Bias and variance in continuous EDA

Fabien Teytaud, Olivier Teytaud

Bias and variance in continuous EDA

F. Teytaud and O. Teytaud

TAO (Inria), LRI, UMR 8623(CNRS - Univ. Paris-Sud), bat 490 Univ. Paris-Sud
91405 Orsay, France, fteytaud@lri.fr

Abstract. Estimation of Distribution Algorithms are based on statistical estimates. We show that when combining classical tools from statistics, namely bias/variance decomposition, reweighting and quasi-randomization, we can strongly improve the convergence rate. All modifications are easy, compliant with most algorithms, and experimentally very efficient in particular in the parallel case (large offsprings).

1 Introduction

EMNA (Estimation of Multivariate Normal Algorithm), a widely known EDA (Estimation of Distribution Algorithms) is very efficient for parallel optimization. It was shown in [10] that, for large population size, λ, it outperforms SA-ES (Self-Adaptive Evolution Strategies, [8, 9]), which itself outperforms the cumulative step-length adaptation of CMA-ES (Covariance Matrix Adaptation Evolution Strategy [4]). These results are based on the comparison of the progress rate (slope of the convergence in log scale).

However, a detailed look at curves in [10] shows that the speed-up, as a function of λ, seemingly stagnates in EMNA for λ very large. This is not consistent with theoretical bounds in [13]: the speed-up, for a well designed algorithm, should increase to infinity, linearly for small values of λ and then logarithmically, but never stop at a constant. By the way, the speed-up of SA-ES is seemingly still increasing when the speed-up of EMNA stagnates; this suggests that for λ larger than in previous works, the robust SA-ES might indeed be faster than EMNA.

We will here first analyze reasons why the speed-up of EMNA stagnates for λ large. We will see that this is due to a bias/variance dilemma. The bias/variance dilemna is as follows. When x is an estimate of x^*, the bias/variance decomposition states that:

$$\mathbb{E}(x-x^*)^2 = \mathbb{E}\left((x-\mathbb{E}x)+(\mathbb{E}x-x^*)\right)^2 = \underbrace{\mathbb{E}(x-\mathbb{E}x)^2}_{variance} + \mathbb{E}\underbrace{(\mathbb{E}x-x^*)^2}_{squared\ bias}$$

Increasing λ reduces the variance, but does not reduce the bias. We will then see that, when using the reweighting as in the EMNA variant in [11], the bias disappears (the right hand side term is zero). Interestingly, when the bias is removed, then the following traditional statistical tricks against variance become much more efficient (as bias is removed, reducing the variance decreases the error

to zero!): use of quasi-random numbers, as in [12]; and use of λ large, *i.e.* parallel case. We point out that λ large is not only theoretical. For tuning our favorite application `http://www.lri.fr/~teytaud/mogo.html`, we use λ very large on a grid (thousands or tenths of thousands[1]).

In this paper, we define a version of EMNA including (i) quasi-random mutations as in [12] (ii) reweighting as in [11] (iii) a modification of the step-size adaptation rule based on [13].

2 Discussion: EMNA when λ is large

In this section, we discuss precisely the three weaknesses of EMNA (or, more generally, of evolutionary algorithms based on random sampling and averages) which are tackled in this paper: the weighting trouble, the limit behavior for λ large, and the possible redundancies of the random mutations.

2.1 The weighting

The "weighting trouble" is not new; it has been pointed out in [11]. It is the fact that the Gaussian distribution induces a bias towards the center of the offspring, regardless of the selection. [11] proposes to reweight offsprings (proportionally to the inverse of the Gaussian density) in order to correct this bias; this method is proved consistent in [11] and illustrated in Fig. 1. We recall the weighting trouble here because it is important for our bias/variance decomposition. We will use this "reweighting" modification in the sequel, as it is necessary for removing the bias: thereafter, we can work efficiently on the variance.

Some important papers around reweighting are [1, 2]; however, their goal is different: they choose weights for improving the convergence rates of ES for fixed values of λ, and in order to take into account the position of unselected points. Maybe their reweighting can be combined to the reweighting by inverse density; this is beyond the scope of this paper, in which we will keep only the reweighting by the inverse density as discussed above and formalized in Algorithm 2.

