DGA and Pareto Elitism : Improving Pareto Optimization

Manuel Clergue **Philippe Collard** **Alessio Gaspar**
University of Nice-Sophia Antipolis, I3S Laboratory — CNRS
250 Av. A. Einstein, Sophia Antipolis, 06560 Valbonne FRANCE
email: {clerguem,pc,gaspar}@alto.unice.fr
web:http://wwwi3s.unice.fr/mage.html

Abstract

Previous works have shown the efficiency of a new approach for the Genetic Algorithms, the Dual Genetic Algorithms, in the multiobjective optimization context. Dual Genetic Algorithms make use of a meta level to enhance the expressiveness of schemata, entities implicitly handle by Genetic Algorithms. In this paper, we show that this approach, coupled with a new method, Pareto Elitism, leads to very interesting results, in particular on an adaptation for multiobjective optimization of Royal Road Functions, the Multi Royal Road Functions.
We begin with a quick reminder on multiobjective optimization, on what makes it different from single objective optimization and what has been done in this context. After this, the Dual Genetic Algorithm principles are briefly exposed, as well as previous results obtained. Then, we present Pareto Elitism, combining steady state and sharing techniques for Pareto optimization, and its behavior on Multi Royal Road Functions.

1 Pareto Optimization

Many problems can be seen as optimization over a cost function. In such a problem, we are given a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the space being searched. For that f, we seek one $x^* \in \mathcal{X}$ or a subset $\mathcal{X}^* \subset \mathcal{X}$, which gives rise to a particular $y^* \in \mathcal{Y}$ or a particular subset $\mathcal{Y}^* \subset \mathcal{Y}$. The most studied case is the one where $\mathcal{Y} = \mathbb{R}$. In this context, we seek the x^*'s that extremize the given f. Many methods, as Simulated Annealing, Hill Climbing or Genetic Algorithms, were developed and each have particular advantages or disadvantages that make them well suited for a particular class of such single objective functions.

However, engineers are very often faced with problems expressed in terms of many criteria, or objectives, often competitive, which should be satisfied simultaneously. In this case, $\mathcal{Y} = \mathbb{R}^n$, where n is the number of criteria in the problem. The mapping f is now a vector function, with n components $f_k, k = 1...n$, where each component has to be optimized. This mapping associates for each $x \in \mathcal{X}$ a vector $y \in \mathcal{Y}$, such that $y = f(x) = (f_1(x), ..., f_n(x))$. The goal is still to find particular x^*'s which give rise to optimal y. But the difficulty in the multiobjective context is that there is no trivial way to say that one y is better than another one and so, should be preferred.

The Pareto optimality gives a definition of optimality for multiobjective problems. A Pareto optimal solution for a multiobjective problem is a solution whose objective vector components could not been improved simultaneously. More formally, consider, without loss of generality, the minimization of the n components of a vector function f. Then, a $x_u \in \mathcal{X}$ is said to be Pareto optimal if and only if there is no $x_v \in \mathcal{X}$ for which $v = f(x_v) = (f_1(x_v), ..., f_n(x_v))$ *dominates* $u = f(x_u) = (f_1(x_u), ..., f_n(x_u))$, that is, there is no $x_v \in \mathcal{X}$ such that:

$$\forall i \in \{1, ..., n\}, v_i \leq u_i \quad \wedge \quad \exists i \in \{1, ..., n\} \mid v_i < u_i$$

The vector u is said to be *non-dominated.*

The set of all Pareto optimal solutions is referred as the Pareto optimal set, the Pareto front or even the trade-off surface. The search of such a set is called Pareto optimization.

Obviously, if the objectives are competitive, the Pareto set is not reduced to a single element. This is mainly why Pareto optimization involves different reflections with regards to classical optimization techniques, including Genetic Algorithms (GA), which take care of a single objective.

