase.

Previous work ([6]) enlightened the ability of DGAs to take advantage of these two different levels (genotype and phenotype). A so-called *mirrorring operator* has been used. It stochastically maps, with rate $\tau$, a chromosome from $< \Omega >$ into its dual counterpart by complementing it. This results in a genotypical diversity not affecting phenotypical uniformity.

In other words, DGAs can converge to a single phenotype while keeping in the current population complementary genotypes. Indeed, it is also worth noticing that both offsprings resulting from a crossover between two complementary genotypes (*dual pair*) will be complementary too. But, they will encode a different phenotype than their parents. So, by maintaining with mirrorring a minimal amount of dual pairs in a population, even when having converged, we offer to DGAs an opportunity to perform such crossovers and thus to introduce diversity without any explicit mutation or randomness in order to enhance the capability to react to its environment.

This property is central to explain why, despite of the growth of the size of the search space, DGAs are efficient Genetic Algorithms that have already been successfully applied to a wide range of areas including optimization [2], machine learning [3], Genetic Programming [16] and Dynamical Environments [5].

The Following section introduces *Folding Genetic Algorithms* as a new class of Genetic Algorithms extending and generalizing the use of meta-genes.

### 2.2 Multiple Meta-Genes

In comparison to the previous approach, chromosomes handled by FGAs are now composed of $\lambda$ *exons* (i.e. genes coding for a chromosome in $\Omega$), and of $\dot{\lambda} \in [0..\lambda]$ meta-genes (*Introns*) at least separated by one exon. Each meta-gene will have influence over all exons following it until the next intron. In order to illustrate this structure, let us take some examples of transcriptions over a $\lambda = 3$ basic search space;

$$\begin{array}{rcl} T([\dot{1}01\dot{0}0]) & = & [100] \\ T([10\dot{1}1]) & = & [100] \\ T([\dot{1}\dot{0}0\dot{0}0]) & = & [100] \end{array}$$

This mechanism clearly generalize DGA's ability to symmetrically unfold the basic space $\Omega$ by increasing the number of meta-genes used.

### 2.3 Partial Mirroring Operator

As Mirroring introduces diversity at the genotypic level while preserving the phenotypic homogeneity of the DGAs populations, FGAs use a *Partial Mirroring Operator* which complement, with rate $\tau_P = 0.01$, each meta-gene and all exons influenced by it.

By introducing *clones*[3] in the population *Partial Mirroring* is functionally equivalent to Mirroring in DGAs. Contrary to DGA, the number of clones potentially available is far higher ($\sum_{i=1}^{\lambda} 2^i = (2^{n+1} - 2)$ vs 2 for a DGA). This leads to an improvement of the diversity potentially available in the population.

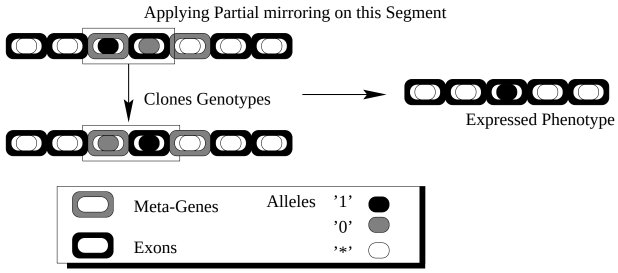


Figure 1: Partial Mirroring Operator.

## 2.4 Folding the search Space

The *Folding Operator* operates on exons with rate $\Phi$ and acts differently according to the presence of a meta-gene on the previous locus.

- If the previous gene is an exon, Folding inserts a meta-gene thus partially unfolding $\Omega$.
- If previous gene is a meta-gene, Folding deletes it and thus partially folds the search space.