2.2 The limit behavior for λ large

A trouble in EMNA, whenever we use reweighting, is that the convergence rate converges to a fixed constant when λ goes to infinity. This is because, as a function of λ, the step-size does not decrease to 0. More formally, with e.g. $\mu = \lambda/4$, and for a given parent x_n and step-size σ_n,

$$\lim_{\lambda \to \infty} \sigma_{n+1} = std(\text{first quartile of the offspring generated at iteration } n) \quad (1)$$

where *std* is the standard deviation of a set of points. This is definitely not reasonable: when λ runs to infinity, the optimum is estimated more and more precisely

[1] Due to constraints on the scheduler, we often use λ larger than the number of cores - $\lambda = 32000$ on a few hundred cores is classical.

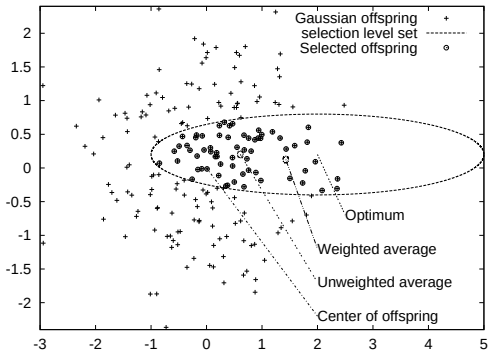

Fig. 1. Illustration of the requirement of weighting. A Gaussian offspring (200 samples) is generated. The best individuals are selected. Their unweighted average is biased towards the center of the distribution. The weighted average, following [11], is much better.

(at least now, as we have removed the bias by reweighting), and therefore σ should get smaller and smaller so that the search is more focused. Therefore, we might want to ensure that

$$\lim_{\lambda \to \infty} \sigma_{n+1} = 0. \tag{2}$$

With Eq. 2 instead of Eq. 1, we focus the search on the most important part. A question is then: to which extent should we force σ_{n+1}/σ_n to decrease to 0 ? We empirically choose the following formula:

$$\sigma \leftarrow \sigma / \max(1, (\log(\lambda)/2)^{1/N}) \tag{3}$$

where N is the dimension. This formula is probably too conservative and should be tuned carefully, but this will be sufficient as a proof of concept as illustrated in later results. This will be done just after the classical EMNA estimate of the step-size; see Alg. 2.

We here investigate the case of λ large. This is the case in which the problem consists in focusing the search more strongly than with standard values of λ, whilst avoiding premature convergence. In [3], the interested reader can find the reverse point of view: how to evaluate the Gaussian distribution (in particular when it includes a full covariance matrix and not only the diagonal case) properly in spite of the finiteness of λ.

2.3 Possibly redundant mutations

Random points can lead to a lot of redundancies between mutations. Therefore, many improved ways of generating more uniform points have been developed in the literature [7, 6, 15]. It was shown in [12] that in many cases using quasi-random sequences instead of random sequences provides big improvements, and

that there's almost no case in which it is harmful. We do not present quasi-random points instead of random points here; this was already published, in the evolutionary context, in [14].

Please note that quasi-random sequences are not intended to change the distribution. We consider Gaussian distributions only. But as well as one can use random Gaussian numbers, one can use quasi-random Gaussian numbers - the distribution is the same, we just take care of avoiding unlucky (redundant) cases when we use quasi-random sequences.

3 Improving EMNA

The standard EMNA algorithm is presented in Alg. 1 (in the isotropic case, but the general case is similar).

Algorithm 1 The standard EMNA algorithm.

Initialize $\sigma \in \mathbb{R}$, $y \in \mathbb{R}^N$.
while Halting criterion not fulfilled **do**
  for $l = 1..\lambda$ **do**
    $z_l = \sigma N_l(0, Id)$
    $y_l = y + z_l$
    $f_l = f(y_l)$
  end for
  Sort the individuals by increasing fitness; $f_{(1)} < f_{(2)} < \cdots < f_{(\lambda)}$.
  $z^{avg} = \frac{1}{\mu} \sum_{i=1}^{\mu} z_{(i)}$
  $\sigma = \sqrt{\frac{\sum_{i=1}^{\mu} ||z_{(i)} - z^{avg}||^2}{\mu \times N}}$
  $y = y + z^{avg}$
end while

We want to add three improvements:

- The first one is the use of reweighting [11]. This is a very simple and efficient modification against premature convergence. It is implemented as follows: when computing the new parent and/or the new covariance matrix and/or the new stepsize, give a weight to each selected individual: this weight is inversely proportional to the density of the Gaussian distribution used for sampling individuals (see algorithm below).
- The second one is the use of quasi-random numbers. We refer to [7, 6, 12] for more information around quasi-random numbers - if you are not very interested in this, you can just remember that there exists quasi-random numbers as well as random numbers, and they have some nice uniformity properties, and they provide improvements in continuous evolutionary algorithms.

