A Multiobjective Methodology for Evaluating Genetic Operators

Ricardo H. C. Takahashi, Joao Antonio Vasconcelos, Jaime A. Ramirez,
Laurent Krähenbühl

A Multiobjective Methodology for Evaluating Genetic Operators

Ricardo H. C. Takahashi, *Member, IEEE*, J. A. Vasconcelos, Jaime A. Ramírez, and L. Krahenbuhl

Abstract—This paper is concerned with the problem of evaluating genetic algorithm (GA) operator combinations. Each GA operator, like crossover or mutation, can be implemented according to several different formulations. This paper shows that: 1) the performances of different operators are not independent and 2) different merit figures for measuring a GA performance are conflicting. In order to account for this problem structure, a multiobjective analysis methodology is proposed. This methodology is employed for the evaluation of a new crossover operator (*real-biased crossover*) that is shown to bring a performance enhancement. A GA that was found by the proposed methodology is applied in an electromagnetic (EM) benchmark problem.

Index Terms—Genetic algorithm (GA), multiobjective performance evaluation.

I. INTRODUCTION

GENETIC algorithms (GAs) are reaching increasing importance in several fields of optimization. This class of algorithms is characterized by the "evolution" of a set of tentative solutions (*population*). The algorithm evolves with some stochastic search and combinatorial rules that, being applied to one population, lead to the next one. The existence of three *genetic operators* (rules) defines a GA: 1) a *crossover operator* that combines the information of two or more tentative solutions (*individuals*), generating another individual; 2) a *mutation operator* that, using the information contained in one individual, stochastically generates another one; and 3) a *selection operator* that, using the objective function evaluation of all individuals in the population, replicates some of them, and eliminates other ones, generating the next population. A GA may be built with these three rules only, or may contain other kinds of rules (*niche*, *local search*, for instance).

There is a large number of meaningful operators that can lead to suitable GAs. The basic GA operators can be implemented in a growing variety of ways (see, for instance, [4], [14]), and there is a growing variety of additional operators (see, for instance, [10], [12]), since the study of operator structures is now an active research area. In most of the cases, there is not any analytical justification for the choice of a specific operator structure; for some discussion on the difficulties associated to the task of analytically predicting the performance of a specific GA, see [9]. It

is known, however, that specific operator structure and operator parameter tuning should be employed for each class of problems, in order to get computational efficiency. The choices of each operator instance and its parameter values should be performed, therefore, on the basis of empirical previous evaluations [8], [14].

There is not, up to now, any integrative information source for guiding such a choice by the user. The usual kind of information that is available in literature fall into the categories. 1) *Tutorial*: Some common operator alternatives are presented and some values for their parameter settings are recommended, in the form of large ranges of values. See, for instance, [7], [6]. 2) *One-algorithm evaluation*: Follows the basic scheme of presenting a new algorithm and evaluating it against some algorithm that is considered to be "classical," or "usual," see, for instance, [10]. 3) *One-operator comparison*: Compares some alternatives of implementation for one operator, like selection [13]. 4) *Application-specific comparison*: Makes comparisons and recommendations that are specifically directed toward an application, see for instance [5].

Some concerns about what should be an "integrative information source" for choosing GA algorithms are: 1) the more recent and complex operator structures should be considered as alternatives to be analyzed, either against the conventional structures and one against the other one; 2) it should be also recognized that some problems may present strong sensibility to parameter tuning, what prevents the "usability" of large ranges of values as a tuning guide [5]; and 3) it should be recognized too that different operators are not independent, what means that, for instance, a crossover operator that presents the best performance with some mutation operator, can be outperformed by another crossover operator when the mutation operator is changed [2]. The choice of suitable GAs for specific applications is a complex task that should be structured in a systematic methodology, in order to: a) allow the usage of the most recent knowledge for the construction of efficient problem-specific GAs and b) keep the effort for reaching efficient algorithms in a feasible level. This paper discusses this problem and proposes a systematic methodology for approaching it.

II. EVALUATION METHODOLOGY

The problems to be approached here are defined as follows:
Definition 1: **Algorithm Evaluation Problem (AEP)** Given a class of problems to be dealt with and a set of genetic operators, find the best algorithms, considering both the criteria of maximum convergence rate and minimal convergence failure. □

As the AEP is defined in terms of two objectives, its solution is defined as a solution of a multiobjective problem, and has the form of a *Pareto-set*. This set is defined with the concept of *dominance*: a solution is said to be dominated if it is worse than another solution in at least one objective, while not being better than that solution in any other objective [3]. The Pareto-set is the set that does not contain any dominated solution. A formal definition is given in (2).

