Misleading Functions Designed from Alternation

Philippe Collard **Manuel Clergue** **Francis Bonnin**

I3S Laboratory, University of Nice Sophia-Antipolis
Les Algorithmes/Euclide, 2000 route des Lucioles, Biot Sophia Antipolis, F-06410
{pc,clerguem,bonnin}@i3s.unice.fr

Abstract- This paper propose to design difficult functions for a GA where the deceptive attractor is at mid-distance from the global optimum. First, piecewise-linear trap functions of alternation are investigated. We consider alternation-based distance to enlighten the ability of fitness distance correlation coefficient to predict GA behavior on such functions. Then, we generalize to any function by the way of the derivative transformation applied on bit strings. These preliminary results support the following conjecture: derivative transforms difficult problems, where competition occurs between complementary strings, into misleading problems where crossover is an effective operator and competitors are at mid-distance from each other.

1 Introduction

A central problem with genetic algorithms is to characterize the functions that are difficult for them to optimize. Many attempts to characterize difficult problem focus on the notion of deception, defined in terms of the average fitness of competing schemas [9]. In this paper, we propose to design difficult functions where the deceptive attractor is at mid-distance from the global optimum. First, trap piecewise-linear functions of alternation will be investigated. We consider alternation-based distance to enlighten the ability of fitness distance correlation coefficient to predict the GA behavior on such functions. Then, we generalize to any function by the way of the derivative transformation applied on bit strings. Using an executable model, we analyze the dynamics of a GA on derived functions. Finally, performing schemata analysis, we will show that the derivative transformation induces a correspondence between Holland schemata and derived schemata. Although the derived function from a fully deceptive function is not fully deceptive, the derived problem remains GA-difficult. These preliminary results strengthen us in the following conjecture: derivative transforms difficult problems, where competition occurs between complementary strings, into misleading problems where crossover is an effective operator and competitors are at mid-distance from each other.

2 Trap Functions and FDC

2.1 Trap Functions

Trap functions are a well known function family designed in such a manner that a genetic algorithm would fail to optimize it. They are piecewise linear functions of the unitation, *i.e.* the number of genes set to '1'. They divide the total search space into two basins in the Hamming space, one leading to a global optimum and the other leading to a local optimum, such that the two optima are the bitwise complement of each other. Without loss of generality, the string with all zeros and the strings with all ones are assumed to be the globally and the locally optimal strings respectively. In this case, the unitation of a string is equal to the Hamming distance between it and the global optimum.

Two parameters, x_b and r, allow to set the width of the basin of attraction for each optimum, and the ratio between these ones. The function f_T is then defined by:

$$f_T(x) = \begin{cases} 1.0 - \frac{u_n(x)}{x_b} & \text{if } u_n(x) \leq x_b \\ \frac{r(u_n(x) - x_b)}{1.0 - x_b} & \text{elsewhere} \end{cases}$$

where $u_n(x)$ is the unitation divided by λ, the length of chromosomes.

Using *schemata theory*, Deb *et al.*([4]), first proposed theoretical conditions over x_b and r to make the function fully or partially deceptive. Under these conditions, the algorithm is deceived and is trapped in the basin of the local optimum.

2.2 Fitness Distance Correlation

Some researchers have established a correspondence between evolutionary algorithms and heuristic state space search ([5, 8]). This correspondence is based on the fact that both approaches can be seen as searching labeled graphs. *Evolutionary landscape* matches *state space*, *individuals* match *potential solutions* and *fitness function* matches *heuristic function*. In the landscape model, the label on a vertex is a measure of the worth of that vertex as a solution. Whereas for many heuristic searches, as *Means-Ends Analysis* or the *A* algorithm*, the label is interpreted as a measure of the distance to a goal. There are many results that show that the better an heuristic estimates the function which gives the exact distance to the goal, the better it will perform. From there, Jones

([5]) suggests that the ideal fitness function for evolutionary algorithms must provide a measure of how far away a good solution is. Then, he proposes to use the correlation between fitness and distance to the goal as a measure of difficulty for a genetic algorithm.

The definition of the fitness distance correlation (FDC) is quite simple. Given a set $F = \{f_1, f_2, \ldots, f_n\}$ of fitness values from n individuals and the corresponding set $D = \{d_1, d_2, \ldots, d_n\}$ of distances to the nearest optimum, the correlation coefficient is calculated as:

$$r = \frac{c_{FD}}{\sigma_F \sigma_D}, \text{ where } c_{FD} = \frac{1}{n}\sum_{i=1}^{n}(f_i - \overline{f})(d_i - \overline{d})$$

is the covariance of F and D, and σ_F, σ_D, $\overline{f}$ and $\overline{d}$ are the standard deviations and means of F and D.