– The third one is new. It is as simple as the two previous ones: after having estimated the step-size as in the classical EMNA, just divide it by $\max(1, (\log(\lambda)/2)^{1/N})$. The justification of this modification is discussed in 2.2.

Our version including these 3 improvements is presented in Alg. 2.

Algorithm 2 Our improved version of EMNA (IEMNA), with quasi-random mutations, reweighting, faster decrease of step-size.

Initialize $\sigma \in \mathbb{R}$, $y \in \mathbb{R}^N$.
while Halting criterion not fulfilled **do**
  for $l = 1..\lambda$ **do**
    $p_l = \sigma \times QRN_l(0, Id)$ // quasi-random Gaussian vector [12]
    $w_l = 1/d(p_l)$ where d is the Gaussian density [11]
    $z_l = \sigma p_l$
    $y_l = y + z_l$
    $f_l = f(y_l)$
  end for
  Sort the individuals by increasing fitness; $f_{(1)} < f_{(2)} < \cdots < f_{(\lambda)}$.
  $s = \sum_{i=1}^{\mu} w_{(i)}$
  Renormalize for all $i \leq \mu$, $w_{(i)} \leftarrow w_{(i)}/s$
  $z^{avg} = \sum_{i=1}^{\mu} w_{(i)} z_{(i)}$
  $\sigma = \sqrt{\frac{\sum_{i=1}^{\mu} w_{(i)} ||z_{(i)} - z^{avg}||^2}{N}}$
  $\sigma = \sigma / \max((\log(\lambda)/2)^{1/N}, 1)$ // a modification proposed in this paper
  $y = y + z^{avg}$
end while

4 Experimental results

All experiments below are based on EMNA (Alg. 1) and IEMNA (Alg. 2), and intermediate versions with only part of the improvements proposed in IEMNA. EMNA is the baseline from [5]. EMNA+QR is EMNA, plus the quasi-random mutations as in [14, 12]. EMNA+QR+reweighting is EMNA+QR, plus the reweighting as in [11] (see the reweighting formula in section 2.1 or Alg. 2). IEMNA is EMNA+QR+reweighting+LB; this is the complete improved version in Alg. 2. We perform our experiments with anisotropic Gaussians, with diagonal covariance matrix (i.e. one step-size per axis).

4.1 Case with good initialization

We experiment IEMNA (Alg. 2) with initial step-size $\sigma = 1$ on each axis, and $x_0 = (1, 1, \ldots, 1)$. The number of generations is 50. The negative convergence rate, estimated by $N \log(||x_{50}||/||x_0||)/50$ (with N the dimension), is presented in Table 1 for the sphere function $x \mapsto ||x||$, Table 2 for $x \mapsto \sum_i \log(|x_i|)+cos(1/x_i)$, Table 3 for the cigar function $x \mapsto \sum_i (10^4)^i x_i^2$. "QR" denotes the use of quasi-random mutations, "weight" the use of reweighting, "LB" the use of step-size decreasing as in section 2.2 ($\sigma \leftarrow \sigma / \max((\log(\lambda)/2)^{1/N}, 1)$). The results clearly show (i) the success of QR (always better than the baseline), (ii) the success of LB for λ large in the reweighted case, (iii) the poor performance of reweighting, which moderately but constantly decreases the performance. Result (iii) can be explained by the fact that we are in a case in which premature convergence is unlikely - we will reproduce the experiments with a poor initialization of $\sigma = 0.01$ in the following section in order to clarify this idea.

Table 1. Experiments on the sphere function. Numbers are the negative convergence rates (the lower the better). Each modification is validated or invalidated separately (p-values are for EMNA+QR versus the initial EMNA (Alg. 1); then EMNA+QR+weighting against EMNA+QR; and finally the complete new version EMNA+QR+LB+reweighting as in Alg. 2) against EMNA+QR+weighting. We see that IEMNA provide by far the best results.

