

# A Robust Memetic Algorithm with Self-stopping Capabilities

José L. Guerrero, Antonio Berlanga and José M. Molina

Group of Applied Artificial Intelligence (GIAA)

Computer Science Department, University Carlos III of Madrid

Av. Universidad Carlos III 22, Colmenarejo 28270 Madrid, Spain

jguerrer@inf.uc3m.es, aberlan@ia.uc3m.es, molina@ia.uc3m.es

## ABSTRACT

Evolutionary algorithms exhibit some traditional handicaps: lack of a stopping criterion, slow convergence towards the minimum, etc. Memetic algorithms try to combine the best exploration qualities of population based approaches with the exploitation qualities of local search ones. The proposed solution in this work, Robust Evolutionary Strategy Learned with Automated Termination Criteria (R-ESLAT) uses a memetic approach, combining an evolutionary strategy with derivative-free local search methods, adding as well a termination criteria based on the population diversity, according to the principles of the original ESLAT algorithm. The original algorithm is analyzed and its features improved towards an increased robustness, comparing the results obtained with the Covariance Matrix Adaptation Evolutionary Strategy (CMAES).

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellan.;  
D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

## General Terms

Algorithms

## Keywords

Evolutionary Strategy, Memetic algorithm, Stopping criterion

## 1. INTRODUCTION

Memetic algorithms [6], combine local search procedures with population-based evolutionary processes under a general cultural evolution framework. The combination of the two sets of techniques theoretically allows them to apply the exploitation capabilities of local search techniques to a better coverage of the search space, attributed to population-based approaches. An additional feature related to these processes is the stopping criterion, closely related to the diversity in the population [8], a concern shared by many iterative processes [1]. Traditional approaches to this issue set an *a-priori* budget of affordable computational cost (which

may be expressed in terms of generations, function evaluations or even time) and stop the process once the budget has been fulfilled. However, establishing this budget can be a difficult process for real problems.

The purpose of this paper is to present the Robust Evolutionary Strategy Learned with Automatic Termination Criteria (R-ESLAT), a memetic algorithm with self-stopping capabilities, which performs a control over the population diversity and search space exploration, based on the original ESLAT algorithm [4]. The technique proposal will be based on the original algorithm analysis, specifying and modifying it according to the exploration and exploitation capabilities required in order to improve its robustness and results quality, finally comparing the obtained results to the Covariance Matrix Adaptation Evolutionary Strategy (CMAES) [3].

## 2. THE R-ESLAT ALGORITHM

R-ESLAT shares the general characteristics of the original algorithm: a memetic approach which combined an evolutionary algorithm with two different sequentially applied local search procedures: the first one based on Matlab's *fminunc* function implementation and the second one based on Kelley's modification [5] of the Nelder-Mead algorithm [7]. It introduces two control artifacts: the *gene matrix* (GM) and the *mutagenesis* operator. The *gene matrix* is responsible of tracking the exploration process and keeping the diversity in the population. It is composed of  $n$  by  $m$  elements, representing the number of variables and the number of sub-ranges each variable is divided into. It is initialized with zeros, and its values are updated whenever an individual is generated in the given sub-range. The *mutagenesis* operator chooses, at the end of each generation, the  $N_w$  worst individuals, changing one of their genes' values to fill a zero in the *gene matrix*. The proposed termination criterion was triggered once the *gene matrix* was full, allowing an additional number of generations equal to the problem's dimensionality.

R-ESLAT deals with the difficulties of the original ESLAT algorithm: control over the search space, definition and configuration of the intensification processes and robustness of the stopping criterion. The violations of constraints over the search space due to mutation procedures are corrected according to the difference between the maximum and minimum value obtained and the boundaries values, while those caused by the intensification processes involve a *death-penalty* over the individual. The *fminunc* function may apply different processes. The one chosen for R-ESLAT is the *Broyden-Fletcher-Goldfarb-Shanno* Quasi-

Newton method (BFGS) with a cubic line search procedure [2]. The setup of its function and variable tolerances, along with the ones for Nelder-Mead method, is set to  $1E^{-30}$ .

The intensification process was determined to be applied to the *best* (best fitness) and *most promising* (greatest difference with its parents) children in the population. Originally, intensification on the *best* child was only applied when it improved the best individual in the population. R-ESLAT always applies this intensification to the child with the best fitness, providing it with the chance of improving the best individual after this intensification. Nelder-Mead method also requires  $n + 1$  individuals to be run over an  $n$  dimensional problem. The remaining  $n$  individuals are chosen as the fittest from the children pool. Finally, the stopping criterion is improved with the use of a generational window of size  $\max(n, 30)$ . This window tracks the fitness value of the best individual in the population. If tracked value improves, the algorithm is not stopped. This mechanism prevents the early stops registered in the original algorithm.