From the correlation coefficient r, problems may be divided in three classes:

1. **deceptive problems** ($r \geq 0.15$), for which the fitness increases with the distance to optimum.
2. **hard problems** ($-0.15 < r < 0.15$), for which there is no correlation between fitness and distance.
3. **easy problems** ($r \leq -0.15$), for which the fitness increases when the distance decreases.

The second class corresponds to problems which are hard to predict, since the FDC brings little information. In this case, Jones recommends to examine the scatter plot distance vs. fitness, especially when r is near zero. This allows to distinguish between a problem such as *a needle in a haystack* (NIAH) from a problem with a symmetrical scatter plot, for example. They both have a null coefficient, but NIAH is a hard problem, while there are symmetrical problems which are easy.

Jones recommends to use distances related to genetic operators. However, it is not sure that such distances exist, and if so, that they are easily computable. Jones claims that Hamming distance allows a first approximation of a FDC more related to the operators. In order to validate the accuracy of the FDC, Jones uses it on problems with known difficulty. Results show undoubtedly the predictive ability of the FDC, even calculated with Hamming distance. These results are surprising, since the distances actually implied by operators are not taken into account. The only operator taken into account is the mutation, as the Hamming distance between two individuals is directly related to the number of bit flips to pass from one individual to the other.

2.3 Trap Functions and FDC

We calculated the exact value of FDC (that is considering all the points of the search space), according to x_b and r. Results are shown on figure 1. Using the boundaries given by Jones on the coefficient for each classes of problems, we are able to determine for each couple x_b and r if the function should be considered as easy or deceptive. In the figure 1, this is shown by the plot of the isolines 0.15, 0 and -0.15 in the plane (x_b,r).

These results are roughly identical to the ones from Deb & Goldberg ([4]), who used schemata theoretical framework to obtain the conditions on x_b and r for which the *Trap* function is partially deceptive.

These results are corroborated by the performance obtained with a "real" GA. We define performance as the number of runs for which the global optimum is found in less than a given number of generations (200), divided by the total number of runs (50). The figure 2 presents such performance measures obtained with a simple GA with standard settings on the *Trap* functions. It is shown that the experimental deceptive area (where performance is less than 0.05) corresponds approximately to the theoretical deceptive area found with the FDC.

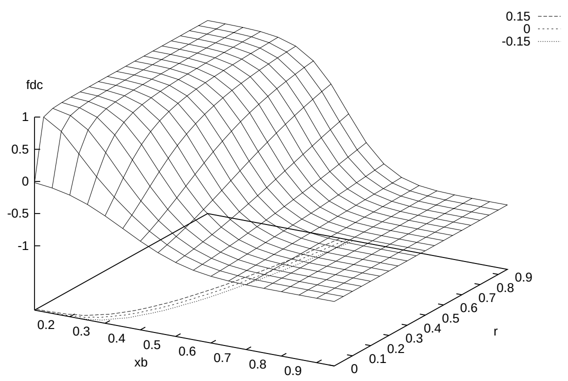

Figure 1: Standard FDC for *Trap* functions. Lines delimit areas in the plane (x_b, r) for respectively deceptive, hard and easy *Trap* functions, according to Jones' classification.

2.4 Altenberg's Counter-Example

According to Altenberg ([1]), the fact that the FDC is only a statistical and static measure, based on a distance which is apparently only bound to mutation, implies two assumptions: either Hamming distance is connected to the way genetic algorithms work; or this relation exists in a fortuitous way among the test set chosen by Jones. In which case, counter examples exist for which this relation does not hold, and which, therefore, deceive the FDC.