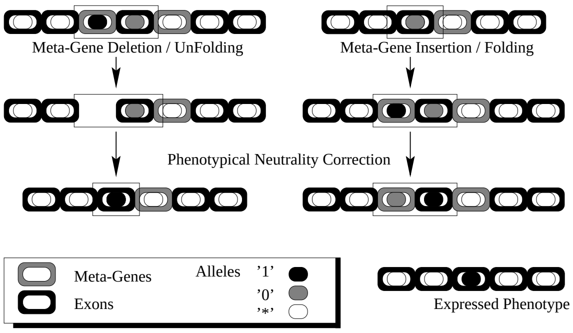


Figure 2: Folding with Active Meta-genes

If folding or unfolding concerns an active meta-gene, the phenotype is altered. Therefore, in such cases, we stochastically complement the concerned exons in order to keep this genetic operator *phenotypically neutral*. This repair process is applied with rate $\Phi_n$, $n$ standing for neutrality. Thus, it allows the exploration of all possible locations for meta-genes over chromosome. Figure 2 illustrates manipulation of an active meta-gene in both folding and unfolding cases.

We now test this new approach on two different models of dynamical environments previously defined.

[3] i.e. different genotypes encoding a similar phenotype

# 3 Empirical Study

In this section, results obtained on both environments are presented and analyzed.

## 3.1 Experimental Parameters

The experimental conditions are the following; average on 10 runs with different seed values, mutation rate $\mu = 0.001$, crossover rate at $\mathcal{X}_c = 0.7$, population size is 100. We run the genetic algorithm each time for 5000 generations on MD-Gaussian and only 250 on Pattern Tracking as transitions are all identical. Tracked optimum is at respective constant fitness value 1.0 and 40 for gaussian and pattern tracking environments. Eventually, we used one point crossover and selection is assumed by a stochastic sampling method without elitism. FGA's specific parameters were tuned to respectively $\Phi_n = 0.7$ and $\tau_P = \Phi = 0.001$ in both dynamical environments. We introduced DGA-n, as a method featuring a fixed amount $n$ of meta-genes (a hand designed meta-mask) dispatched over the chromosome. We conjecture that such fixed meta-mask may give particularly good results when especially designed for a given environment. In following experiments, we chose to use one meta-gene by function dimension in Md-gaussians. Mirroring rate was $\tau = 0.01$ for both kind of DGAs.

## 3.2 Pattern-Tracking Experiments

First Results obtained on this simple model are illustrated in figure 3 which plots best Fitness vs generations. This kind of analysis criterion involves an expected best fitness of $\lambda = 40.0$ corresponding to a Genetic Algorithm having tracked the optimum all the time. On the plottings, such a behavior would correspond to an horizontal line. Instead of this, one can see (fig. 3) that, at each transition, the GA may unhook even if its convergence to the optimum is quick because of the GA-easiness of this function within periods. Results clearly show;

- For $\delta_d = 5$ (Fig 3.A) :
  Problem is too easy, little transitions does not allow to distinguish between all four GAs.
- For $\delta_d = 20$ (Fig 3.B) :
  For reasons detailed latter and in [5], this was the worst case for DGAs even if they nevertheless performed better comparing to a SGA.

- For $\delta_d = 35$ (fig 3.C) :
  Here DGA revealed to be better than any other GAs as they naturally handle dual structures.

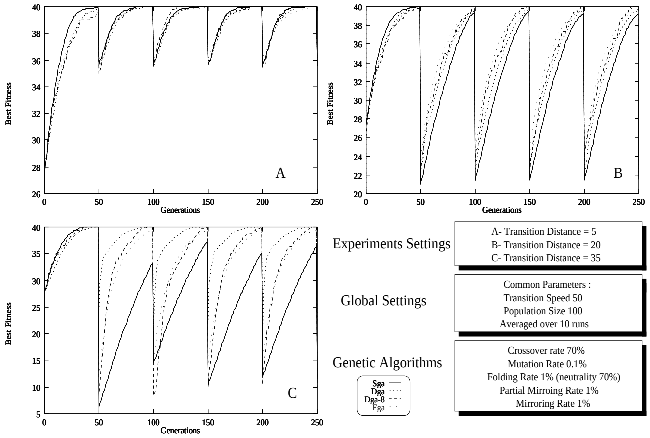


Figure 3: Pattern Tracking Results

In all cases, FGA revealed to keep very close to DGA results and even overtook it in some case despite of the considerably increased difficulty (especially in terms of search space growth). In a real complex environment, transitions are about to be somewhat unpredictable and irregular. According to these experiments FGAs seem more able to adapt to a wider range of perturbations without significant lost in their overall efficiency comparing to DGAs. For this reason, They seem more about to be applied to arbitrary unknown environments as featuring an enhanced adaptiveness. These results must nevertheless be completed in order to conclude on FGA efficiency as we have chosen to keep the population size identical. The following MD-Gaussian model features more complex transitions which difficulty changes over time. Such an environment is thus more realistic as we also decided to make continuous alterations of the fitness landscape.