2 Genetic Algorithms and Multiobjective Optimization

Scalar optimizations methods such as Hill Climbing, Simulated Annealing and even Genetic Algorithms, traditionally perform poorly on multiobjective optimization (MOO). Nevertheless, unlike the other methods, GAs can be easily adapted to such optimization problems. To date, evolutionist approaches to MOO can be considered as belonging to one of the two following category.

2.1 Non Pareto Approaches

Since numerous relevant scalar optimization methods are available, the first approach has been to reduce MOO to single objective optimization. This can be done by using a weighted sum to combine all objectives into an unique scalar value. Limitations come from the fact that choosing weights in such cases is critical and need knowledge about the problem at hand. Moreover, such a linear combination does not enable the GA to find solutions located in concave regions of trade-off surfaces. One can also optimize separately each objective ([11]), but it has been shown ([6]) that both approaches are identical as they are both limited by their intrinsic linearity. Although previous approaches seek for non dominated final solutions, none makes use of the notion of Pareto optimality.

2.2 Pareto Approaches

Pareto partial order has been used by Goldberg ([8]) as well as Fonseca and Fleming ([5]) to address MOO. For instance, one can start with $i = 1$, assign rank i to all non-dominated individuals of the current population, remove them, increase i and iterate. Different versions have been developed based on the number of dominating individuals, or using a tournament selection method rather than ranking ([9]).
In all case, the GA must converge toward the set of Pareto solutions and stabilize it. Such sets are most of the time unstable because crossover is not able to be closed between individuals subsumed by non compatible schemata. Next section explains the concept of schemata which is essential to Genetic Algorithms. We then present a new approach allowing a wider range of sets to be compatible in order to easer GAs optimization in the multiobjective context.

3 Dual Genetic Algorithms

This section introduces Dual Genetic Algorithms (DGA) and analyses in details schemata they rely on.

3.1 Relational Schemata

Given chromosomes defined by $c = \{0, 1\}^\lambda$, a schema is a generic pattern $s = \{0, 1, *\}^\lambda$ which subsumes (can relax to) a set of chromosomes by instanciating occurrences of the joker character (*) inside it. For instance, $*1$ corresponds to $\{01, 11\}$ but $\{00, 11\}$ can not be represented by a single schema. This is the expressiveness limitation for *Positional Schemata* (P-schemata) i.e. schemata providing information about each locus independently to the others.
Relational Schemata (R-schemata) have been introduced ([2]) to express informations about relations between different loci. Such schemata are defined by $s = \{\mathrm{X}, \overline{\mathrm{X}}, *\}^\lambda$ where X stands for a boolean variable and $\overline{\mathrm{X}}$ its complement. Now, sets such as $\{00, 11\}$ and $\{10, 01\}$ can be expressed by R-schemata as respectively XX and $\mathrm{X}\overline{\mathrm{X}}$.

3.2 Implementing R-Schemata

The key idea is to design a Genetic Algorithm able to implicitly process R-schemata just as Simple Genetic Algorithms do with P-schemata. By doing so we take advantage of the enhanced expressiveness of schemata to increase possible stable sets of individuals size.
Basically, R-schemata are implemented by the use of meta-genes ([2]). Each chromosome is added a meta-gene in front of it which manages the interpretation of the rest of the string. If meta-gene is set to 0, rest of the string remains unchanged, otherwise it is complemented.
Thus, for a given problem involving λ gene long chromosomes and thus defining a search space[1] $\Omega = \{0, 1\}^\lambda$, the DGA will have to optimize a so-called *dual search space* $< \Omega >= \{0, 1\} \times \Omega$. Evaluation will be achieved by first *transcriptising* genotypes from $< \Omega >$ back to Ω according to the previously stated rule in order to be able to apply them the fitness function corresponding to our considered problem.
In such a context, a P-schema of $< \Omega >$ can either represent a P-schema or R-schema of Ω. When the meta-locus is instanciated, for instance, $1 * 0$, the P-schema of $< \Omega >$ corresponds to a P-schema of Ω, here $*1$. On the contrary, when the meta-locus is not

[1] There is a mapping, called decoding, from Ω, the space the GA searches, to $\mathcal{X}$, the search space of the problem.

instanciated, for instance $*01$, it will corresponds to a R-schema of Ω, here $X\overline{X}$.

