This document is published in:

"Omatu, G. et al. (eds.) (2012) *Distributed Computing and Artificial Intelligence*: *9th International Conference*, (Advances in Intelligent and Soft Computing, 151) Springer, pp. 725- 732.
Doi: http://dx.doi.org/10.1007/978-3-642-28765-7_87**"**

Mutagenesis as a Diversity Enhancer and Preserver in Evolution Strategies

José L. Guerrero, Alfonso Gómez-Jordana, Antonio Berlanga and José M. Molina

Abstract Mutagenesis is a process which forces the coverage of certain zones of the search space during the generations of an evolution strategy, by keeping track of the covered ranges for the different variables in the so called gene matrix. Originally introduced as an artifact to control the automated stopping criterion in a memetic algorithm, ESLAT, it also improved the exploration capabilities of the algorithm, even though this was considered a secondary matter and not properly analyzed or tested. This work focuses on this diversity enhancement, redefining mutagenesis to increase this characteristic, measuring this improvement over a set of twenty-seven unconstrained optimization functions to provide statistically significant results.

1 Introduction

Evolutionary computation [2] arose as a powerful technique to deal with optimization and search problems, particularly evolution strategies [13], focused on real value representations, which have been successfully tested on a wide variety of problems, both theoretical and practical in nature. The increase in the computational resources of computers and the increasing number of parallel implementations [4] have lead this growth, making them more appealing for practitioners focused on solving particular problems, rather than theoretical research of the algorithms themselves. There are, however, a number of issues for these applications.

Local optima constitute a drawback for evolutionary algorithms, since they do not provide (as most metaheuristics [16]) a measurement of the proximity of the solutions found to global optima, performing a best-effort approach. Early conver-

José L. Guerrero, e-mail: jguerrer@inf.uc3m.es. Alfonso Gómez-Jordana, e-mail: alfonso.gomez-jordana@alumnos.uc3m.es. Antonio Berlanga, e-mail: aberlan@ia.uc3m.es. José M. Molina e-mail: molina@ia.uc3m.es ·
University Carlos III of Madrid, Computer Science Department, Group of Applied Artificial Intelligence. Avda. Universidad Carlos III, 22, Colmenarejo, Spain

gence arises as a concern regarding this topic, being closely related to the diversity preservation in the population as the algorithm progresses. Many approaches have been proposed to deal with this issue, from the restriction of certain operator applications (such as the incest prevention proposed in [6]) or multi-objective approaches [5] where the diversity of the population is treated as an additional objective function [17]. General stopping criteria are also a concern for practitioners using evolutionary techniques, which is in fact shared by many different iterative processes [1], but the stochastic nature of evolutionary computation makes it probably more important and, at the same time, harder to solve. Finally, evolutionary algorithms tend to favor the exploratory nature of the search process, leading to a slow convergence towards the minimum. This handicap has been faced with memetic algorithms [12], which combine evolutionary techniques with local search [10] in an attempt to obtain a good exploration of the search space with a fast information exploitation once the most interesting zones have been determined.

ESLAT (Evolution Strategies Learned with Automated Termination criteria) technique was introduced to deal with some of those issues. ESLAT is a memetic algorithm which combined the evolutionary cycles of an evolution strategy with two different local search procedures which were applied sequentially: the *Broyden-Fletcher-Goldfarb-Shanno* Quasi-Newton method (BFGS) with a cubic line search procedure [3] and Nelder Mead's algorithm [14], based on Kelley's modification [11]. Along with the local search techniques, it introduced an artifact used to control the stopping generation: the *gene matrix*, which tracked the zones of the search space covered and forced the exploration of those which had not been reached by the main search cycle, and also determining the generation at which the algorithm should be stopped according to that search space coverage.

ESLAT presented a series of difficulties in its results, which were faced in R-ESLAT (Robust ESLAT) [8]. These faced difficulties included control over the search space, the configuration of the local search techniques used and the stopping criterion used which led to a comparison with CMA-ES algorithm [9] including promising results in terms of solution quality. However, the basis of the proposal was the increased exploratory capabilities introduced by the gene matrix, which were not individually tested, only as a part of the overall algorithm performance.