### 3.3 MD-Gaussian Model Experiments

Results are presented in figure 4; the curve plots best fitness versus time and we detailed overall averaged values of both best and average fitness. Concerning the Average Best fitness, the ideal behavior is not straightforward. It may consist in a quasi perfect convergence characterized by an average value as close as possible to the best one. Or, in the other hand, as we have to keep diversity in population, the difference between best and average fitness should be a proper indicator of the phenotypical diversity quality. Even if, when using meta-genes, genotypical diversity is sufficient, this would nevertheless indicate that the system features an improved reactiveness against transitions through *Polymorphic Populations*.

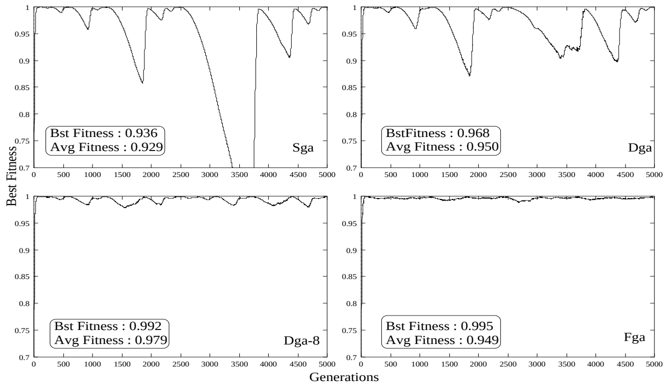


Figure 4: MD-Gaussian Results.

Regarding to these criteria, we can say that FGAs have revealed to be the most efficient approach both in terms of overall best fitness value and unhooking phenomena in the plottings. DGA already performed better than SGA as analyzed in previous work ([5]), this experiment enlighten the advantage of locating several meta-genes in the chromosome, and to use a method to evolve their number and location automatically. DGA-n were settled for this environment with one meta-genes per dimension, indeed FGAs revealed to be able to design (evolve) even a more efficient meta-mask. Moreover, we may also notice that they managed to increase the difference between their average best and average fitness. As the function is easy to optimize during periods (i.e. from a static point of view), average and best fitness values are very close in all cases, characterizing a particularly good convergence. With FGA, we find the larger difference among the four methods tested. This is interesting as it shows that FGAs easily feature diversity, even at a phenotypical level, without significant lost of effectiveness. This diversity plays a central role during transitions.

FGAs effectiveness is more than encouraging; as they achieved the design of both solution and coding while being more efficient than traditional approaches. We are thus facing a meta-system having itself chosen which meta-mask to use in order to optimize its own effectiveness. This characterizes, from our own point of view, the *Artificial Life approach* which tends to realize the transition between systems externally controlled or heavily parameterized to fit a particular task and *adaptive*, *autonomous* ones behaving well without any expertise on their working (black box).

Comparing to the work realized by FGAs, DGA-8's effectiveness is very poor and does not justify in any way their need of external parameters as specific as a meta-mask.

# 4 Explaining FGA's efficiency

This section explains previous experimental results by underlining FGA's ability to evolve metrics over the search space. At first we present the link between meta-genes and metrics and then consider cases study enlightening advantage of introducing several meta-genes (DGA-n). We eventually discuss in details why FGA is more efficient than other approaches on the basis of the different metrics it can feature.

## 4.1 Metrics and Search Spaces

What is a metrics over a search space ? Basically it's a distance relation between two points. In Simple Genetic Algorithms, mutation on one gene represents the canonical move in the search space. It is strongly correlated to the hamming distance that is the number of genes different between two chromosomes. In the context of DGAs, we have defined in previous work ([4]) *Dual Distance* between two points of $< \Omega >$. This measure takes into account the possibility to flip the meta-gene instead of exons. By doing so, one can "jump" of $\lambda$ bits at the cost of only one flip. The distance between two chromosomes can thus be measured by either the hamming distance ($\delta_h$) or its complementary *Dual Distance* ($\delta_{\overline{h}} = \lambda - \delta + 1$). Moreover, DGAs have been conjectured to be able to dynamically choose, according to a fitness criterion, the most interesting distance for exploring the search space ([4]). Here we simply define the *Effective Distance*[4] as an optimal choice between the two metrics available i.e $\delta_{Dga} = Min(\delta_h, \delta_{\overline{h}})$.