Since there seems that there is no relation between recombination operators and Hamming distance, and that mutation is supposed to play a marginal role in genetic algorithms, Altenberg claims that it is possible to construct a counter example. The counter example he constructs is GA-easy, but the correlation between distance

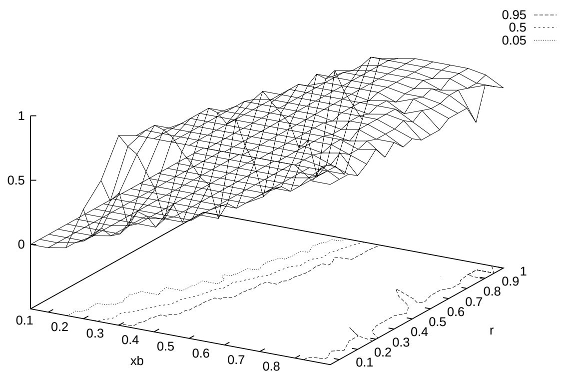

Figure 2: Performances of a GA on *Trap* functions, according to parameters r and x_b.

and fitness to optimum is null by construction. Further, the observation of the scatter plot gives no more informations. This counter example deceives the Jones' conjecture which claims that if the FDC is close to 0 and if the scatter plot exhibits no particular structure, then the problem is GA-difficult.

Moreover, Quick *et al.* ([6]) construct a class of problems, called *ridge functions*, which are GA-easy with a high positive correlation. While the Altenberg's counter example is prone to discussion, in particular on the definition of the GA-easiness, the counter example of Quick *et al.* is clear: there are functions that the FDC predicts misleading and which are in fact easy. Nevertheless, these two counter examples exploit known weaknesses of the FDC: its nullity for the symmetrical functions and the low contribution of a particular path in the global calculation. Besides, Quick *et al.* ([7]) recognize that the FDC calculated with the points actually sampled by the GA gives better results.

Nevertheless, the success of the FDC on a large number of functions remains an unsolved question. Collard *et al.* ([3]) bring some elements of response, exhibiting a correlation between Hamming distance and instability implied by crossover.

3 Alternation Trap Functions

In this section we design trap functions from alternation, *i.e.* the number of discontinuities between consecutive bits in the bitstring. In designing such functions, our aim is twofold: primarily, we hope for a new kind of misleading GA behaviors; secondly, we conjecture that this new type of GA hardness is rather related to crossover than to mutation.

3.1 Definition

Designing such piecewise-linear fitness functions of alternation, we should be able to guide the genetic search along a path, either towards the global optima or towards local optimum. Conditions are that fitnesses are monotonically increasing along the path, and that alternation measure is really correlated with crossover. We call *A-Trap* such functions. They depend on the total number of sequences 01 or 10 in a string and not on the positions of the alternations. So, they are defined on the binomial distribution of the space induces by alternations. As a consequence, *A-Trap* functions would have the following properties:

1. All the points with the same number of alternations have the same fitness.
2. Symmetry with respect to bit value, that is $f(x) = f(\overline{x})$, where $x \in \{0,1\}^\lambda$ is a bitstring of length λ, and $\overline{x}$ is its bitwise complement.
3. According to the above property, the fitness Hamming distance correlation coefficient is zero.

A-Trap functions, as their unitation based cousins, divide the search space into two basins. Each one is associated with a path in the alternation space, one leading to the global optima and the other leading to the local optima.

More precisely, we define *A-Trap* functions this way:

$$f_{AT}(x) = \begin{cases} 1.0 - \frac{a_n(x)}{x_b} & \text{if } a_n(x) \leq x_b \\ \frac{r(a_n(x) - x_b)}{1.0 - x_b} & \text{elsewhere} \end{cases}$$

where $a_n(x)$ is the number of alternations divided by $\lambda - 1$, the max number of alternations in a string of length λ. The parameter x_b allows to set the width of the attractive basin for each optimum, and the parameter r controls the ratio between the fitness of two optima. In fact, *A-Trap* functions admit as optima two pairs of complementary strings, the global one $(00..0, 111..1)$ and the local one $(0101.., 1010..)$. Let us notice that, opposite to the case of unitation trap functions (*U-Trap* in the following), the global optima and the local ones are not complementary strings. Instead, the Hamming distance between global optima and local ones is $\lambda/2$.

3.2 A-Trap Functions and Crossover

An open problem is to characterize the types of fitness landscapes for which crossover will be an effective operator. Altenberg ([1]) proposed to use the number of alternations (discontinuities) as a measure of *crossover* distance between solutions. His argument is based on the following fact: considering a sequence of crossover events on complementary strings that produce a path to the optimum, as one moves farther from the optimum along the path, the number of alternations increases by one with each step. So, the number of single-point crossovers needed to transform a complementary pair into the global optimum and its complement can be used as a measure of difficulty to move thought the search space by crossing over.