Definition 2: **New Algorithm Evaluation Problem (NAEP)** Given a new algorithm and a class of problems, compare it with the AEP best algorithms for that class, assigning to the new algorithm the status of an *enhanced solution* or of a *dominated solution*, possibly updating the set of "best solutions." □

The new algorithm, in the NAEP, must be compared only with the *Pareto-set* of algorithms, that was found with AEP. This means that the comparison database may discard most of alternatives that were considered in the former problem, and keep only the Pareto-set ones. After the NAEP, three things can occur. 1) The new algorithm is not better than the "usual" ones. It is discarded, and the Pareto-set is kept with the same former composition. 2) The new algorithm reveals to be nondominated, but it also does not dominate any "usual" algorithm that was already in the Pareto-set. In this case, the new algorithm is included in the Pareto-set, that grows. 3) The new algorithm dominates one or more solutions that were in the Pareto-set. It is included in the Pareto-set, and the dominated solutions are excluded.

The computational effort in NAEP is associated only with the evaluation of the test functions performed by the new algorithm. The effort associated with the construction of the Pareto-set in AEP is implicitly reused.

The following subsections discuss in detail the steps of the proposed methodology.

A. Selection of Representative Analytical Functions

The class of problems of interest is possibly constituted of functions that are not expressed in the form of analytical functions but, instead, are given by simulation models that are hard to be evaluated. This is the case, for instance, of electromagnetic (EM) device models [11]. The problem of evaluation of the usability of GAs in a specific class of problems, however, does not depend on actually using one function of the class for evaluating the algorithm, but only on using some function that keeps some fundamental characteristics of the actual functions [8]. With this procedure, a function that is fast to evaluate can be used, which makes feasible executing a large number of test runs for algorithm evaluation purposes. The work [8] presents procedures for tailoring analytical functions with some specified properties. Other possible way for building analytical models that present the properties of more complex systems is via an approximation technique [11].

In the present paper, the *Rotated Rastrigin function* is employed, in order to present the methodology proposed here

$$f(x) = x^T A^T A x - 10 [1 \; \cdots \; 1] \cos(2\pi A x) \quad (1)$$

in which $f(\cdot): \mathbb{R}^n \mapsto \mathbb{R}$, $A \in \mathbb{R}^{n \times n}$ is positive definite and the remaining variables are such that the dimensions are com-

TABLE I
USUAL OPERATOR ALTERNATIVES

ssga factor	1	1
	2	0.8
	3	0.5
	4	0
mutation	1	each bit
	2	bit per variable
	3	bit per individual
crossover	1	one-point
	2	two-point
	3	one point per variable
	4	uniform
selection	1	roulette
	2	deterministic sampling
	3	stochastic remaining s.
elitism	1	simple
	2	global
population size	1	30
	2	100
	3	300

patible. This is a simple prototype of multimodal functions with large-scale tendencies and coupled coordinates.

B. Database of GAs

A set of operator alternatives is defined in Table I. The alternatives are essentially the ones described in [14], Each version of GA is tested with all the functions of the representative set, with two merit figures: 1) the mean number of function evaluations for finding the global minimum and 2) the fraction of algorithm executions that finds the global minimum. Each such merit figure is evaluated for a number (say, 100) of algorithm executions.

There are 864 different instances of GAs as combinations of these parameters. The merit figures are determined for each algorithm and each representative function. A database is composed, with the structure

i_1	i_2	$\cdots$	i_k	f	m_1	m_2

The indexes $i_1, \ldots, i_k$ point to the different alternatives of the k operator alternatives that are under study. Index f identifies a "representative function," and m_1 and m_2 are the resulting merit figures that come from the application of the algorithm defined by $i_1, \ldots, i_m$ to its optimization. In the case studied here, index i_2, for instance, means the mutation operator, and $i_2 = 3$ means "one bit per individual" mutation.

Defining m_1 and m_2 for all instances of f as the objective functions of a multiobjective analysis problem, and $I = [i_1, \ldots, i_n]$ the vector of indices that identifies an instance of GA, the database is "pruned" according to the law

$$\mathcal{X}^* = \{I^* | \; \not\exists I \neq I^* \text{ such that } (m_1, f)(I) < (m_1, f)(I^*) \text{ and } (m_2, f)(I) < (m_2, f)(I^*)\}. \quad (2)$$

After this procedure, the resulting *Pareto-set* database $\mathcal{X}^*$ becomes much smaller than the initial database. In this database, only the *nondominated* algorithms are kept.