## 4.2 Case Study with DGA-n

Before to provide an analysis of the effective distance featured by FGAs (noticed $\delta_{Fga}$), in order to fully explain obtained empirical results, we first enlighten the advantages of using several meta-genes with DGA-n. For display convenience, we restrict to examples involving small number of genes.

Our first example consists in 6 genes long chromosomes. We consider, for all GAs investigated (SGA, DGA and DGA-2[5]), the *Effective Distance*. This distance is defined as the best distance we can feature between two consecutive chromosomes in a binary enumeration of the search space similar to the one provided by multi-dimensional gaussians (see fig. 5). It

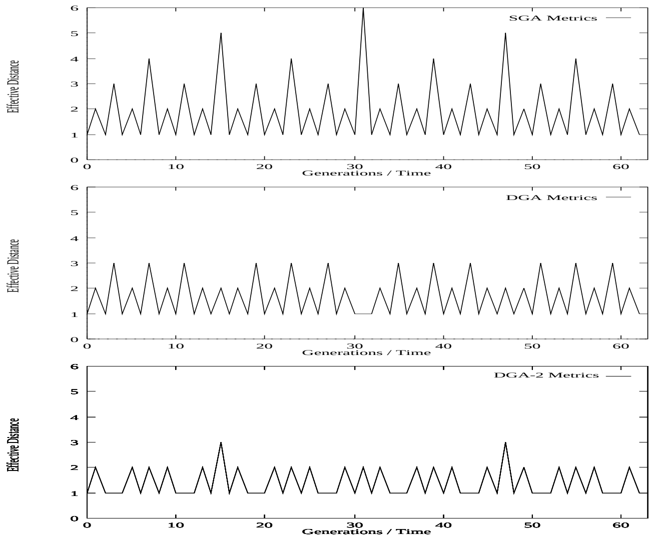


Figure 5: Metrics during Transitions.

is obvious, on this example, that increasing the number of meta-genes leads to a more convenient metrics over the search space as it allows to reduce the effective distance between two successive optima. Because they feature diversified values for $\delta_d$, multi-dimensional gaussians and more generally such enumerations are good candidates to evaluate the adaptiveness of a particular GA.

Nevertheless, in order to precisely quantify the effects of a particular metrics, we may average the Effective Distance value over *all* possible transitions and not only consider the ones resulting from a binary enumeration. If previous method allows to feature all possible values for $\delta$, it lack in considering every possible transitions (i.e. pairs of chromosomes). We now consider, for all possible hamming distances between two arbitrary chromosomes, all locations of the genes different from one chromosome to the other. By doing so, we consider all cases and thus provide more reliable and relevant results, especially for FGAs which, contrary to SGAs and DGAs, are able to feature different optimal distances for a similar value of $\delta_d$ according to differing genes location relatively to meta-genes. FGAs thus provide a more accurate metrics over the search space taking into account differences ignored by previous models.

We consider a second simple problem featuring 12 genes long chromosomes. We average the *Effective Distance* over all possible pairs of chromosomes for SGAs ($\delta_{Sga} = 6.001$), DGAs ($\delta_{Dga} = 5.03$) and DGA-n (for respective values $n = 1, 2, 3, 4, 6, 12$ we have

[4] The effective distance, for a SGA, is the only one it can use i.e. the hamming one. DGAs can use both Hamming or Dual ones.

[5] Which replace DGA-8 by considering 2 segments each 3 genes long.

$\delta_{Dga-n} = 6.001, 4.501, 5.00, 4.68, 4.81, 5.03)$.

The later being evaluated with various number of meta-genes ($\dot{n}$). In all cases, DGA-n feature better average distances than SGAs and even DGAs. All this clearly advocates the hypothesis that the metrics used is strongly correlated with the easiness featured by the Genetic Algorithm to adapt to successive transitions. It is thus relevant to analyze which kind of metrics do FGAs induce over the search spaces.

