Dual Genetic Algorithm And Pareto Optimization

Manuel Clergue Phillipe Collard
University of Nice Sophia Antipolis, I3S Laboratory
bat 4,250 av. A.Einstein, 06560 Valbonne, France
clerguem@alto.unice.fr , pc@alto.unice.fr

Abstract

This paper deals with an important class of optimization problems, the Multiobjective Problems. A new Genetic Algorithm, called Dual Genetic Algorithm, is presented. Through two theoretical problems, we show that this approach appears to be efficient for Multiobjective Optimization.

1 Introduction

In general, real world problems are expressed in terms of many criteria, or objectives, often competitive, which should be satisfied simultaneously. Multiobjective Optimization (MOO) involves some different reflections with regard to classical optimization techniques, including Genetic Algorithms (GA), which often take care of a single objective.
In the Multiobjective Problem (MOP) context, there is no trivial way to say that a solution is better than another and should be prefered.
Pareto Optimality is a way to do this. 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 $\mathbf{f}$. A decision vector x_u is said to be Pareto optimal if and only if there is no x_v for which $v = \mathbf{f}(x_v) = (f_1(x_v), ..., f_n(x_v))$ *dominates* $u = \mathbf{f}(x_u) = (f_1(x_u), ..., f_n(x_u))$, that is, there is no x_v such that: $\forall i, v_i \leq u_i \quad \wedge \quad \exists i \mid v_i < u_i$. The vector u is said to be *non-dominated*.

1.1 Genetic Algorithms and MOPs

As we will see, scalar optimization methods, like Hill Climbing, Simulated Annealing or Classical Genetic Algorithms, aren't well suited for MOO. The power of GAs is that they are not uniquely a scalar optimization method. They can be easily adapted by incorporating the notion of Pareto Optimality.

Non Pareto Approaches

When powerful and well studied scalar methods are provided, the temptation is great to reduce anything to a scalar problem and then to apply these methods. This is done naturally in the MOO context by combining all the objectives in a single scalar value, by means of a linear combination for example. By this way the MOP is reduced to a single objective problem and classical GAs can be applied.
Two remarks have to be made about this method. Firstly, the combination give more or less importance to each objective. This involves some *a priori* knowledges on the problem. Secondly, points in concave regions of a trade-off surface cannot be found by optimizing a linear combination of the objectives, for any set of weights, as pointed out by Fonseca and Fleming[2].
To avoid the design of the combination, one could optimize each objective separately. With this point of view, Schaffer[5] proposed a GA extension called Vector Evaluated GA (VEGA). In VEGA, the next generation is selected by n fractions, each one according to each of the objectives. Although VEGA and other similar population based methods aim to obtain non-dominated solutions, they don't make a direct use of the Pareto optimality.

Pareto Approaches

Goldberg[3], first proposed a direct use of the Pareto optimality. His method is a ranking method based on a Pareto partial order. The non-dominated individuals are assigned rank 1, and then removed; the new non-dominated individuals are assigned rank 2, and so forth. Fonseca and Fleming[2] proposed a similar method based on another partial order defined by the number of dominating individuals.
Horn and Nafpliotis[4] gave another method, called

Niched Pareto GA (NPGA), not based on a population ranking, but on a tournament selection technique: the two candidates are compared to a random comparison subset of the population; if a candidate is dominated by the comparison set and the other not, the latter win; if neither or both are dominated the tie is broken by a niching technique.

The difficulty for the GA is now to converge toward a set of equivalent solutions, the Pareto optimal ones, and to stabilize it in the population. The solutions may be so different, formally may belong to incompatible *schemata*, that these tasks are impossible. This is why we introduce Dual Genetic Algorithms. Theoretically, this extension of Genetic Algorithms allows a wider class of solutions to be present in the same stable population.

2 Dual Genetic Algorithms

A schema is a way to identify a subset of individuals sharing some characteristics. Schemata link individuals having the same allele at given loci.