Dimension, lambda	Baseline	+QR	+ weight	+LB (IEMNA)	P-value for QR	P-value for weight	P-value for LB
2,20	-0.345	-1.252	-1.743	**-2.103**	0	3.894e-07	0
3,30	-0.697	-2.299	-2.277	**-2.398**	0	0.636	0.03
4,40	-0.749	-2.541	-2.397	**-2.578**	0	0.998	0.03
5,50	-1.330	-2.885	-2.677	**-2.730**	0	1	0.31
2,60	-1.967	-2.086	-2.022	**-2.713**	0.001	1	0
3,90	-2.330	-2.392	-2.282	**-3.047**	1.588e-13	1	0
4,120	-2.543	-2.627	-2.443	**-3.271**	0	1	0
5,150	-2.790	-2.858	-2.622	**-3.488**	0	1	0
2,200	-2.050	-2.089	-2.112	**-3.004**	0	3.308e-14	0
3,300	-2.340	-2.404	-2.293	**-3.302**	0	1	0
4,400	-2.601	-2.658	-2.480	**-3.547**	0	1	0
5,500	-2.828	-2.908	-2.673	**-3.750**	0	1	0
2,600	-2.080	-2.101	-2.061	**-3.190**	4.447e-08	1	0
3,900	-2.369	-2.443	-2.320	**-3.519**	0	1	0
4,1200	-2.642	-2.717	-2.519	**-3.764**	0	1	0
5,1500	-2.886	-2.964	-2.718	**-3.975**	0	1	0
2,2000	-2.103	-2.188	-2.111	**-3.434**	0	1	0
3,3000	-2.404	-2.476	-2.367	**-3.726**	0	1	0

We also show graphically the negative convergence rate (defined as in previous tables) in Fig. 2 in the case of the sphere function.

4.2 Case with poor initialization

We now reproduce the experiments, comparing EMNA (Alg. 1) and our improved version IEMNA (Alg. 2), but in a different setting. Now, σ is initialized at a small value 0.01. Results are presented in Tables 4, 5, 6. QR is still always efficient (or has no effect). We clearly see that, now, with risk of premature convergence, reweighting is very efficient; on the other hand, LB is less efficient.

Table 2. Experiments on the multimodal function (see text). The lower, the better (as previous figures and following the definition of negative convergence rate in the text). QR is the only stable and efficient modification here.

Dimension, lambda	Baseline	+QR	+ weight	+LB (IEMNA)	P-value for QR	P-value for weight	P-value for LB
2,20	-0.709	**-1.301**	-1.105	-0.529	0	1	1
3,30	-0.971	**-1.332**	-0.799	-0.537	1.721e-09	1	1.00
4,40	-1.204	**-1.388**	-0.713	-0.858	6.262e-06	1	0.01
5,50	-1.359	**-1.520**	-0.702	-0.445	0.000	1	1.00
2,60	-1.139	**-1.181**	-0.655	-1.157	2.315e-07	1	0
3,90	**-1.231**	-1.229	-0.481	-0.619	0.559	1	0.00
4,120	**-1.357**	-1.353	-0.344	-0.480	0.624	1	0.00
5,150	-1.477	**-1.503**	-0.351	-0.391	0.010	1	0.05
2,200	**-1.104**	-1.074	-0.402	-0.822	1	1	0
3,300	-1.210	**-1.243**	-0.178	-0.233	1.415e-05	1	0.00
4,400	-1.352	**-1.368**	-0.205	-0.183	0.013	1	0.98
5,500	-1.495	**-1.518**	-0.145	-0.161	0.001	1	0.34
2,600	-1.100	**-1.133**	-0.119	-0.534	0	1	0
3,900	-1.240	**-1.252**	-0.181	-0.179	0.031	1	0.61
4,1200	-1.389	**-1.427**	0.224	0.093	7.638e-10	1	0.01
5,1500	-1.539	**-1.579**	0.726	0.715	2.300e-08	1	0.37
2,2000	-1.124	**-1.146**	-0.124	-0.210	3.187e-09	1	0
3,3000	-1.269	**-1.307**	0.081	-0.031	6.342e-12	1	0.01
2,6000	-1.144	**-1.162**	-0.173	-0.181	7.199e-11	1	0

We reproduce graphically in Fig. 4 the same results for the sphere function; we see that only EMNA+QR+weight is nearly as efficient as IEMNA, and that these two algorithms (the only difference between them is that LB is used in IEMNA) are the only algorithms with non negligible convergence rates (all other algorithms have convergence rates nearly zero).