3.2.1 FDC Coefficients and Trap functions

For *A-Trap* functions, the FDC coefficient calculated with Hamming distance is null, since they are symmetrical functions. According to Altenberg ([1]), alternation can be viewed as a distance related to crossover. Then, he proposed to calculate a FDC coefficient using alternation instead of Hamming distance. This coefficient should give a better indication of the difficulty, since this *distance* is more related to GA dynamics.

In the case of *A-Trap* functions, it is worth noticing that function giving the FDC coefficients calculated with alternation as distance (FAC in the following), according to r and x_b, is exactly the same than the one using Hamming distance for *U-Trap*. Indeed, as we will see later, there is a mapping from $\{0,1\}^{\lambda}$ to $\{0,1\}^{(\lambda-1)}$, such that a *A-Trap* defined on $\{0,1\}^{\lambda}$ becomes a *U-Trap* with the same r and x_b on $\{0,1\}^{(\lambda-1)}$. So the FAC coefficients, according to r and x_b, for *A-Trap* are the same than FDC for *U-Trap* (see fig. 1).

3.3 Scatter Plots Distance/Fitness

In order to predict the dynamics of a GA on trap functions, two kind of landscape are taken into consideration: the unitation/fitness one and the alternation/fitness one. We expect that, examining the scatter plots on the first (resp. last) space, gives information on the ability of a crossover (resp. mutation) based GA to move through the landscape. The structure of the scatter plots alternation/fitness is given by the shape of the trap itself as the function is designed from this space. The structure of the scatter plots unitation/fitness for *A-Trap* functions is symmetrical and complex. There are many local optima with a high fitness. Points with only one alternation may be far away from the global optima (see fig. 3).

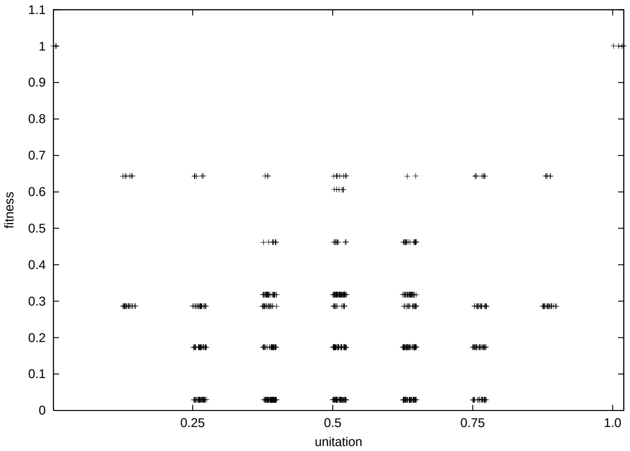

Figure 3: Scatter plot in the unitation fitness plane for a *A-Trap* with $\lambda = 8$ (Noise is added to unitation to render point density).

3.4 Experiments

3.4.1 Converge rates vs. x_b and r

One can think that the fitness alternation correlation (FAC) coefficient is relevant to represent the GA difficulty as alternation measures the distance between points according to crossover. More, examining this coefficient allows to raise the ambiguity of a zero FDC value.

We conduct experiments, on *A-Trap* functions with a simple genetic algorithm. The length of strings was 26, we used a population of 50 and a tournament selection scheme without elitism. As in the case of *U-Trap*, the actual performance measures are coherent with the FAC predictions (see figure 4). However, in the *easy area*, the performance rates are lower than for *U-Trap* (0.8 instead of 1.0). This should be due to the fact that the more important the diversity is, the more efficient the crossover is. When diversity becomes too low, crossover becomes inoperative. Such behavior reinforces our hypothesis that the crossover is the main operator for *A-Trap* function. Moreover, if we use an operator which complements strings at a given (low) probability (remember that complementary strings have the same fitness), performances are improved (see fig. 5).

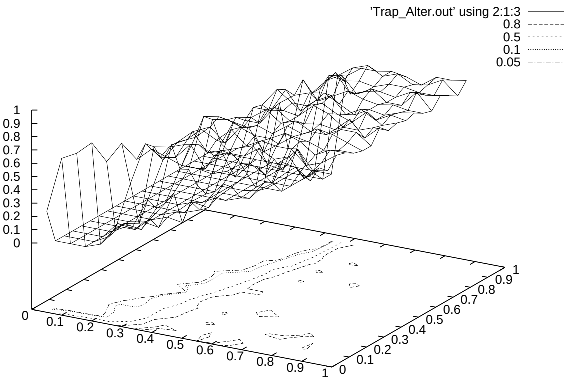

Figure 4: Performances of a GA on *A-Trap* functions.