Unfortunately, schemata are unable to represent some sets of individuals. For instance the set $\{00, 11\}$ doesn't correspond to a schema. This is why we introduce the notion of Relational Schemata[1] and their implicit implementation, the Dual Genetic Algorithms.

2.1 Relational Schemata

Relational Schemata, or R-schemata, as defined by Collard and Escazut[1], are another type of equivalence class that can be defined over the search space. Instead of having equivalence relations based on the value at given positions, R-schemata are equivalent classes based on relations between loci. They are defined over the ternary alphabet $\{\mathrm{X}, \overline{\mathrm{X}}, *\}$, where the two symbols X and $\overline{\mathrm{X}}$ represent two complementary variables: if X is bound to 0 then $\overline{\mathrm{X}}$ is bound to 1 and vice versa. For instance, the set $\{00, 11\}$ corresponds to the R-schema XX and the set $\{01, 10\}$ corresponds to the R-schema $\mathrm{X}\overline{\mathrm{X}}$.

2.2 Dual Genetic Algorithms

Dual Genetic Algorithms (DGAs) are a quite simple implementation of the notion of R-schemata through a new encoding of the binary string developped by Collard and Escazut[1]. A *head-bit* is added to the string and manages the interpretation of the rest of the string. When this bit is set to 0, the rest of the string remains unchanged. If it is set to 1, the rest of the string is interpreted as its binary complement.

In a more formal way, let us consider λ-bit strings. The search space is then $\Omega = \{0, 1\}^{\lambda}$. The product space $\langle\Omega\rangle = \{0, 1\} \times \Omega$ is defined as the *dual space*. We defined a mapping T, called *transliteration*, from the dual space to the basic search space by: $\forall\omega \in \Omega$, $T(0\omega) = \omega$ and $T(1\omega) = \overline{\omega}$, where $\overline{\omega}$ is the binary complement of ω.

We apply genetic operators on the dual space. Strings from the dual space are mapped by *transliteration* to strings from the basic space and then the fitness function is applied.

A central property is that P-schemata over the dual space may represent P-schemata as well as R-schemata over the basic space. For instance, if $\lambda = 3$, the P-schemata $\underline{0}1 * 1$ and $\underline{1}0 * 0$ of $\langle\Omega\rangle$ are associated with the P-schema $1 * 1$ of Ω. On the other hand, the P-schema $\underline{*}1 * 0$ becomes the R-schema $\mathrm{X} * \overline{\mathrm{X}}$.

As DGA is a GA that operates on the dual space, any works concerning classical GA apply there, especially methods for MOO.

3 Minimal MOPs

Minimal Multiobjective Problems (MMOPs) are a class of problems with two objective functions and solutions coded with binary chromosomes of length 2. We constructed two problems from this class. The first one ranks solutions, from the highest to the lowest : $[\{00, 10\}, \{01, 11\}]$. So, the best trade-offs are $\{00, 10\}$. This set forms a stable predicate or equivalently, is strictly represented by the P-schema $*0$. The second problem ranks the solutions, still from the highest to the lowest : $[\{01, 10\}, \{11\}, \{00\}]$. The set of the best trade-offs is now $\{01, 10\}$. It can't be strictly represented by a P-schema, but it can by the R-schema $\mathrm{X}\overline{\mathrm{X}}$. Instances of these problems are shown Figure 1.

We realized a simulation of the dynamic of a SGA, with equations found in [6], that we adapted for a niched Pareto ranking method, for the two problems (Figure 2). For the first problem, the SGA converge rapidly toward a population composed only of the two best trade-offs. The population is stable, the oscillations are due to the sharing method. On the second problem, the SGA fails to find the best trade-offs. The most represented individual is not one of the two best trade-offs.

A DGA associated with the same niched Pareto ranking method, for both problems, converges toward the sets of best trade-offs (Figure 3). In the first problem, the final population contains the individual set

[1] To avoid confusion, we call classical schemata Positional Schemata, or P-schemata, as they link individuals having given values at given positions.

