

QR Mutations Improve Many Evolution Strategies - A Lot On Highly Multimodal Problems

Fabien Teytaud
Univ. Lille Nord de France,
Calais, France
teytaud@lisic.univ-littoral.fr

Olivier Teytaud
TAO (Inria), LRI, UMR 8623
CNRS - Univ. Paris-Sud
France
olivier.teytaud@inria.fr

ABSTRACT

Previous studies have shown the efficiency of using quasi-random mutations on the well-know CMA evolution strategy [13]. Quasi-random mutations have many advantages, in particular their application is stable, efficient and easy to use. In this article, we extend this principle by applying quasi-random mutations on several well known continuous evolutionary algorithms (SA, CMSA, CMA) and do it on several old and new test functions, and with several criteria. The results point out a clear improvement compared to the baseline, in all cases, and in particular for moderate computational budget.

1. INTRODUCTION: DERANDOMIZED EVOLUTIONARY ALGORITHMS

Several works have been devoted to the derandomization of evolutionary algorithms, typically using quasi-random (QR) methods. QR numbers are designed in order to be more uniformly distributed than classic pseudo-random numbers. We here focus on the continuous case. Some works apply QR sequences for the initialization of the individuals [7] or for the mutations [1, 14]. [13] pointed out that QR mutations are moderately better than pseudo-random ones in several settings. Inspired by [8], [13] concluded that in order to have good QR mutations it is important to have good QR sequences. This is particularly true for optimization of strongly multimodal functions [4, 13]. Another application of QR sequences is for restarts algorithms [12]; in this article, authors show that QR restarts outperform classic restart algorithms for multi-modal optimization. In this article, we point out the generality of the use of QR mutations by extending results from [14] to more algorithms, including CMSA and SA. We also discuss various criteria and include new highly multimodal objective functions. Recently, [2] experiments several algorithms on the CEC2015 testbed and notes that all algorithms which perform the best are the ones with QR mutations.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO'16 Companion July 20-24, 2016, Denver, CO, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: <http://dx.doi.org/10.1145/2908961.2909060>

2. ALGORITHMS

Generating quasi-random sequences in the square $[0, 1]^N$ is classical, many toolboxes can be found on the web. Converting from the uniform distribution in $[0, 1]^N$ to the standard Gaussian distribution can be done by composition, coordinate-wise, with the inverse cumulative distribution function of the standard Gaussian; in Octave/Matlab notation, *quasiRandomGaussian* = *norminv(quasiRandomInThe01Interval)*. More details around this can be found in [14].

Our modification is generic, in the sense that it can be applied to many evolution strategies. QRCMA refers to CMA with quasi-random mutations. QRSA refers to SA with quasi-random mutations. QRCMSA refers to CMSA with quasi-random mutations. SA is defined in [9, 10]. CMA and CMSA are respectively defined in [6] and [3]. We also use the $(1 + 1)$ -ES with one-fifth rule, multiplying the step-size by 1.5 in case of success and dividing by $1.5^{\frac{1}{4}}$ otherwise.

3. OBJECTIVE FUNCTIONS

In this study we experiment 4 unimodal functions (sphere, cigar, Ellipsoid and Schwefel) and 8 multimodal functions (Griewank, Rastrigin, Schaffer, Hump, Nsine, Randcuts and Randsines). The Schwefelmult function has been rescaled ($f_{rescaled}(x) = f(100x)$) so that its range matches the usual range of other functions. These functions are basically the same as in the CMAES source code, except that the SchwefelMult function is rescaled by a factor 100 so that its relevant range is similar to the one of other functions. The hump function [11] has been slightly modified (the optima are distributed as a Gaussian random variable with standard deviation 5 and one of the optimum has been slightly improved so that there is a unique optimum. The Nsine function [5] has been slightly modified so that the optimum is unique. The randcut function is defined as $f(x) = \sum_{i=1}^{30} 3b_i(a_i \cdot (x_i - x_i^*) - \lfloor a_i \cdot (x_i - x_i^*) \rfloor)^2 + \frac{N\|x\|^2}{10000}$ and the randsines as $f(x) = \sum_{i=1}^{30} 3b_i \sin(a_i \cdot (x_i - x_i^*))^2 + \frac{N\|x - x^*\|^2}{10000}$

4. EXPERIMENTS

In all our experiments, we measure (i) the simple regret, i.e. the fitness at the proposed individual minus the fitness at the optimum (ii) for various thresholds ϵ , the frequency at which the simple regret was greater than ϵ . All experiments are reproduced 99 times, moreover, in order to improve the

statistical significance, for all functions which have a random optimum, the 99 randomly drawn optima are the same for all algorithms (statistical pairing). The starting point is 0, and the optimum is randomly drawn, for each coordinate, as $U([-2, 2]) + \sigma\mathbb{N}$, where $U(I)$ is the uniform random variable on interval I , \mathbb{N} is the standard Gaussian random variable, and $\sigma = 3$ is the initial step-size used in all algorithms.