3.4.2 Dynamics in the Unit./Alter. Space

In order to track the role of each operators, crossover or mutation, in the GA dynamics, it can be useful to

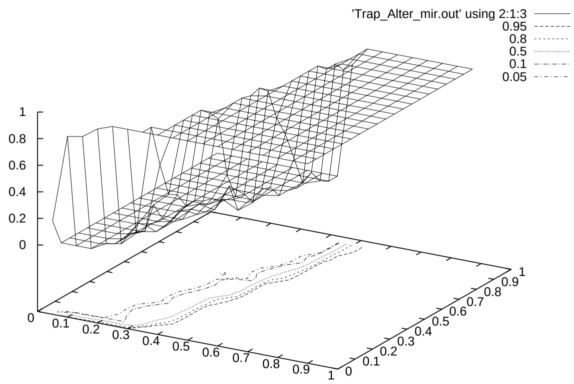

Figure 5: Performances of a GA which the complement operator on *A-Trap* functions.

plot the trajectory of the population over time in the unitation/alternation space.

The idea is to exploit another degree of freedom in order to visualize the dynamics by plotting the trajectory of the population centroid in the unitation-alternation space. This kind of plotting gives useful insight on the respective role of crossover and mutation in the GA dynamics (figures 6, 7 and 8). We plot such graph for different crossover rate, with the mutation rate set to 0.005, for the *A-Trap* defined by $r = 0.75$ and $x_b = 0.4$. For an high crossover rate, the GA is deceived, while it finds one global optimum when the crossover rate is lower (under 0.3).

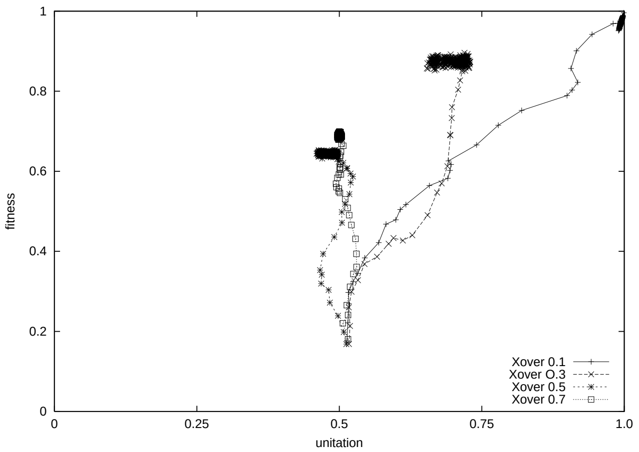

Figure 6: Population trajectory in the scatter plots unitation/fitness

3.5 Extension to Other Optimum

One can extend the definition of *A-Trap* functions by choosing any point as the global optimum. Let us remember that for the *A-Trap* functions defined until now,