### 4.3 Evolved Metrics

Previous section enlightened the advantage of multiples meta-genes as well as the relevance of their positions in chromosome. FGAs addresses both issues by evolving the meta-genes and consequently the metrics being in use for this individual. In order to characterize the metrics induced by a FGA, we may consider the fact that it emulates any of the previous metrics.

The optimal metrics $\delta_{Fga}$ is then taken as the minima among the ones associated to each of the $2^\lambda$ meta-mask available. From an informal point of view, $\delta_{Fga}$ can also be viewed as the number of isolated segments of consecutive alleles '1' in the x-or vector between the two considered chromosomes. This comes from the fact that, for each segment representing an isolated region of adjacent differing genes between two chromosomes, we can delimit it by two meta-genes and then get its bitwise complement in one step only (meta-gene flipping). This clearly confirms and explains how that FGAs take into account, beyond the distance, the location of differing genes between the two considered chromosomes.

As a result, we can no more consider the plotting of their effective distance according to $\delta_d$ only. We may rather, for each value of $\delta_h$, average the effective distance value $\delta_{Fga}$ over all possible pairs of chromosomes located at this hamming distance from each other. Figure 6 shows this average value compared to $\delta_{Sga}$ and $\delta_{Dga}$, for a 20 genes long chromosomes. This clearly enlightens the average quantitative improvement. The average has been done by choosing, for each pair of chromosomes being distant of $\delta_d$ genes, the metrics providing best results as we did for all GAs. This leads us to consider the qualitative improvement provided, not at the previous global and averaged level, but rather for each possible distance $\delta$. More formally, let us note $\#^\delta_n$ the number of chromosomes being $\delta$ genes different and having these different genes located such as they form $n$ isolated (i.e. separated by at least one gene identical in both chromosomes) segments on the $\lambda$ long chromosome. Informally, this corresponds to the number of pairs of chromosomes over the search space having $\delta_{Fga} = n$.

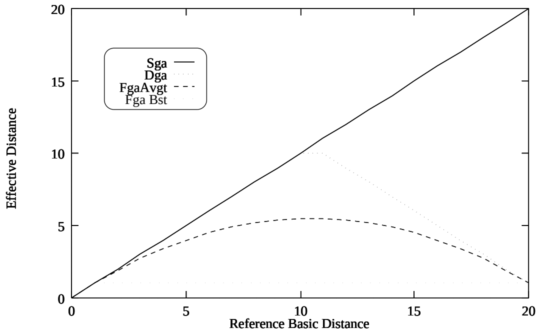


Figure 6: Effectives Distances.

The number of isolated segments of adjacent differing genes being an informal expression of the Effective $\delta_{Fga}$.

Beyond the comparison with other effective distances, we now quantify the qualitative improvement provided by FGAs; Figure 7 plots the density of cases where ($\delta_{Fga} < \delta_{Dga}$) vs Effective Distances featured by the FGAs over this Search Space. In other words, this plotting considers for each value of $\delta$ ($x$ axis on the curve) ranging from 0 to $\lambda$ ($\lambda = 20$ in our example), the number of chromosomes pairs featuring $\delta_{Fga} < \delta_{Dga}$ among the $C^\lambda_\delta$ possible ones which have $\delta$ distant chromosomes. This density is evaluated given $\lambda$ and $\delta$ and represents in fact the density of cases, given any pair of arbitrary chromosomes, where FGAs lead to an improvement via a metrics comparing to DGAs. This density is computed by summing $\#^\delta_i$ whereas $\delta_{Fga} \geq \delta_{Dga}$ for all number of isolated segments of adjacent differing genes that can be formed.