5 Conclusion

Thanks to a proper bias/variance decomposition, we have emphasized the reasons for some troubles in EMNA: premature convergence as emphasized in [11], and the poor speed-up for λ very large. Thanks to this understanding, we could apply classical statistical techniques: reweighting and quasi-random. Also, thanks to these good estimates, the "LB" modification, reducing the step-size, leads to improved convergence rates. Results show that using sound bias/variance principles, we can improve Estimation of Distribution Algorithms. All suggested modifications are simple and quite general. The improvements for $\lambda = 20, N = 2$ can reach 700 % for EMNA+QR over EMNA, 500 % for IEMNA over EMNA, and replace premature convergence by real convergence for small step-sizes (thanks to reweighting). For λ very large, the speed-up seemingly goes to infinity; we guess that a more carefully tuned formula for the step-size adaptation should provide much better results.

Quasi-random (QR) is for sure easier in continuous domains (not necessarily for monomodal EDA - quasi-random numbers can be used for many distributions as explained in e.g. [14]), but they can also be experimented in discrete cases (this is, however, not straightforward). We confirm here results from previous publications, in the case of EMNA and different fitness functions. Interestingly,

Table 3. Experiments on the cigar function (see text). The number are negative convergence rates as defined in the text; the lower, the better. IEMNA is clearly the best algorithm here.

Dimension, lambda	Baseline	+QR	+ weight	+LB (IEMNA)	P-value for QR	P-value for weight	P-value for LB
2,20	-0.222	-1.833	**-1.885**	-0.758	0	0.024	1
3,30	-0.209	-1.397	**-1.643**	-1.096	0	0.055	1.00
4,40	-0.192	-1.071	-1.353	**-1.469**	2.635e-13	0.027	0.23
5,50	-0.103	**-0.787**	-0.426	-0.566	1.607e-12	0.996	0.16
2,60	-1.774	-1.954	-1.950	**-2.734**	0.002	0.796	0
3,90	-1.586	-2.127	-2.018	**-2.686**	1.056e-07	1	0
4,120	-1.391	-2.043	-1.842	**-2.498**	9.712e-11	1	0
5,150	-1.034	-1.922	-1.579	**-2.212**	0	1	0
2,200	-1.919	-2.003	-1.967	**-2.834**	0.016	1	0
3,300	-2.067	-2.143	-1.998	**-2.905**	7.749e-14	1	0
4,400	-1.982	-2.095	-1.853	**-2.719**	0.001	1	0
5,500	-1.798	-1.941	-1.527	**-2.356**	1.642e-05	1	0
2,600	-2.014	-2.060	-1.994	**-3.075**	1.064e-12	1	0
3,900	-2.106	-2.176	-2.025	**-3.084**	3.609e-08	1	0
4,1200	-2.072	-2.154	-1.855	**-2.894**	5.221e-10	1	0
5,1500	-1.936	-2.040	-1.603	**-2.520**	2.829e-06	1	0
2,2000	-2.015	-2.059	-2.023	**-3.371**	0	1	0
3,3000	-2.157	-2.232	-2.070	**-3.288**	9.525e-12	1	0
2,6000	-2.047	-2.122	-2.019	**-3.476**	0	1	0

quasi-random seemingly comes as a free lunch and sometimes very strongly improves the result.

Reweighting can be used in all algorithms based on empirical distributions. The sample of selected points can be replaced by the corresponding sample of weighted selected points, for most (if not all) EDA, both in discrete and continuous domains. This makes sense also for evolution strategies. Importantly, reweighting is not a free lunch - as shown in Table 2, there are cases in which reweighting is harmful. On the other hand, it's the only efficient tool against premature convergence.

The fact that the step-size should decrease faster than the standard deviation of selected points when λ increases is also quite general. A simple and different way of developing this idea is to reduce μ - perhaps this would have the same effect. Decreasing the step-size according to lower bounds (LB) is not a free lunch: it is possibly harmful in particular for moderate values of λ and when there is a risk of premature convergence.