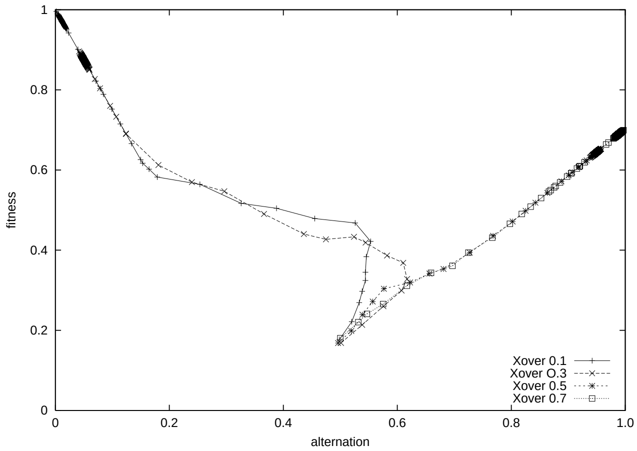

Figure 7: Population trajectory in the scatter plots alternation/fitness

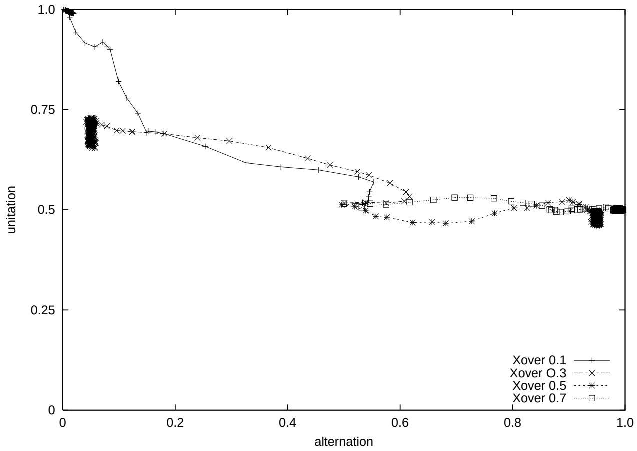

Figure 8: Population trajectory in the scatter plots alternation/unitation

global optima are the strings of all 0's or all 1's, and deceptive attractors are for maximal alternation. Any string $\dot{x}$ and its complement will become the global optima by mapping each string x in $x \oplus \dot{x}$, where $\oplus$ denotes componentwise exclusive-or. So the string of all 0's becomes $\dot{x}$, the string of all 1's becomes $\overline{\dot{x}}$, and deceptive attractors differ from global optima on the odd or even locus only. Consequently, the hamming distance between a local optimum and a global one is still $\lambda/2$.

4 Generalization

In the previous section we have directly designed *A-Trap* functions from the number of alternations in the bit strings. One another way to build such a function is to derive it from one *U-Trap* function by a mapping defined on the search space.

4.1 Derivative Defined on the Search Space

The class of *U-Trap* functions could be transformed into *A-Trap* functions through the derivative transformation defined as follows: for each string x in $\{0,1\} \times \Omega$ (noted Ω^+), the string $d(x)$ in Ω is obtained by replacing each pair ab of two consecutive bits in x, by 0 if a is equal to b and 1 elsewhere. For instance $d(0010) = 011$. Let be T a *U-Trap* function defined on Ω, the following diagram induces the *A-Trap* function T' defined on Ω^+ by commutativity: $T'(x) = T(d(x))$.

Observe that the transformed function T' is symmetric as, for all string x, $T'(x) = T'(\overline{x})$. As a consequence, the fitness Hamming distance coefficient is zero. More, we have the following property:

$$u_n(d(x)) = u_n(d(\overline{x})) = a_n(x)$$

4.2 Using Derivative to Transform Problems

Our aim is not to transform deceptive functions into easy functions but rather into *A-deceptive* functions. As we do for Trap functions, one can generalize the use of derivative to transform any problem defined on Ω into a problem defined on Ω^+. Let f be a real valued fitness function such that $f : \Omega \to R$, the derived function f' is defined by: $f'(x) = f(d(x))$. As a consequence, the derived function f' would have the following properties:

1. Symmetry with respect to bit value, $f'(x) = f'(\overline{x})$ (Induced by $d(x) = d(\overline{x})$).
2. The fitness Hamming distance correlation coefficient is zero.
3. The hamming distance between two complementary strings in Ω becomes equal to $\lambda/2$ in Ω^+.
4. Only schemata with more than one defining bit may have an average fitness different from the overall average fitness

4.3 Schemata Analysis: an Example

The derivative transformation induces a correspondence between Holland schemata in Ω and derived schemata from Ω^+. The difference between a GA working on f and a GA working on f' is in the related schemata; a GA explores f with respect to the Holland schemata on Ω, while it explores f' with respect to the derived schemata.

We are going to establish the mapping between Holland schemata belonging to Ω^+ and derived schemata on Ω. We examine the case of a fully deceptive function f defined on 3-bit strings ([9] and see table 1). For instance, for schemata at order 1, the three partitions are deceptive: $f(1\sharp\sharp) < f(0\sharp\sharp)$, $f(\sharp 1\sharp) < f(\sharp 0\sharp)$ and $f(\sharp\sharp 1) < f(\sharp\sharp 0)$. The symmetrical derived function f' is defined on $\{0,1\}^4$ (see table 2). Examining the six order two partitions in Ω^+ according to their image under d in Ω, we can distinguish three classes:

Table 1: Fitness value for the f function

String	fitness
000	*28*
001	26
010	22
011	0
100	14
101	0
110	0
111	**30**

Table 2: Fitness value for the f' function

String	fitness
0000,1111	*28*
0001,1110	26
0010,1101	0
0011,1100	22
0100,1011	0
0101,1010	**30**
0110,1001	0
0111,1000	14