3.3 Mirroring

Along with this enhanced schemata expressiveness, DGAs also provide two separate levels for the evolutionist dynamics to be considered. Phenotypic homogeneity is now enabled to coexist along with genotypic diversity obtained through presence in the population of complementary chromosomes.
In order to ensure such a diversity, we introduce a new operator, the mirroring operator. This one simply inverts the whole chromosome by complementing each bit. The genotype of the individual is modified, but its phenotype remains unchanged. Moreover, when a chromosome is cross with an inverted version of itself, the result may be any point in Ω. So, while not changing the efficiency of the population for the problem, mirroring increases the population capacity to produce new individuals.

4 DGA and Pareto Optimization

Previous work ([1]) has enlightened the benefits of using DGA on two multiobjective problems. In this section, we summarize results obtained. Then, in the next section, we introduce Multi Royal Road Functions.

4.1 MO Minimal Problem

We define as a MO Minimal Problem a problem involving two objective functions on $\lambda = 2$ gene long chromosomes. Our first problem ranks[2] solutions, highest first, as follows: $[\{00, 10\}, \{01, 11\}]$. The set $\{00, 10\}$, represented by schema $*0$ (and thus being a stable predicate) is the best tradeoff for this problem. We define a second problem ranking differently solutions : $[\{01, 10\}, \{11\}, \{00\}]$. This time, the best trade-off is set $\{01, 10\}$ which is unstable as not representable by a P-schema. Nevertheless, it is now possible to represent it by R-schema $X\overline{X}$ thus strongly suggesting DGAs as good candidates for such problem. Figures 1 and 2 show an instance of respectively the first and the second problem, where the two functions have to be minimized.

Simulations of either SGA or DGA dynamic showed that on first problem convergence was quick toward a stable population oscillating only between the two

[2] In the following, rank refers to Pareto rank as defined in 2.2

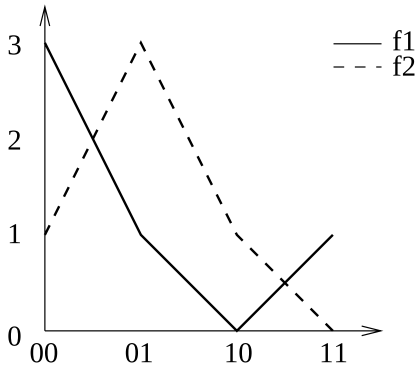

Figure 1. An instance of the first minimal problem

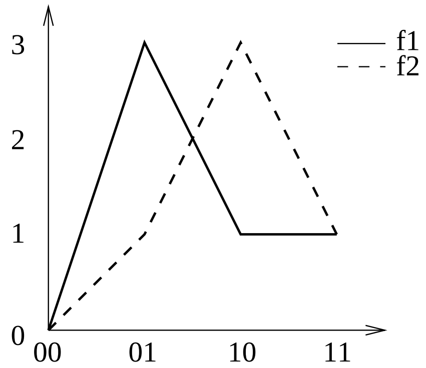

Figure 2. An instance of the second minimal problem

best trade-offs[3]. Second problem revealed to be intractable for SGAs while DGAs were plenty successful by finding out a stable set of genotypes transcriptising to phenotypes of the optimal set.

4.2 MO Trap Problem

As previous problem was based on $\lambda = 2$ genes long chromosome to be computable by an equational model of GA we chose to investigate a little more realistic problem called Multiobjective Trap Problems. The figure 3 shows the two designed functions. They are defined on the normalized unitation of the chromosomes[4], feature a local and a global optima, return values between 0 and 1, and should be minimized.
There are two Pareto sets;

- The local one (right) located between 0.9 and 1.