The rotated Rastrigin function was tested with all 864 GA instances. The resulting merit figures are plotted in Fig. 1. This figure shows that, among these algorithms that employ "usual" operators, there are relatively few ones that can be considered
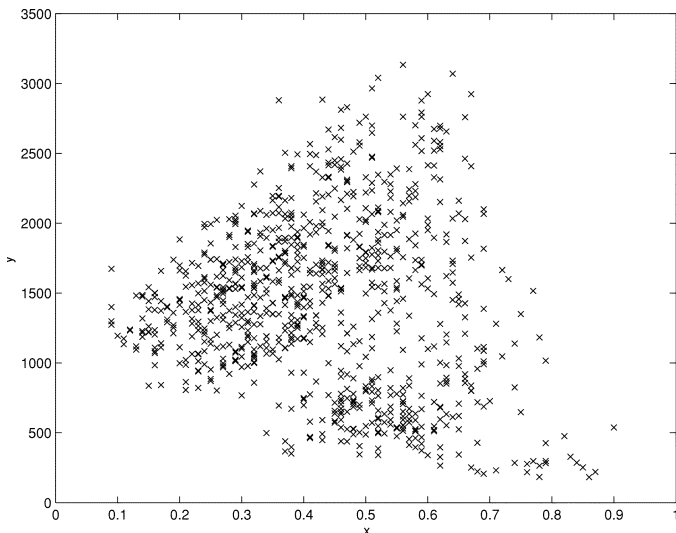
Fig. 1. Plot of the merit figures for the algorithms composed of "usual" operator combinations: number of function evaluations up to convergence (vertical) versus fraction of nonconvergence runs (horizontal). The function is the Rotated Rastrigin.

Fig. 2. The Pareto-sets of the "usual operators" only (x) and of their combination including the new operator "real-biased crossover" (o). Merit figures: number of function evaluations up to convergence (vertical) versus fraction of nonconvergence runs (horizontal). The function is the Rotated Rastrigin.
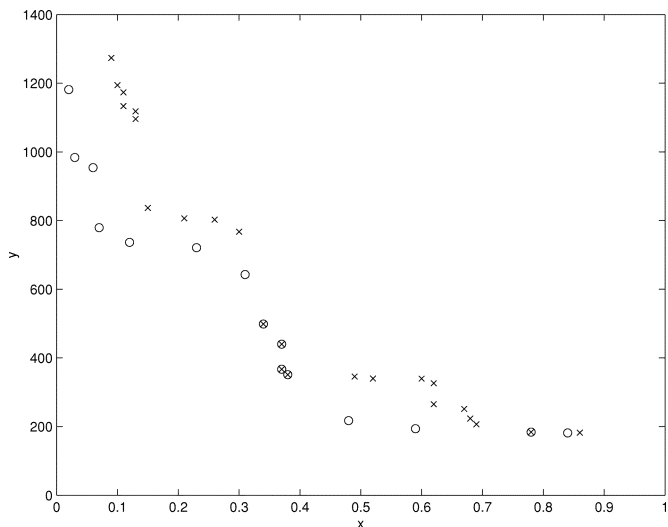
"good," in the sense that, for instance, they fail in less than 20% of the runs, and they need less than 1000 function evaluations for finding the optimum of the function. On the other hand, there is a large number of algorithms that can be very bad, in the sense, for instance, that they need more than 2000 function evaluations, or fail in more than 50% of the runs. This result shows that an algorithm being "usual" does not mean that it can be considered to be a good comparison standard for new algorithms. The Pareto-set is extracted from this data (Fig. 2). This set is constituted of only 25 solutions that are "nondominated." These 25 solutions are sufficient for the purpose of comparison of any new algorithm, with the same test function.

C. New Operator and New Algorithm Evaluation Procedures

For testing new *algorithms* (i.e., a specific combination of particular operators), the procedure to be followed is somewhat

obvious: evaluate it with the same test function, computing the two merit figures. After that, reevaluate the Pareto-set, using (2) to include the information about the new algorithm.