As a summary of this paper: (1) Quasi-random mutations improve the results. The improvement can be moderate or huge, depending on the framework - almost always at least a few percents (with also decreased variance of performance), and up to 800 % improvement for $\lambda = 10N$ for the cigar function in dimension N or 300 % on the sphere function for $\lambda = 10N$. (2) Reweighting performs very well against premature convergence; on the other hand, it decreases the performance for cases in which the initialization avoids premature convergence. (3) LB (forced decrease of the step-size for λ large) performs incredibly well when there's no risk of premature convergence. On the other hand, it can of course (as it decreases σ!) be harmful when there can be premature convergence. The main limitation of this paper is that we only tested our ideas on EMNA, for a small set of fitness functions and small dimension, and especially for λ not too small. It has been

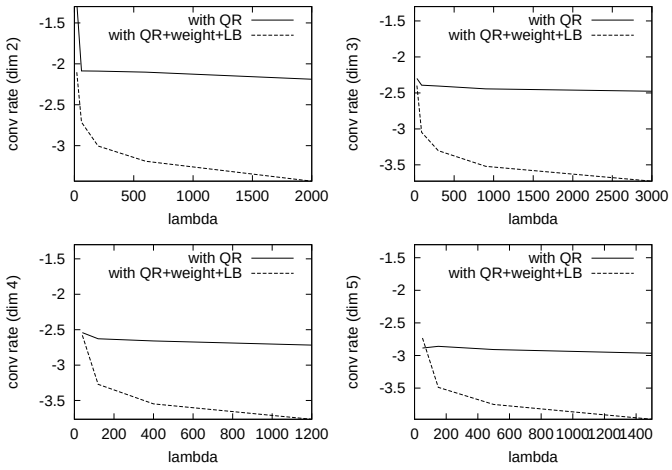

Fig. 2. Synthesis of the performance of LB in the case of the sphere. Convergence rate of EMNA with QR mutations (as in [14, 12]), which provide the state of the art convergence rates for EMNA with λ large before this paper, and convergence rates of IEMNA=EMNA+QR + reweighting + LB as in this paper (Alg. 2). The improvement is very clear for λ large. This is not in the case with poor initialization; with poor initialization (see text), results are much more impressive for reweighting (which avoids premature convergence, as shown in [11]), but in that case LB would be harmful (LB makes it more difficult to recover from the bad initialization).

shown in previous papers that QR mutations are efficient in many other cases [14, 12]. For reweighting, further work is for sure useful for the case of λ small. The "LB" modification is dedicated to λ large and is of course not relevant for λ small - if λ is small, our modification, in Eq. 3, will not change anything due to the max.

References

1. D. V. Arnold. Weighted multirecombination evolution strategies. *Theor. Comput. Sci.*, 361(1):18–37, 2006.
2. D. V. Arnold and D. C. S. V. Wart. Cumulative step length adaptation for evolution strategies using negative recombination weights. In M. Giacobini, A. Brabazon, S. Cagnoni, G. D. Caro, R. Drechsler, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, A. Fink, J. McCormack, M. O'Neill, J. Romero, F. Rothlauf, G. Squillero, S. Uyar, and S. Yang, editors, *EvoWorkshops*, volume 4974 of *Lecture Notes in Computer Science*, pages 545–554. Springer, 2008.
3. W. Dong and X. Yao. Unified eigen analysis on multivariate gaussian based estimation of distribution algorithms. *Inf. Sci.*, 178(15):3000–3023, 2008.
4. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
5. P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