[3] This oscillation being due to the sharing method.
[4] i.e. the number of bits set to 1 divided by the total number of bits

- The global one (left) located between 0 and 0.1.

These functions rely on two parameters; r which measures the importance of the global Pareto set with regard to the local one, and x_b, which measures the relative importance of the basin of attraction of the two Pareto sets.
We experimented while tuning both parameters. Results allowed us to conclude that, contrary to SGAs, DGAs revealed to be able to find the good set for any parameters values with a good proportion of populations inside it.

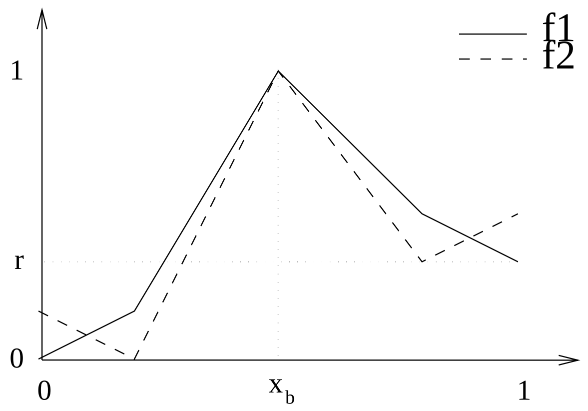

Figure 3. Multiobjective Trap Problems

For $r = 0$, both Pareto sets are equivalent so that there is a unique Pareto set where each element of each subset is at least at a hamming distance of 2 from any element of the other subset. Contrary to DGAs, SGAs were able to find only one subset as it is hard to make individual from incompatible schemata coexist. The explanation comes from the fact that individuals at a distance δ from each other in the basic space, may be either at distance δ or at distance $\lambda + 1 - \delta$ (where λ is the chromosome length in the basic space) in the dual space [3].

5 Pareto Elitism

As previously exposed, there are two ways to include multiobjective concept into Genetic Algorithms. For the reasons invoked above, we use the second one, which is the Pareto approach.
To design our method, we made use of two distinct techniques. The first one is the so-called steady state technique. The second one is the sharing technique.

5.1 Steady State for Pareto Optimization

Steady state technique, as defined by Syswerda in [12], replaces only a few members during one generation, leaving the others quiet. In order to do this, some new individuals are generated, using classical parental selection, eventually crossover and mutation operators. In order to insert these new individuals in the population, room must be made in this one. The worst technique to make room is random deletion. This leads to equivalent performance than SGA. There is a somewhat clever strategy, that is to delete the worst individuals. This technique is well suited for Pareto optimization, since there is a way to design a relevant ranking scheme in this context. The population is ranked and individuals with lower ranks are removed. From this technique, one can expect a improved stability, since it can be compared to a super elitism.

5.2 Sharing for Pareto Optimization

Pareto optimization aims to obtain a set of Pareto optimal solutions. This set, in general, is not reduced to a single point, since the objectives are most often competitive. However, the natural trend of a Genetic Algorithm is to converge toward a unique individual. To counterbalance this effect, several techniques have been developed. The general goal of such methods is to preserve diversity in the population. We use here the *sharing* technique ([8]). This method shares the number of expected children between individuals belonging to a same *niche*. The niches are, in general, defined with a neighborhood along either the genotypic space (e.g. Hamming distance between chromosomes) or the phenotypic one (e.g. Euclidean distance between fitnesses). For Pareto optimization, we use a sharing based on the Euclidean distance between the objective vectors. Sharing is used to obtain a density measure around an individual. The Pareto ranking considers equivalently all the non-dominated individuals in the population. This may lead to premature convergence or convergence toward a perhaps Pareto optimal but unique solution. This equivalence is removed by assigning an higher rank to individuals with smaller density.
The sharing has an orthogonal effect to the selection pressure. While selection makes the population converges towards a Pareto optimal solution, sharing makes the population spread over the Pareto set.