For testing new *operators*, the question becomes subtle: an operator (for instance, crossover) must be combined with other operators (mutation and selection, at least), in order to give rise to an executable GA. Is it a reasonable heuristic for performing the test choosing "good" formerly known algorithms (that belong to the Pareto-set of *usual operators*), and replacing the operator to be tested in them? This would mean, in some extent, that the operators are "independent" one to each other. Or it is necessary to test the new operator within other operator combinations that did not belong to the former Pareto-set? This question is answered below, in the context of the evaluation of a new crossover algorithm that is being presented here: the *real-biased crossover*. This new operator is described in the Appendix.

The *real biased crossover* operator was tested in all combinations with the "usual operator set," which means 240 combinations. The Pareto-set extracted from this data is shown in Fig. 2, superimposed to the Pareto-set that was obtained with the "usual" operators only. The Pareto-set is now constituted of only 15 solutions that are "nondominated." Comparing now the two Pareto sets in Fig. 2: The most reliable algorithm among the "usual ones" fails in 9% of the runs and needs about 1270 function evaluations for reaching the function optimum, while in the "real biased crossover" set, it fails only 2% of the runs, and needs less than 1200 function evaluations for reaching the function optimum. There is one algorithm in the "real biased crossover" set that needs less than 800 function evaluations, and fails less than 10% (this algorithm is labeled GA-1, for the purpose of performing further numerical evaluations with it). The best algorithm that needs less than 800 function evaluations, in the "usual algorithms" set fails about 30%. The *real biased crossover* operator has proved to be an enhancement in this case, leading to the more reliable and to the faster algorithms, and to most of the intermediate Pareto solutions (10 of the 15 ones).

Unfortunately, the answer to the question of if operators are independent is: **no**. Inspecting the ten new Pareto algorithms that were constituted with the *real-biased crossover* operator, it is found that only two of them could be generated by the replacement of the crossover operator in the initial "usual" 25 algorithms. Almost all the new Pareto algorithms, in this case, use operator combinations that did not lead to former Pareto solutions. This does not mean that an operator cannot be evaluated: the data shown earlier clearly shows that the *real-biased crossover* operator constitutes an enhancement in relation to formerly known alternatives (at least for the class of problems that share the features of the Rotated Rastrigin function). However, the task of "operator evaluation" should be more carefully stated than the standard procedure that is reported in most of the papers on the subject, that implicitly rely on operator independence.

III. RESULTS IN AN EM PROBLEM

The GA-1, selected earlier, has the following selection of parameters, in the order of Table I: [315 223]. This algorithm has been tested in TEAM Workshop problem 22 benchmark with

three variables (for the definition of the problem, see [1]). The objective function to be evaluated is defined as

$$F(x) = \frac{B_{\text{stray}}^2}{B_{\text{norm}}^2} + \frac{|\text{Energy} - \text{E}_{\text{ref}}|}{E_{\text{ref}}} + 5\left(|B|_{\text{pred}} - |B|_{\text{calc}}\right)^2 \tag{3}$$

where $E_{\text{ref}} = 180$ MJ and $B_{\text{stray}}^2 = \sum_{i=1}^{22} |B_{\text{stray}_i}^2|$ with $|J| = (-6.4|B|_{\text{pred}} + 54.0)$ A/mm^2. The algorithm total number of evaluations has been fixed in 2400 (three times the mean number of necessary evaluations for convergence of GA-1 in the Rotated Rastrigin function). The following optimization parameters that were obtained $[r_2 = 3.05$ m, $h_2 = 0.492$ m, $d_2 = 0.400$ m$]$. The resulting constraint values are $E = 179.74$ MJ, $B_{\text{stray}} = 0.936$ μT, and the objective function becomes $F = 0.122$.

IV. CONCLUSION

A multiobjective analysis methodology has been proposed for evaluating GAs. This methodology was shown to be suitable for dealing with the fundamental problem of aggregating the knowledge that is already available in the field of GA theory, leading to answers for questions such as: 1) What are the "good" GAs for dealing with some class of problems? 2) What is the tradeoff among these GA alternatives? The question of what are the best *operators* has shown to be more intricate, since the operators have been shown to be performance-dependent one to another. The multiobjective methodology has shown to be an effective procedure for dealing with this question too.

A new crossover operator (the *real-biased crossover*) has been presented too. The new operator introduces some directional search properties that are not available in conventional GA operators.

The results obtained in the EM problem confirm the prediction that the algorithm GA-1 would converge to a good solution in a reasonable number of function evaluations.