Because of space limitations, results are in <http://www.lri.fr/~teytaud/qrr.pdf>.

Additional results and downloads

The experiments were reproduced with random full rotations (built with Gram-Schmidt transformation) without changing the results (figures available at <http://www.lri.fr/~teytaud/withrot.pdf>, except for some functions, in the case of the results with highest precision and lowest dimension - for the Cigar function in dimension 5 and 10, CMA and CMA(r) perform better without restarts for the frequency of solving with precision 1 (all other results are preserved).

5. CONCLUSION

Previous results have shown that CMA is improved by QR mutations on BBOB. We here extend these results to

- more algorithms (CMSA, SA, CMA, and CMA with restarts), showing that QR works in most cases;
- additional functions, showing a bigger difference on highly multimodal functions.

CMA(R) (CMA with restarts), CMA, SA and CMSA are tested with random mutations and with quasi-random mutations. QR typically brings a significant improvement - there are cases in which the use of quasi-random numbers is the main difference between tested algorithms, i.e. all ES with QR outperform all ES without QR. Some multimodal functions from the global optimization literature (such that the hump and the NSine functions), challenging and not that usual in testbeds, and new multimodal functions have been included in tests, as well as randomly constructed multimodal functions - with these highly multimodal functions (hump, Schaffer, SchwefelMult, RandomCuts, RandomSines) QR sometimes switches from 80% failure to a negligible proportion of failure.

Furtherwork. Restarts should also benefit from quasi-random numbers; this is left as further work. SchwefelMult is an interesting stable counterexample - CMA outperforms QRCMA in a stable manner - this is not explained.

6. REFERENCES

- [1] Auger, A., Jebalia, M., Teytaud, O.: Xse: quasi-random mutations for evolution strategies. In: *Proceedings of Evolutionary Algorithms*, 12 pages (2005)
- [2] Berthier, V.: Experiments on the cec 2015 expensive optimization testbed. In: *Evolutionary Computation (CEC), 2015 IEEE Congress on*. pp. 1059–1066. IEEE (2015)
- [3] Beyer, H.G., Sendhoff, B.: Covariance matrix adaptation revisited - the CMSA evolution strategy. In: Rudolph, G., Jansen, T., Lucas, S.M., Poloni, C., Beume, N. (eds.) *Proceedings of PPSN*. pp. 123–132 (2008)
- [4] Chaslot, G., Hoock, J., Teytaud, F., Teytaud, O.: On the huge benefit of quasi-random mutations for multimodal optimization with application to grid-based tuning of neurocontrollers (2009), <http://hal.inria.fr/inria-00380125/en/>
- [5] Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: Schaffer, J.D. (ed.) *Proceedings of the 3rd International Conference on Genetic Algorithms*. pp. 42–50. Morgan Kaufman (1989)
- [6] Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 11(1) (2003)
- [7] Kimura, S., Matsumura, K.: Genetic algorithms using low-discrepancy sequences. In: *GECCO*. pp. 1341–1346 (2005)
- [8] L'Ecuyer, P., Lemieux, C.: *Recent Advances in Randomized Quasi-Monte Carlo Methods*, pp. 419 – 474. Kluwer Academic (2002)
- [9] Rechenberg, I.: *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart (1973)
- [10] Schwefel, H.P.: Adaptive Mechanismen in der biologischen Evolution und ihr Einfluss auf die Evolutionsgeschwindigkeit. Interner Bericht der Arbeitsgruppe Bionik und Evolutionstechnik am Institut für Mess- und Regelungstechnik Re 215/3, Technische Universität Berlin (Juli 1974)
- [11] Singh, G., Kalyanmoy Deb, D.: Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In: *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. pp. 1305–1312. ACM, New York, NY, USA (2006)
- [12] Teytaud, F., Schoenauer, M., Teytaud, O.: A rigorous runtime analysis for quasi-random restarts and decreasing stepsize. In: et al., J.K.H. (ed.) *Artificial Evolution*. Angers, France (Oct 2011)
- [13] Teytaud, O.: When does quasi-random work?. In: Rudolph, G., Jansen, T., Lucas, S.M., Poloni, C., Beume, N. (eds.) *PPSN. Lecture Notes in Computer Science*, vol. 5199, pp. 325–336. Springer (2008), <http://dblp.uni-trier.de/db/conf/ppsn/ppsn2008.html#Teytaud08>
- [14] Teytaud, O., Gelly, S.: Dcma: yet another derandomization in covariance-matrix-adaptation. In: *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. pp. 955–963. ACM, New York, NY, USA (2007)