5.3 Progressive Sharing

Sharing involves a strong side effect. Indeed, consider two Pareto optimal solutions present in the

population. If at a given generation, a solution has a greater density than the other, this solution gets a lower expected number of children, and so will have a lower density in the next generation. There is a kind of instability that might be prejudical.
To overcome this difficulty, we modify the classical sharing scheme. Instead of calculate the density of an individual relatively to the whole population, we calculate it with only the population part following this individual, given any ordering of the population. This way, we ensure that in a niche, there is always one individual with the lowest density (i.e. 1) and so, that each optimal niche has at least one member ranked in the head of population.
This mechanism, combined with steady state and ranking, allows a greater stability and a wider found Pareto set, and prevents from premature convergence. We choose the name *Pareto Elitism*, since once a relative optimal solution is found, it stays in the population (elitism), but the search toward other regions of the Pareto front is encouraged through sharing.

6 Multi Royal Roads

6.1 Royal Road Functions

Royal Road Functions were designed to provide a set of easy problems for Genetic Algorithms [10, 7]. In fact they revealed to be not so easy and lead to interesting studies, in particular with DGA ([4]). These functions reward the presence of user defined blocks in the chromosomes. Since Genetic Algorithms implicitly use blocks to obtain better and better solutions, by specifying explicitly such blocks, one attempts to offer a "royal road" towards the optimum.
In this paper, we are interested with the first version of Simple Royal Road Functions. In this version, there are height blocks, defined by a user specified pattern (Figure 4). For an individual, each block matched by its chromosome, gives a constant reward (here 4). We use here height bits long patterns, giving 64 bits long chromosomes.

6.2 Combining Several RR Functions

Consider a two objectives problem where each objective is a RR Function, as defined above. Obviously, if the required patterns of both functions are identical, the two objectives aren't competitive and so the problem admits an unique Pareto optimal solution. Thus, in the following, we don't consider this case. On the contrary, if the two patterns aren't identical, the objectives are truly competitive. The solution is

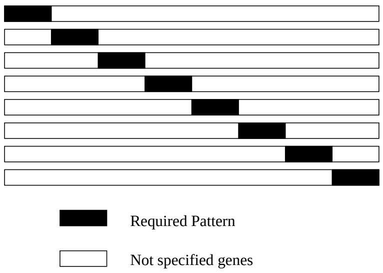

Figure 4. Blocks defining the Royal Road Function

a set of Pareto optimal solutions.
We construct such a problem because the number of such solutions is known, and depends only on the number of blocks. They are one more different[5] Pareto optimal solution than the number of blocks. So, in our case, the cardinality of the Pareto set is 9. In figure 5 the elements of this set are shown with cor-

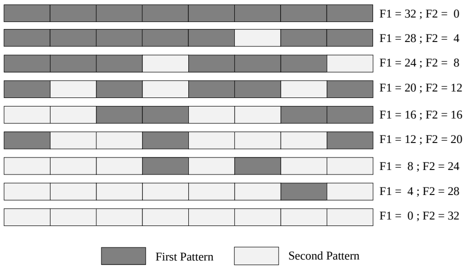

Figure 5. Solutions for the Multi Royal Road Function

responding chromosomes. It is worth noticing that there may be several chromosomes corresponding to a particular solution. These solutions are characterizable by the relation $f_1 + f_2 = 32$.
Thus, the Pareto set is well known. So, it is not difficult to count, for a given method the number of different Pareto optimal solutions found. This leads to the definition of another criterion to measure efficiency for Pareto optimization. This criterion is the coverage of the trade-off surface. The coverage must be as wider as possible. In the case of Multi RR function, since the cardinality of the Pareto set is small, one can expect that all the Pareto optimal solutions will be found.