1. Three partitions correspond to the order one partition in Ω: $d(00\sharp\sharp) = 0\sharp\sharp$, $d(\sharp 00\sharp) = \sharp 0\sharp$, and $d(\sharp\sharp 00) = \sharp\sharp 0$. In these partitions, the winner is the deceptive attractor.
2. Two partitions $0\sharp 0\sharp$ and $\sharp 0\sharp 0$, are not relevant for competition between optima as a schemata contains both the global optima and the deceptive attractor.
3. One partition where the winner is the global optimum: $f'(0\sharp\sharp 0) < f'(0\sharp\sharp 1)$. The derivative of a schema from this partition is not an Holland schema. For instance, $d(0\sharp\sharp 0) = \{011, 000, 101, 110\}$ can not be represented by a schema. As a consequence, the derivative function is not fully deceptive.

4.4 Experimental Results

We analyze the dynamics of a GA on derived function using the Whitley's executable model ([10]). It uses an infinite population, no mutation and requires the enumeration of all points in the search space. Assuming proportionate reproduction, one point crossover, no mutation and random mating, an equational system can be used for modeling the proportion of individuals over time. We apply this model on the derivative fully deceptive function defined above (see table 2).

For a low crossover rate, the algorithm finds the global optimum (see figure 9 and 10). If the crossover rate is set higher, the GA is trapped (see figure 11 and 12). On figures 9 and 11, each point corresponds to the mean unitation and the mean alternation of the population at a given generation. On figure 9, with a low crossover rate (0.3), it can be observed that the GA finds the global optimum (unitation 0.5 and alternation 1.0). While, with a higher crossover rate (0.9), see figure 11, the GA is trapped in the local optimum (unitation 1.0 and alternation 0.0). These observations are corroborated by figures 10 and 12, which are the evolution over time of the proportions of the two global optima and the two local ones.

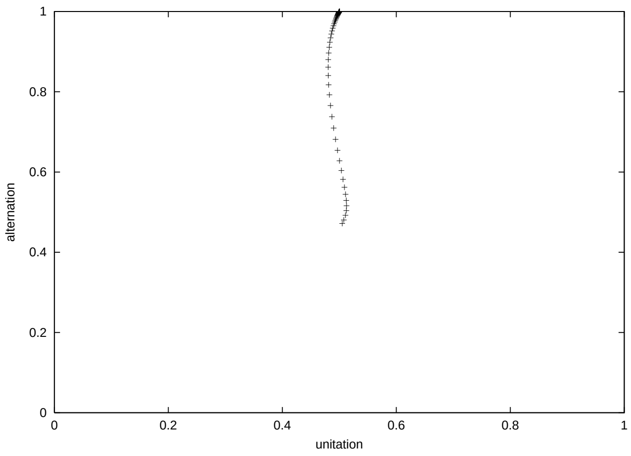

Figure 9: Trajectory in the normalized unitation/alternation plane (crossover rate is 0.3).

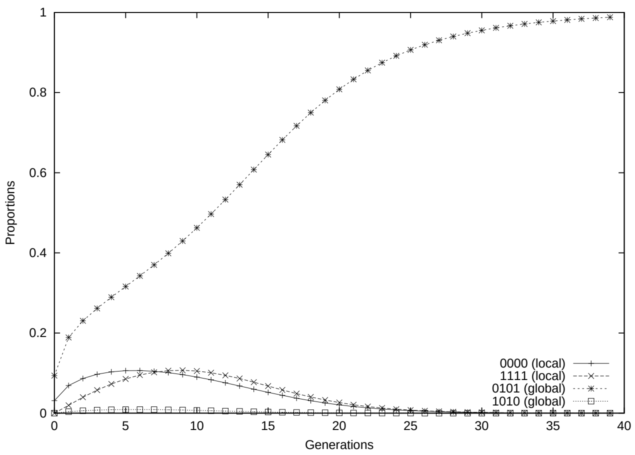

Figure 10: Evolution over time of global optima and local optima proportions. (crossover rate is 0.3).