We can see clearly that, even if extrema distances (near 0 or $\lambda$) are moderately advantageous, FGAs are outstanding in most of cases at a rate close to 100 percent. Hence, there remains very little probability to

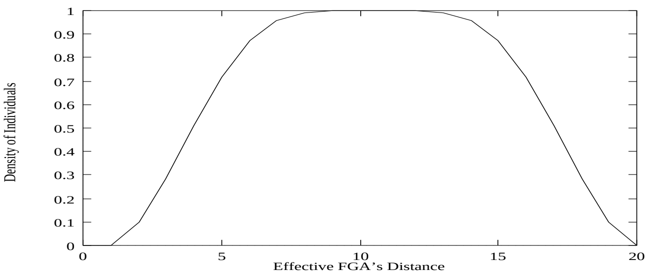


Figure 7: Density of Distances for fixed distance $\delta$.

face worst cases. This leads to a qualitative improvement (and moreover adaptation) of the metrics over

$\Omega$. This may be particularly relevant against encoding dependency related issues which are central to the Genetic Algorithms community.

Moreover it is worth noticing that altering metrics can improve the adaptiveness without large amounts of diversity in the populations. Individuals having good meta-masks are just potentially able to move quicker than other to interesting areas of the search space. At a low cost in terms of distance, meta-masks can allow them to reach any other area quicker. When evolving a good meta-mask they thus become able to have in their close neighboring, particularly fitted areas of the search space. In opposition to explicit diversity given by genotypes, we will refer to this potential diversity corresponding to potential quick moves over the search space as *Kinetic Diversity*.

## 5 Perspectives and further work

This paper has presented *Folding Genetic Algorithms* as a new efficient class of Genetic Algorithms based on real meta-level evolution of a suitable metrics over the search space. From an empirical point of view, Dynamical Environments have revealed to be of a great help in characterizing metrics properties. Such Adaptive Metrics explains the effectiveness of FGAs in Dynamical Environments which opened this new field to evolutionist tools. Further work will consist in studying in more details how can such an optimal meta-mask be evolved and what properties it must feature thus enlightening us on FGAs dynamics.

## References


[1] H.C. Cobb and J.J. Grefenstette. Genetic algorithms for tracking changing environments. In *Icga-5*, 1993.

[2] P. Collard and J.P. Aurand. Dga: An efficient genetic algorithm. In *ECAI'94*, Amsterdam, 1994.

[3] P. Collard and C. Escazut. Relational schemata: A way to improve expressiveness of classifiers. In *Icga'95* [8], pages 397–404.

[4] P. Collard and C. Escazut. Fitness distance correlation in a dual genetic algorithm. In *Ecai'96*, 1996.

[5] P. Collard, C. Escazut, and A. Gaspar. An evolutionary approach for time dependant optimization. In IEEE Computer Society Press, editor, *IcTai96*, pages 2–9, November 1996.

[6] P. Collard and A. Gaspar. Royal road landscapes for a dual genetic algorithm. In *ECAI-96*, Budapest, 1996.

[7] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection : algorithms, analysis and implications. In *Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy*, 1996.

[8] L.J. Eschelman, editor. *Sixth International Conference on Genetic Algorithms*, San Francisco, California, July 1995. Morgan Kaufmann.

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

[10] J.J. Grefenstette. Genetic algorithms for changing environments. In R. Manner abd B. Manderick, editor, *Parallel Problem Solving from Nature 2*, pages 465–501. Elsevier Science Publishers B.V., 1992.

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

[12] J.R. Levenick. Inserting introns improves genetic algorithms success rate: Taking a cue from biology. In R.K. Belew and L.B. Booker, editors, *Fourth International Conference on Genetic Algorithms*, pages 123–127, San Francisco California, July 1991. Morgan Kaufmann.

[13] J.R. Levenick. Metabits: Generic endogenous crossover control. In Eschelman [8], pages 88–95.

[14] K. Peow NG and K.C. Wong. A new diploid scheme and dominance change mechanism for non-stationary function optimization. In Eschelman [8].

[15] K. Pettit and E. Swigger. An analysis of genetic based pattern tracking and cognitive based component tracking models of adaptation. In *Proceedings of National Conference on AI (AAAI-83).* Morgan Kaufmann, 1983.

[16] J.-L Segapeli, C. Escazut, and P. Collard. Dgp: How to improve genetic programming with duals. In *Proceedings of the 3rd International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA'97)*, 1997.

[17] F. Vavak and T.C. Fogarty. A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. In *Proceedings of the Society for the Study of Artificial Intelligence and Simulation of Behavior, workshop on Evolutionnary Computation'96*, pages 301–307. University of Sussex, 1996.