6.3 Experimental Results

Figures 6 and 7 show the evolution of the number of different optimal solutions within the current pop-

[5]in the objectives space ($\mathcal{Y}$) not in the variables space ($\mathcal{X}$).

ulation. The two objectives are RR functions with required patterns set to a 8 ones block for the first function, and set to a 8 zeros These graphs are obtained by averaging 50 different runs for each methods.
The figure 6 is a comparison between Dual Genetic Algorithm and Standard Genetic Algorithm using both the Pareto Elitism scheme for selection and reproduction. The steady part of the population is 60 percent, that is, each generation only 40 percent of the population is generated. The population size is set to 100, that is very much more than the cardinality of the target set. The crossover and the mutation rate are respectively set to 1.0 and 0.005. For the mirroring operator, the application rate is 0.1.
As one can see, for the DGA, there is a convergence toward around 8, when the SGA finds only around 1 solution. A remark has to be done. These graphs don't give a true view of what is really happens. Indeed, the final populations obtain with the DGA get either 9 or 0 solutions. Once an optimal solution appears, the others are quickly found, as figure 8 shows it. In introduction to RR functions, we said that these functions were not so easy. The fact that the DGA is not able to find one solution is due to an intrinsic hardness of RR functions, which is the difficulty to construct a rewarded block by mutation or crossover without information telling how near the current block is. For instance, considering only mutation, if the current block is at a Hamming distance of δ from the pattern, δ mutations must occur to find it. So, a more relevant way of reading the DGA curve is the percentage of runs that have found the entire Pareto set at the generation t. These considerations don't stand for the SGA as the number of found Pareto solution in the final population is between 0 and 3 for the 50 runs.
The figure 7 shows comparison between a DGA with Pareto Elitism and a DGA with a Niched Pareto Tournament method (drawn from [9]). The settings are the same than above except that for the tournament method, steady state is set to zero (this is a generational method), crossover rate is set to 0.8 and the tournament size parameter, which tells how many individuals are implied in a tournament is set to 10. As we expect, the tournament method curve is strongly noisy. This instability unable the method to find the 9 solutions.