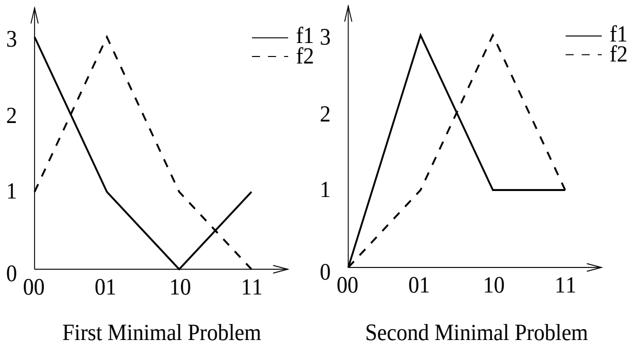

Figure 1: Instances of the first and the second minimal problem

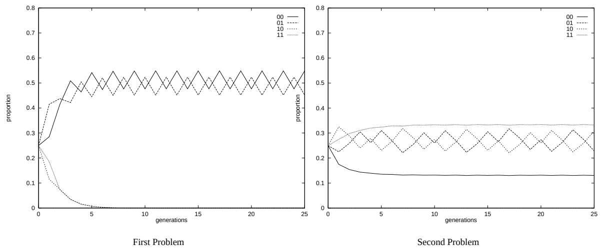

Figure 2: Pareto ranking SGA

$\{000, 001\}$, transliterated to $\{00, 01\}$. In the second problem, the final population contains the stable individual set $\{001, 101\}$, transliterated to $\{01, 10\}$.

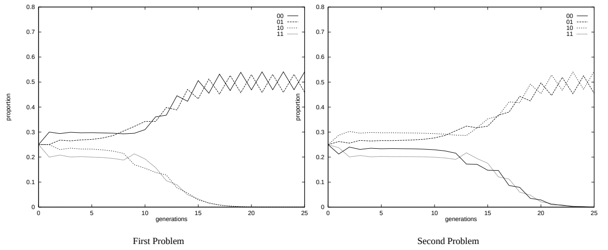

Figure 3: Pareto ranking DGA

4 Multiobjective Trap Problems

We construct Multiobjective Trap Problems (MOTPs) in order to obtain a kind of Pareto deception. As their single objective homologous, MOTPs are problems with two optima, a local one and a global one. The difference is, in the case of multiobjective problems, the optima is in fact a set of points. The figure 4 shows the two functions involved. They are defined on the normalized unitation of the chromosomes, i.e. the number of bits set to 1 divided by the total number of bits, and they return values between 0 and 1. These two functions have to be minimized. There are two Pareto sets. The right one, which lays between 0.9 and 1, is the local one: all of the points from it are dominated by those of the left one, which lays between 0 and 0.1. The problem is parameterized by two values. The first one, r, measure the importance of the global Pareto set with regard to the other. The second one, x_b, measure the relative importance of the basin of attraction of the two Pareto sets.

Firstly, we tested the two approaches, the SGA one and the DGA one, with several values for r and x_b in order to find their domain of efficiency on the plane $(r, x_b), r \in [0, 1], x_b \in [0.1, 0.9]$.

There is a particularity for $r = 0$. Indeed, for this value the two Pareto sets have to be considered equivalent. The DGA is able to find the two sets at the same time, as we will show it.