This work analyzes the gene matrix as a diversity preservation technique in evolution strategies, modifying its original behavior according to this new role and testing its performance against a canonical evolution strategy. This test is based on a set of twenty-seven unconstrained optimization functions with different characteristics, in order to highlight the statistical significance of the obtained results.

The paper is structured according to the following sections: the second section will introduce mutagenesis and gene matrix concepts, along with the paper proposal for their definition and use. The third section will present the experimental setup used and the obtained results, along with their analysis. Finally the fourth section will present the conclusions obtained from the available results and the future lines of the work.

2 Gene matrix and diversity preservation

The *gene matrix* (GM) is responsible of tracking the exploration process and keeping the diversity in the population. It is composed of n by m elements, where n is the number of genes in the chromosome and m is the number of sub-ranges in which the search space of that chromosome is divided. This matrix is initialized with zeros, and those zeros are updated to ones as elements with genes covering the different sub-ranges are found in the different populations as the evolution progresses. Figure 1 shows an example of a GM with two variables.

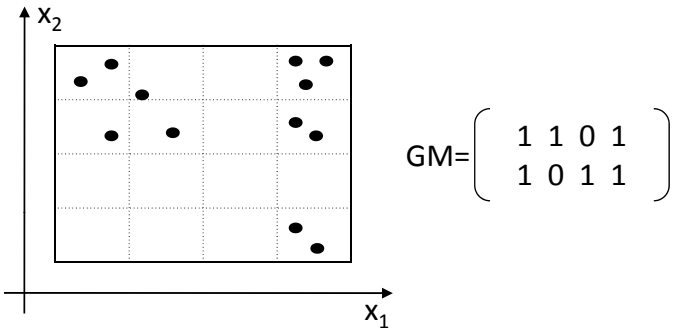

Fig. 1: Gene Matrix example

The *GM* is used, therefore, as a measurement of the depth in the exploration process. In order to use it to keep the diversity in the population as well, the *mutagenesis* operator is introduced. At the end of every generation, the mutagenesis operator chooses the N_w worst individuals which have survived to the next generation and changes the values of one of their genes in order to cover new zones of the search space (according to the information in the GM). Specifically, for each of the N_w worst individuals in the population, one of the sub-ranges containing a zero value in the *GM* is selected randomly, that GM position updated to a one, and the value in the correspondent gene of the individual is updated according to a random value within the sub-rage boundaries. Figure 2 presents an example of this process.

According to its original definition, once the gene matrix had been completely filled with ones, the mutagenesis procedure was stopped, and the algorithm stopped after a certain number generations (originally configured to be the problem dimensionality). This stopped the diversity increase once the search space had been covered, and focused the gene matrix use on a simple mechanism to control the automated stopping criterion. As a diversity enhancer, however, this behavior is not acceptable. Another issue concerning the gene matrix was the number of subranges required. The number of these subranges was fixed on 30. However, preliminary tests for this paper showed that different configurations in the number of subranges may improve the results for different problem instances.