The configuration parameters for the algorithm can be looked up in [4], along with the generational window and the function and variable tolerances for the intensification processes proposed in this work. The test-set used for the comparison contains 27 different functions with a wide set of different characteristics, regarding separability, dimensionality, search space, presence of local minima, etc. The details of the used dataset may be found online<sup>1</sup>. The dimensionality used for those functions which allow the choice of this parameter is 30, and the fitness values presented are always the difference to their known optima. Regarding CMAES, the implementation used is Matlab's version 3.54. Its main setup parameters are the search space boundaries (set according to the concrete problem characteristics), the initial individual (chosen as a random value between the given boundaries) and the initial mutation step, set as one third of the problem's search space.

Thirty different runs were performed for each test function, obtaining the results presented in table 1. If Wilcoxon test results are obtained from them, R-ESLAT obtains significantly better results in 20 out of the 27 test functions. These benefits come from the use of the local search techniques, which allow it to get values closer to the global optima, along with the enhanced exploration capabilities introduced with the mutagenesis operator (visible in the results of Easom's function, f8, where CMAES is not able to locate the zone of the search space where the fitness values actually provide exploitable information). On the other hand, the derandomized search performed by CMAES allows it to run in a much more reduced budget of function evaluations. These conclusions lead to considering the comparison on similar function evaluation budgets along with the improvement of this computational cost in R-ESLAT as the future lines.

## Acknowledgements

The authors would like to thank Alfonso Gómez-Jordana for his help in the development of the original algorithm. This work was supported in part by Projects CICYT TIN 2008-06742-C02-02/TSI, CICYT TEC 2008-06732-C02-02/ TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

<sup>1</sup>[http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/TestGO\\_files/Page364.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm)

**Table 1: Techniques comparison results**

id	Fitness		F. evals	
	R-ESLAT	CMAES	R-ESLAT	CMAES
f1	4.44E-15	8.34E+00	2.88E+05	7.40E+03
f2	7.36E-15	5.08E-02	1.29E+04	6.16E+02
f3	0.00E+00	6.08E-02	8.31E+03	6.59E+02
f4	2.89E-16	6.75E-16	9.64E+03	5.71E+02
f5	3.58E-07	4.62E-01	8.47E+03	6.11E+02
f6	8.85E-14	4.43E-16	3.75E+04	2.32E+03
f7	6.67E-01	6.67E-01	2.52E+05	1.07E+04
f8	4.00E-01	1.00E+00	7.87E+03	8.20E+00
f9	6.81E-16	8.10E+00	1.02E+04	7.44E+02
f10	0.00E+00	1.80E-03	3.01E+05	9.43E+03
f11	1.99E-06	3.97E-02	2.22E+04	1.79E+03
f12	4.65E-08	4.65E-08	8.68E+03	5.64E+02
f13	4.83E-20	7.86E-01	2.01E+05	8.29E+03
f14	3.85E-21	1.53E-16	9.83E+03	5.23E+02
f15	8.41E-02	2.27E+00	7.23E+04	5.82E+03
f16	8.62E+81	1.50E+85	1.76E+05	2.28E+05
f17	2.37E-12	1.05E-11	4.47E+05	4.40E+04
f18	1.13E-06	8.72E-12	2.93E+04	1.88E+04
f19	3.65E+00	6.54E+01	1.77E+05	1.31E+04
f20	3.98E-11	5.32E-01	4.81E+05	4.91E+04
f21	5.48E+02	5.34E+03	1.70E+05	2.71E+04
f22	5.92E-16	5.11E+00	2.06E+04	1.42E+03
f23	5.08E-12	5.57E+01	1.13E+04	1.05E+03
f24	5.22E-23	1.17E-15	1.42E+05	6.74E+03
f25	1.10E-20	2.03E-15	2.65E+05	9.12E+03
f26	0.00E+00	6.06E-14	4.97E+04	3.33E+03
f27	1.20E-14	3.80E-15	6.23E+05	1.75E+04

## 3. 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] C. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76, 1970.
- [3] N. Hansen, S. 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.
- [4] A. Hedar and M. Fukushima. Evolution Strategies Learned with Automatic Termination Criteria. In *Proceedings of SCIS-ISIS*, pages 1126–1134, 2006.
- [5] C. 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.
- [6] 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.
- [7] J. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308, 1965.
- [8] R. Ursem. Diversity-guided evolutionary algorithms. *Parallel Problem Solving from Nature*, pages 462–471, 2002.