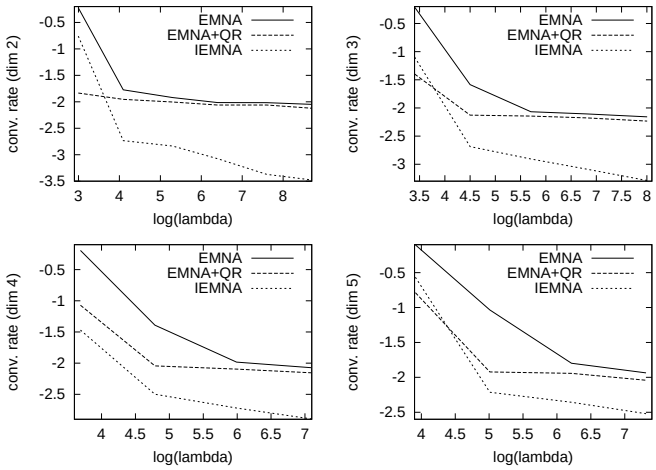

Fig. 3. Results on the cigar function. Negative convergence rates for EMNA, EMNA+QR, IEMNA (the lower, the better). For small values of λ, EMNA+QR already provides a great improvement over QR; but for λ large, only IEMNA makes a huge difference. This is due to the fact that when bias is removed (by reweighting) and step-sizes are reduced accordingly (by LB), then variance is the main term, and variance is strongly reduced by QR mutations.

6. P. L'Ecuyer and C. Lemieux. *Recent Advances in Randomized Quasi-Monte Carlo Methods*, pages 419 – 474. Kluwer Academic, 2002.
7. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. 1992.
8. I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
9. H.-P. Schwefel. 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.
10. F. Teytaud and O. Teytaud. Eda are fast when lambda is large. In *Proceedings of EvoStar 2009*, page accepted, 2009.
11. F. Teytaud and O. Teytaud. Why one must use reweighting in estimation of distribution algorithms. In *Proceedings of Gecco*, 2009.
12. O. Teytaud. When does quasi-random work?. In G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors, *PPSN*, volume 5199 of *Lecture Notes in Computer Science*, pages 325–336. Springer, 2008.
13. O. Teytaud and H. Fournier. Lower bounds for evolution strategies using vc-dimension. In G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors, *PPSN*, volume 5199 of *Lecture Notes in Computer Science*, pages 102–111. Springer, 2008.
14. O. Teytaud and S. Gelly. DCMA: yet another derandomization in covariance-matrix-adaptation. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 955–963, New York, NY, USA, 2007. ACM.

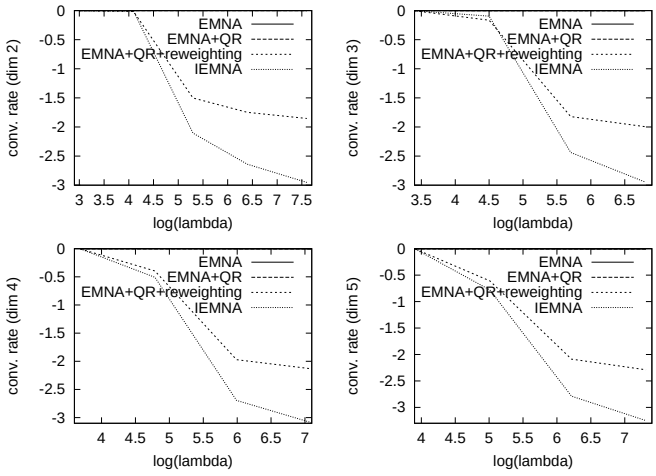

Fig. 4. Results of EMNA, EMNA+QR, EMNA+QR+reweighting, and IEMNA=EMNA+QR+LB+reweighting, on the sphere function with very small initial step size. Numbers are negative convergence rates. The lower, the better the result; EMNA, EMNA+QR have convergence rate nearly 0. The main important tool here is reweighting, but IEMNA still improves the results over EMNA+QR+reweighting.

15. B. Vandewoestyne and R. Cools. Good permutations for deterministic scrambled halton sequences in terms of l2-discrepancy. *Computational and Applied Mathematics*, 189(1,2):341:361, 2006.

Table 4. Experiments on the sphere function with small initial step-size. The lower the number, the better the result: EMNA+QR+LB+reweighting is usually the best algorithm for λ large enough; for λ small, EMNA+QR+reweighting is stronger - LB is only efficient for λ large, in particular with too small initial step-size. Each modification is validated (p-values are for EMNA+QR versus the initial EMNA (Alg. 1); then EMNA+QR+LB against EMNA+QR; and finally the complete new version EMNA+QR+LB+reweighting as in Alg. 2) in this case with small initial step-size (*i.e.* we test the robustness against premature convergence.).