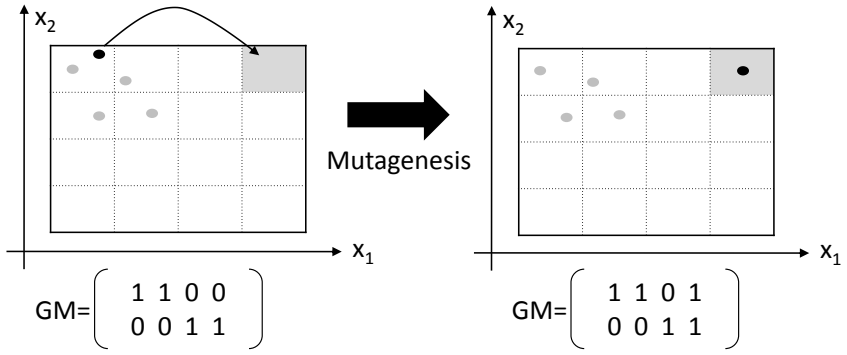

Fig. 2: Mutagenesis example

The novel gene matrix proposal is focused on diversity enhancement, rather than its application as a termination criterion. To do so, an initial number of subranges is set a-priori. Once the gene matrix is filled with ones at a certain generation, it is *restarted*, reinitializing it with zeros and updating it with the individuals in the population which caused this reinitialization. Every time the gene matrix is reinitialized, its number of contained subranges is doubled. This mechanism achieves a constant diversity enhancement and also a more thorough coverage of the search space as the algorithm progresses, depending on the dimensionality of the problem faced (since higher dimmensionalities will imply a higher number of subranges to be covered).

The mutagenesis procedure has also been reviewed. As previously explained, it originally introduced a certain number of modifications on the worst individuals of the population, changing concrete values from the chromosome to unexplored subranges of the chosen gene. This behavior may not introduce enough diversity in a population heavily dominated by the best individual, so an additional probability is added to the algorithm configuration: p_{rm}, random mutagenesis probability. According to this probability, mutagenesis may generate a random individual covering the chosen subrange instead of modifying just one gene from one of the worst individuals in the population.

Additional controls have also been added to mutagenesis. If an individual has covered a new subrange in current generation, it is never changed any further by the mutagenesis procedure, regardless of its rank. This allows the new information introduced during the evolutionary cycle to survive at least one generation, in order to give the new individual the chance to procreate and mutate before any directed change is applied to it. This also implies a change in the mutagenesis configuration. Instead of N_w changed individuals, the user configures a more versatile N_c parameter, establishing the number of new subranges covered each generation. If the evolutionary cycle covers the required number of changes, no mutagenesis is applied. In other case, the worst individuals (as many as required in order to cope with the desired N_c changes) are picked to go through the mutagenesis procedure.

Finally, the stopping criterion used in R-ESLAT implied that the best fitness was repeated over a certain window of generations. This exact repetition may be too strict for a stopping criterion, since very small changes in fitness values (which might even be affected by the representation precision) would lead to a continuation in the evolutionary algorithm once the search process had stagnated regarding all practical purposes. For these reasons this exact comparison was changed to the comparison quotient presented in equation 1, which provides a more flexible mechanism to control the relevance of the changes.

$$\frac{previous_{best} - current_{best}}{previous_{best}} \leq Improvement_{factor} \tag{1}$$

3 Experimental validation

For the experimental validation of the proposed technique, twenty-seven different functions have been used, with different characteristics: dimensionality, presence of local optima, search space range... Table 3 shows a brief description on them, providing their name, dimensionality and their search space boundaries. Several parameters (according to their description included in the previous section) have to be established for the proposed technique, which are presented in table 1. As included in that table, four different population sizes are used to cover the comparison of the two different techniques. The complete results for population size five are presented in table 3. Following [7], the individual comparison for the different test functions is performed according to parametric and non-parametric tests. The normality test used is the Shapiro-Wilk test [15], the parametric test is Student's t-test and the non-parametric test is Wilcoxon signed-rank test [18]. The statistical best results are provided according to the t-test if the data follows a normal distribution and according to the non-parametric in other case. Fifty iterations have been run in order to establish the statistical significance of the results.

To test the final performance comparison, a Wilcoxon rank-sum test is carried out over the mean results for the twenty-seven functions and the four considered population sizes. The p-value obtained is 0.0275, which implies that with a significance level as low as 3% (lower than the usual 5% considered for these tests) the proposed gene matrix diversity enhancer allows evolution strategies to perform better.