APPENDIX
REAL-BIASED CROSSOVER OPERATOR

Define the parameters:

- p_{pol}: the probability of the crossover being "biased," $0 \leq p_{\text{pol}} \leq 1$;
- e: the "extrapolation factor," $-0.5 < e \leq 1$;

In the case that was studied in this paper, the values were chosen as $p_{\text{pol}} = 0.3$ and $e = 0.1$.

The real-biased crossover operator is defined as

1) Take two individuals, x_1 and x_2 from the population. If they are in binary code, they must be put in real coding.
2) Evaluate the objective function values $J(x_1)$ and $J(x_2)$. Suppose a minimization problem and, without loss of generality, suppose $J(x_1) \leq J(x_2)$.
3) Choose $-e \leq \alpha_1 \leq (1 + e)$ with uniform probability.
4) Decide if the crossover is "biased" or not, with the probability factor p_{pol}. If no, choose $-e \leq \alpha_2 \leq (1 + e)$ with uniform probability and go to step 7). If yes, go to step 5).
5) Choose $0 \leq \beta_1 \leq 1$ and $0 \leq \beta_2 \leq 1$, both with uniform probability.
6) Make $\alpha_2 = (1 + 2e)\beta_1\beta_2 - e$.
7) Create two new individuals, x_1^g and x_2^g, according to the law $x_i^g = \alpha_i x_1 + (1 - \alpha_i)x_2$.

The real-biased crossover operator as defined earlier essentially generates new individuals that are located over the line segment that goes from one point to the other one, possibly with an "extrapolation" outside this segment, over the same line, up to the factor e. At least one individual is generated over this segment with uniform probability. The other one, if the crossover is biased, is generated over this segment with quadratic probability distribution, with the greater probability of being generated near the "best" of the two parents.

The "biased" operation mimics a tendency search (like a gradient information) that is not performed by any conventional GA operator, while keeping the GA advantage of evaluating the objective function only (without any derivative computation). In the case of parent individuals that are near one to another, a subgradient-like step is performed. This accelerates local convergence to the optimum. In the case of individuals that are distant, the search will be interpreted as a "long-range" tendency information search that has no counterpart in the deterministic algorithms.

REFERENCES

[1] Team Benchmark Problem 22 Definition (2002, May). [Online]. Available: http://www-igte.tugraz.ac.at/team/index.htm.
[2] E. Belmont-Moreno, "The role of mutation and population size in genetic algorithms applied to physics problems," *Int. J. Modern Phys. C*, vol. 12, no. 9, pp. 1345–1355, 2001.
[3] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making—Theory and Methodology*. New York: North-Holland, 1983.
[4] D. H. Choi and S. Y. Oh, "A new mutation rule for evolutionary programming motivated from backpropagation learning," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 188–190, July 2000.
[5] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 153–158, Feb. 1997.
[6] J. M. Johnson and Y. Ramat-Semii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas Propagat. Mag.*, vol. 39, pp. 7–25, Aug. 1997.
[7] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: Concepts and applications," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 519–534, 1996.
[8] Z. Michalewicz, K. Deb, M. Schmidt, and T. Stidsen, "Test-case generator for nonlinear continuous parameter optimization techniques," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 197–215, Sept. 2000.
[9] B. Naudts and L. Kallel, "A comparison of predictive measures of problem difficulty in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 1–15, Apr. 2000.
[10] J. C. Potts, T. D. Giddens, and S. B. Yadav, "The development and evaluation of an improved genetic algorithm based on migration and artificial selection," *IEEE Trans. Syst., Man Cybern.*, vol. 24, pp. 73–86, Jan. 1994.
[11] K. Rashid, J. A. Ramirez, and E. M. Freeman, "A general approach for extracting sensitivity analysis from a neuro-fuzzy model," *IEEE Trans. Magn.*, vol. 36, pp. 1066–1070, July 2000.
[12] B. Sareni and L. Krahenbuhl, "Fitness sharing and niching methods revisited," *IEEE Trans. Evol. Comput.*, vol. 2, pp. 97–106, Sept. 1998.
[13] T. Schell and S. Wegenkittl, "Looking beyond selection probabilities: Adaptation of the χ^2 measure for the performance analysis of selection methods in GAs," *Evol. Comput.*, vol. 9, no. 2, pp. 243–256, 2001.
[14] J. A. Vasconcelos, J. A. Ramirez, R. H. C. Takahashi, and R. R. Saldanha, "Improvements in genetic algorithms," *IEEE Trans. Magn.*, vol. 37, pp. 3414–3417, Sept. 2001.