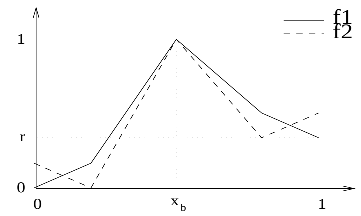

Figure 4: Multiobjective Trap Problems

Domains of Efficiency

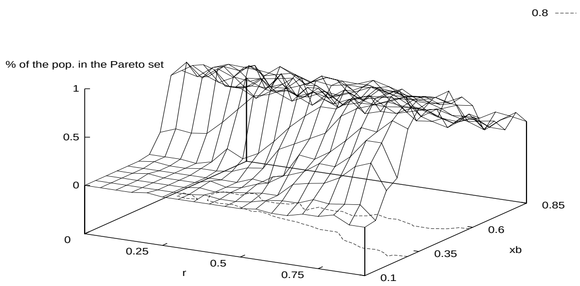

Figure 5: proportion of the population in the global Pareto set with SGA

The Figures 5 and 6 are the results of the experimentations. For each couple $(r, x_b), r \in [0, 1], x_b \in [0, 1]$, the value drawn is the average of the proportion of the population in the global Pareto set after 200 generations for 100 successive runs. On the figure 5, one can see that SGA is unable to find the good Pareto set for low values of r and x_b. This is due to the importance

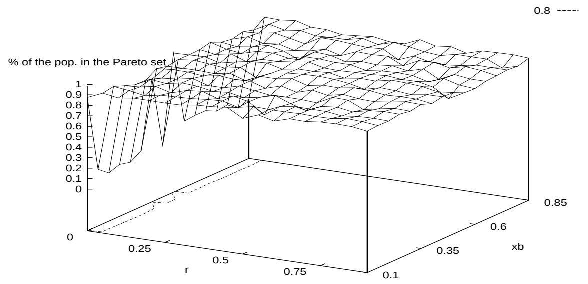

Figure 6: proportion of the population in the global Pareto set with DGA

of the basin of attraction of the bad set. Similarly, for x_b near 0.9, the proportion of the population in the Pareto set falls under 0.8. For this region, the basin of attraction of the global Pareto set is too important, and many individuals get "lost" in it.
For DGA, as we can see on figure 6, these remarks don't apply. On the totality of the plane $(r, x_b), r \in [0, 1], x_b \in [0.1, 0.9]$, the good set is found and a good proportion of the population is in it.

Disjoint Pareto sets

The case $r = 0$ is interesting as the two Pareto sets are then equivalent. In fact, there is a unique Pareto set, composed of two disjoint subsets. These two subsets are disjoint in the sens that each element of a subset is at least at a Hamming distance superior to 1 away from each element of the other subset.
A SGA, as shown in figure 7, is able to converge in a single subset. As in MMOP, it is unable to make chromosomes from incompatible schemata coexisted in the same population. So, after a while, the entire population goes to one of the subsets. A DGA, thanks to R-schemata, is able to do it. As we can observe it in the figure 8, the population firstly converge to the first subset. After a while, a part of the population goes in the second subset. Then, the two subsets are equally represented.

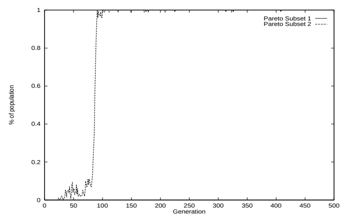

Figure 7: proportion of the population in each of the two Pareto subsets with SGA

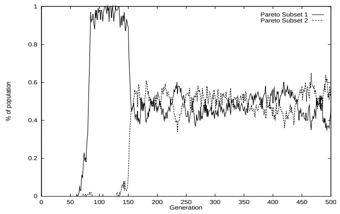

Figure 8: proportion of the population in each of the two Pareto subsets with DGA

5 Further Work

The experimentations proposed here are obviously *ad hoc* problems, constructed to demonstrate interesting characteristics of DGAs. Our future works will be to try these successful associations on more complex problems and on real engineering problems.

Acknowledgments

This work was partly supported by a Brite-Euram project (EC contract BRPR-CT96-0282).

References

[1] P. Collard and C. Escazut. Relational schemata: A way to improve the expressiveness of classifiers. In *Proceedings of the Sixth International Conference on Genetics Algorithms*, 1995.

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

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

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

[5] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and Their Application: Proceedings of the First International Conference on Genetic algorithms*, 1985.

[6] M. D. Vose. Modeling simple genetic algorithms. In *Foundations of Genetic Algorithms 2*, 92.