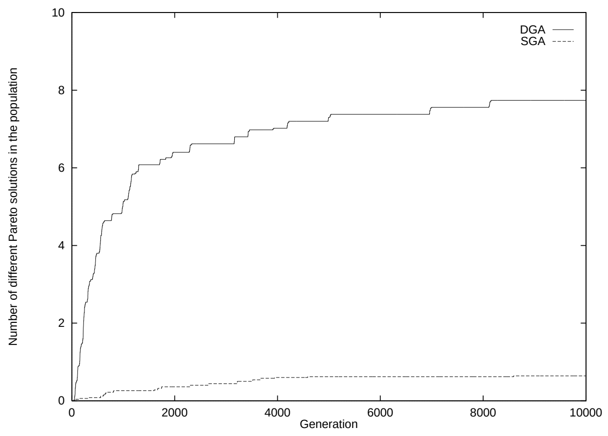

Figure 6. Comparison between DGA and SGA with Pareto Elitism

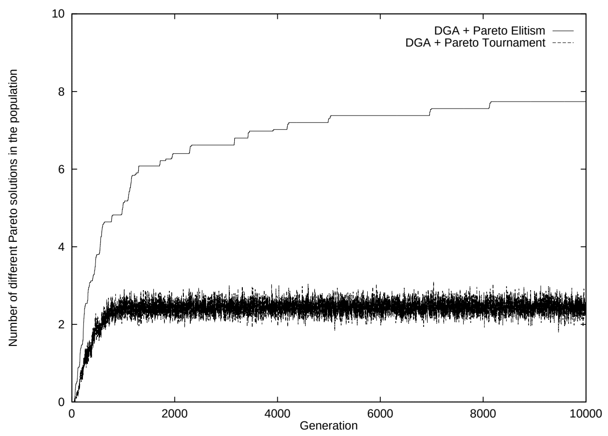

Figure 7. Comparison between Pareto Elitism and Pareto Tournament

7 Conclusion

Previous works showed the efficiency of the Dual approach on multiobjective problems. This efficiency was mainly due to the expressiveness improvement the DGA involves. Here, we present a new method to address this class of problem. This method, called Pareto Elitism, offers a greater stability, a wider final Pareto set, than a tournament like method for example. Since we only modify the algorithm, the Duality concept still apply. So, the combination DGA and Pareto Elitism, seems to be a very efficient way of thinking the multiobjective optimization. We get very good results on theoretical problems. However, we still lack real engineering problems in order to validate our approach. There is an open question: do the advantages of such methods still remains when the problem size increases.
Our future works would be to look for such a real size problem, and to test different methods in order to see if the theoretical results still apply. In the same time, we work to improve more the expressiveness of the schemata the GAs implicitly processes.

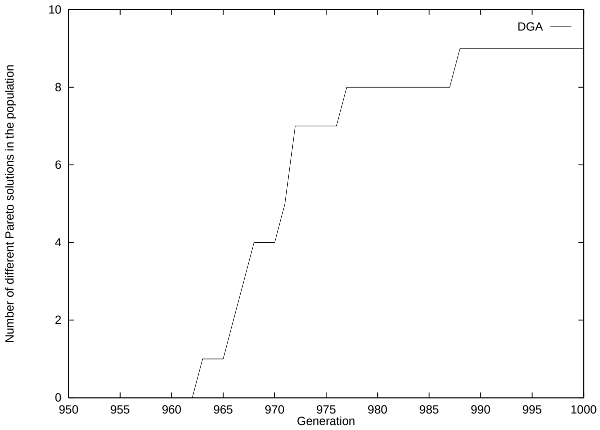

Figure 8. Zoom on a typical run for DGA with Pareto Elitism

The *meta* approaches, which the DGAs belongs, are just beginning, so there is still a lot of works to do.

8 Acknowledgment

Some of the experiences contained in this paper were conducted using the SUGAL Genetic Algorithm package, written by Dr. Andrew Hunter at the University of Sunderland, England.

References

[1] M. Clergue and P. Collard. Pareto optimization and dual genetic algorithm. In *Proceedings of the Third International Conference on Artificial Neural Networks and Genetic Algorithms*, 1997. in press.

[2] P. Collard and C. Escazut. Relational schemata: A way to improve the expressiveness of classifiers. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetics Algorithms*, pages 397–404, 1995.

[3] P. Collard and C. Escazut. Fitness distance correlation in a dual genetic algorithm. In W. Wahlster, editor, *Proceedings of the Twelfth European Conference on Artificial Intelligence*, pages 218–222. Wiley, 1996.

[4] P. Collard and A. Gaspar. Royal road landscapes for a dual genetic algorithm. In W. Wahlster, editor, *Proceedings of the Twelfth European Conference on Artificial Intelligence*, pages 213–217. Wiley, 1996.

[5] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.

[6] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[7] S. Forrest and M. Mitchell. Relative building-block fitness and the building-block hypothesis. In *Foundations of Genetic Algorithms 2*, 1993.

[8] D.E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison Wesley, Massachussets, 1989.

[9] J. Horn and Nafpliotis N. Multiobjective optimization using the niched pareto genetic algorithm. In *Proceedings of the First IEEE Conference on Evolutionnary Computation*, 1994.

[10] M. Mitchell, S. Forrest, and J. Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. In F.J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems -First European Conference on Artificial Life-*. MIT Press, 1991.

[11] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J.J. Grefenstette, editor, *Genetic Algorithms and Their Application: Proceedings of the First International Conference on Genetic algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[12] G. Syswerda. A study of reproduction in generational and steady-state genetic algorithms. In Rawlins, editor, *Foundations of Genetic Algorithms*, pages 94–101, San Mateo, California, 1991. Morgan Kaufmann.