Table 1: Experimental configuration

Parameter	Description	Value
μ	Population size	5, 10, 15, 30
$init_{sr}$	Initial subranges	10
min_{sr}	Minimum subranges covered per generation	μ/5
p_{rm}	random mutagenesis probability	0.5
I_f	Improvement factor	1E-05

Table 2: Results comparison for the different considered population sizes

Population size	Statistical Best	Statistical Worst	Best
5	7	3	19
10	7	8	14
15	10	6	15
30	3	10	13

Analyzing the individual results, the effectiveness of the diversity enhancement is, in general, more representative at lower population sizes (where the risk of falling into local optima is higher and the exploration capabilities are reduced) but, at the same time, since the number of required changes per generation are configured as a certain percentage of the population, the use of the gene matrix is more accused on higher population sizes. The balance between these two factors determines the effectiveness of the mutagenesis changes. This is reflected in the variable number of significant best and worst results obtained for the different population sizes, which shows that the mentioned effectiveness does not only depend on the additional exploration required.

Finally, regarding the individual analysis of the results for the different test functions, it must be noted that the non-parametric tests do not seem to be able to properly measure some behavior differences (due to their zero median null hypotheses). This can be seen, for instance, in table 3, function f14, where, even though the mean value obtained by the evolution strategy using mutagenesis is several orders of magnitude bellow the one obtained without it, the Wilcoxon test does not determine it to be the best. This points to the requirement of mean based statistical tests not requiring normality distribution over their measures to perform quality comparisons between algorithms.

4 Conclusions

Gene matrix and mutagenesis were originally presented as part of the ESLAT memetic algorithm. They were used as guidance for the automated stopping criterion, even though they also increased the exploration capabilities of the evolution strategy included. This work isolates these artifacts and focuses on their diversity enhancement, redefining the processes in order to maximize these characteristics, and tests the results comparing them to the performance of canonical evolution strategies. The obtained results show that the exploration improvements lead the algorithm to an overall better performance, with a different impact regarding the population size and the percentage of the population which goes through mutagenesis processing. For a set of twenty-seven unconstrained optimization functions, the algorithm is statistically better considering four different population sizes and fifty iterations, providing a fair statistical significance. The testing process also highlights

the requirement for mean centered statistical tests, since non-parametric alternatives may not be able to measure performance differences under certain specific circumstances due to their median analysis. Future lines include the redefinition of the original memetic algorithm, the inclusion of these techniques in different algorithms and the study of novel performance comparison measures to cover the possible lacks of non-parametric statistical tests.

Acknowledgements this work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. M. Arioli, I. Duff, and D. Ruiz. Stopping criteria for iterative solvers. *SIAM Journal on Matrix Analysis and Applications*, 13:138, 1992.
2. T. Back, D.B. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation*. IOP Publishing Ltd., 1997.
3. CG Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76, 1970.
4. E. Cantu-Paz. *Efficient and accurate parallel genetic algorithms*. Springer Netherlands, 2000.
5. C.A.C. Coello, G.B. Lamont, and D.A. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer-Verlag New York Inc, 2007.
6. L.J. Eshelman and J.D. Schaffer. Preventing premature convergence in genetic algorithms by preventing incest. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 115–122. Morgan Kaufmann Publishers, 1991.
7. S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization. *Journal of Heuristics*, 15(6):617–644, 2009.
8. J.L. Guerrero, A. Berlanga, and J.M. Molina. A robust memetic algorithm with self-stopping capabilities. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pages 113–114. ACM, 2011.
9. N. Hansen, S.D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.
10. H.H. Hoos and T. Stützle. *Stochastic local search: Foundations and applications*. Morgan Kaufmann, 2005.
11. CT Kelley. Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, 10(1):43–55, 2000.
12. N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *Evolutionary Computation, IEEE Transactions on*, 9(5):474–488, 2005.
13. Z. Michalewicz and Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*, volume 19. Springer-verlag Berlin, 1992.
14. J.A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308, 1965.
15. S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.
16. E.G. Talbi. *Metaheuristics: From design to implementation*. Wiley Online Library, 2009.
17. A. Toffolo and E. Benini. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation*, 11(2):151–167, 2003.
18. F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Table 3: Test-set functions overview and results for population size five