5 Conclusions

In this paper, we have proposed to design difficult functions for a GA to optimize where the deceptive attractor is at mid-distance from the global optimum. First, piecewise-linear trap functions of alternation call *A-Trap*

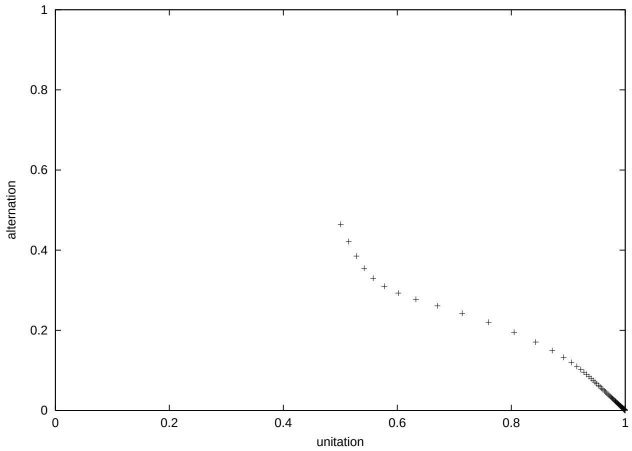

Figure 11: Trajectory in the normalized unitation/alternation plane (crossover rate is 0.9).

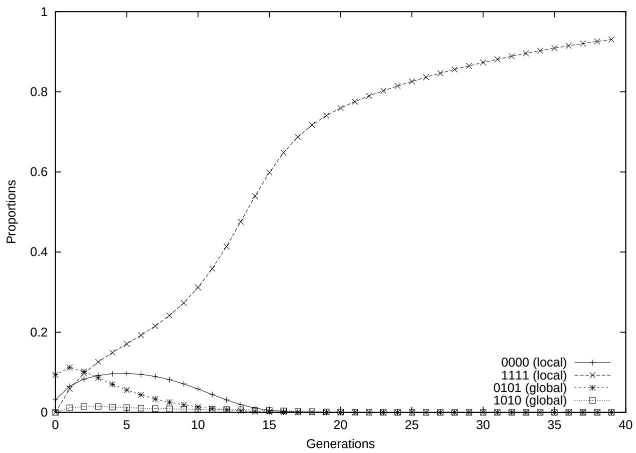

Figure 12: Evolution over time of global optima and local optima proportions. (crossover rate is 0.9).

functions have been investigated. It is shown that the fitness alternation based distance correlation coefficient permits predicting the GA behavior on such function. So, it allows to raise the ambiguity induces by a zero fitness Hamming distance correlation coefficient. Further than Trap functions, this first stage is generalized to any function by the way of the derivative transformation applied on bit strings. Using the Whitley's executable model, we analyze the dynamics of a GA on derived functions. Preliminary experimental results corroborate our conjecture: that is derivative transforms difficult problems, where competition occurs between complementary strings, into misleading problems where crossover is an effective operator and competitors are at mid-distance from each other. Performing schemata analysis on an example, we have shown that the derivative transformation induces a correspondence between Holland schemata and derived schemata. The difference between a GA working on a function and a GA working on its derivative is in the related schemata. Although the function derived from a fully deceptive function is not fully deceptive, the problem remains GA-hard; so, in future works, it may be necessary to refine the schemata analysis on a-deceptive problems.

Bibliography

[1] L. Altenberg. Fitness Distance Correlation Analysis: An Instructive Counterexample. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*. Morgan Kaufmann, 1997.

[2] M. Clergue and P. Collard. Genetic heuristic for search space exploration. In *proceedings of the 99' International Artificial Intelligence*, pages 1218–1224. Morgan Kaufmann, 1999.

[3] P. Collard, A. Gaspar, M. Clergue, and C. Escazut. Fitness distance correlation, as statistical measure of genetic algorithm difficulty, revisited. In *Proceedings of the European Conference on Artificial Intelligence*, pages 650–654. John Witley & Sons, Ltd, 98.

[4] K. Deb and D. E. Goldberg. Analysing deception in trap functions. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 93–108. Morgan Kaufmann, 1993.

[5] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. Eshelman, editor, *ICGA'95: Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.

[6] R.J. Quick, V.J. Rayward-Smith, and G.D. Smith. Fitness Distance Correlation and Ridge Functions. In *Proceedings of the fifth Conference on Parralel Problems Solving from Nature*, 1998.

[7] R.J. Quick, V.J. Rayward-Smith, and G.D. Smith. Ridge Functions. Technical report, University of East Anglia, 1998.

[8] W. A. Tackett. Greedy recombination and genetic search on the space of computer programs. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 271–298. Morgan Kaufmann, 1995.

[9] L. D. Whitley. Fundamental principles of deception in genetic search. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann, San Mateo, CA, 1991.

[10] L. D. Whitley. An executable model of a simple genetic algorithm. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 45–62. Morgan Kaufmann, 1993.