Function information					Gene Matrix results		Canonical results		Statistical tests results			Techniques comparison	
id	Name	n	min. bound.	max. bound.	mean	std	mean	std	normal	ttest	wilcoxon	statistical best	best
f1	Ackley	30	-15	30	1,41E+00	1,45E+00	1,54E+00	1,95E+00	1	0	0	-	Gene Matrix
f2	Beale	2	-4,5	4,5	3,05E-02	1,51E-01	3,04E-01	4,13E-01	1	1	1	Gene Matrix	Gene Matrix
f3	Bohachecsky	2	-100	100	6,12E-02	1,78E-01	8,31E-03	5,85E-02	1	1	1	Canonical	Canonical
f4	Booth	2	-10	10	8,19E-08	2,62E-07	1,53E-07	5,86E-07	1	0	0	-	Gene Matrix
f5	Branin	2	-5	15	4,58E-07	4,44E-07	4,79E-02	3,26E-01	1	0	0	-	Gene Matrix
f6	Colville	4	-10	10	6,04E-01	1,39E+00	1,02E+00	2,87E+00	1	0	0	-	Gene Matrix
f7	Dixon-Price	30	-10	10	2,45E+00	1,98E+00	1,80E+00	1,46E+00	1	1	0	-	Canonical
f8	Easom	2	-100	100	8,83E-01	3,21E-01	9,50E-01	1,78E-01	1	0	0	-	Gene Matrix
f9	Goldstein-Price	2	-2	2	2,16E+00	7,40E+00	9,92E+00	2,66E+01	1	0	0	-	Gene Matrix
f10	Griewank	30	-600	600	1,21E-01	2,15E-01	1,29E-01	1,97E-01	1	0	0	-	Gene Matrix
f11	Hartmann	6	0	1	3,82E-02	5,62E-02	3,36E-02	5,39E-02	1	0	0	-	Canonical
f12	Hump	2	-5	5	1,63E-02	1,15E-01	6,03E-06	2,96E-05	1	0	0	-	Canonical
f13	Levy	30	-10	10	1,14E-01	3,50E-01	4,76E-01	1,31E+00	1	1	1	Gene Matrix	Gene Matrix
f14	Matyas	2	-10	10	5,17E-09	1,23E-08	5,86E-06	4,13E-05	1	0	0	-	Gene Matrix
f15	Michalewicz	10	0	pi	5,81E-01	3,35E-01	1,01E+00	5,01E-01	0	1	1	Gene Matrix	Gene Matrix
f16	Perm	30	-30	30	2,59E+85	4,57E+85	1,35E+86	1,56E+86	1	1	1	Gene Matrix	Gene Matrix
f17	Powell	28	-4	5	1,08E-02	1,97E-02	6,51E-03	1,11E-02	1	1	1	Canonical	Canonical
f18	Power Sum	4	0	4	4,05E-02	3,24E-02	1,24E-01	4,16E-01	1	0	0	-	Gene Matrix
f19	Rastrigin	30	-5,12	5,12	2,35E+01	1,20E+01	4,25E+01	1,86E+01	0	1	1	Gene Matrix	Gene Matrix
f20	Rosenbrock	30	-5	10	9,90E+01	1,66E+02	7,72E+01	1,21E+02	1	0	0	-	Canonical
f21	Schwefel	30	-500	500	7,69E+02	4,72E+02	2,35E+03	4,07E+02	0	1	1	Gene Matrix	Gene Matrix
f22	Shekel	4	0	10	5,54E+00	3,27E+00	5,28E+00	3,41E+00	0	0	0	-	Canonical
f23	Shubert	2	-10	10	2,72E-02	1,84E-01	2,65E+00	1,87E+01	1	0	0	-	Gene Matrix
f24	Sphere	30	-5,12	5,12	5,14E-05	2,01E-04	1,46E-05	5,58E-05	1	0	1	Canonical	Canonical
f25	Sum Squares	30	-10	10	2,84E-04	1,03E-03	1,53E-03	9,70E-03	1	0	0	-	Gene Matrix
f26	Trid	10	-100	100	6,00E-02	1,36E-01	1,07E-01	4,56E-01	1	0	1	Gene Matrix	Gene Matrix
f27	Zakharov	30	-5	10	1,97E+01	4,37E+01	2,25E+01	4,39E+01	1	0	0	-	Gene Matrix