Dimension, lambda	Baseline	+QR	+ weight	+LB (IEMNA)	P-value for QR	P-value for weight	P-value for LB
2,20	-0.000	-0.001	**-0.001**	-0.001	2.204e-05	0	1
3,30	-0.001	-0.001	**-0.003**	-0.002	1.056e-06	0	1
4,40	-0.002	-0.003	**-0.005**	-0.004	4.436e-11	0	1.00
5,50	-0.003	-0.004	-0.007	**-0.008**	1.042e-08	0	0.04
2,60	-0.001	-0.001	**-0.014**	-0.005	0.000	0	1
3,90	-0.002	-0.003	**-0.165**	-0.096	6.729e-12	7.252e-11	1.00
4,120	-0.004	-0.004	-0.395	**-0.508**	4.912e-11	5.747e-11	0.13
5,150	-0.005	-0.006	-0.609	**-0.779**	0	0	0.08
2,200	-0.001	-0.001	-1.501	**-2.106**	5.588e-12	0	0
3,300	-0.003	-0.003	-1.821	**-2.437**	0.001	0	0
4,400	-0.004	-0.005	-1.970	**-2.693**	1.701e-11	0	0
5,500	-0.006	-0.007	-2.087	**-2.786**	0	0	0
2,600	-0.001	-0.001	-1.748	**-2.640**	0.009	0	0
3,900	-0.003	-0.003	-1.995	**-2.945**	0.006	0	0
4,1200	-0.005	-0.005	-2.131	**-3.086**	5.772e-07	0	0
5,1500	-0.007	-0.007	-2.288	**-3.250**	0.046	0	0
2,2000	-0.001	-0.001	-1.853	**-2.952**	0.002	0	0

Table 5. Experiments on the multimodal function (see text) with small initial step-size (*i.e.* we test the robustness against premature convergence). Numbers are the negative convergence rates; the lower the better. LB is not efficient in this case in which we have a risk of poor local minima.

Dimension, lambda	Baseline	+QR	+ weight	+LB (IEMNA)	P-value for QR	P-value for weight	P-value for LB
2,20	-0.001	-0.001	**-0.001**	-0.001	0.032	0	1
3,30	-0.001	-0.002	**-0.003**	-0.002	0.002	0	1
4,40	-0.002	-0.003	**-0.006**	-0.005	0.016	0	1.00
5,50	-0.004	-0.004	**-0.008**	-0.007	0.000	0	0.81
2,60	-0.001	-0.001	**-0.018**	-0.006	7.251e-11	0	1
3,90	-0.002	-0.003	-0.109	**-0.144**	0	0	0.036
4,120	-0.004	-0.004	**-0.240**	-0.227	1.265e-14	0	0.70
5,150	-0.005	-0.006	**-0.281**	-0.231	0	0	0.984
2,200	-0.001	-0.001	-0.250	**-0.693**	0.014	0	0
3,300	-0.003	-0.003	-0.191	**-0.210**	2.797e-09	0	0.02
4,400	-0.004	-0.005	**-0.219**	-0.202	0	0	0.962
5,500	-0.007	-0.007	**-0.211**	-0.192	8.039e-12	0	0.89
2,600	-0.001	-0.001	-0.240	**-0.312**	3.719e-05	0	0.01
3,900	-0.003	-0.003	**-0.177**	-0.168	3.195e-08	0	0.89
4,1200	-0.005	-0.005	-0.200	**-0.202**	0.034	0	0.44
5,1500	-0.007	-0.007	**-0.024**	-0.002	0.000	0.273	0.71
2,2000	-0.001	-0.001	**-0.154**	-0.053	0.038	0	1

Table 6. Experiments on the cigar function (see text) with small initial step-size (*i.e.* we test the robustness against premature convergence). Numbers are the negative convergence rates; the lower the better. QR clearly works as usual; other modifications are unstable of harmful (this should be different for λ larger).

Dimension, lambda	Baseline	+QR	+ weight	+LB (IEMNA)	P-value for QR	P-value for weight	P-value for LB
2,20	-0.000	-0.000	**-0.000**	-0.000	0.062	8.180e-12	1
3,30	-0.000	**-0.000**	-0.000	-0.000	0.030	0.961	1.00
4,40	-0.000	**-0.000**	-0.000	6.412e-05	0.108	0.999	0.89
5,50	-0.000	**-0.001**	0.000	0.000	0.075	1	0.06
2,60	-0.000	-0.001	**-0.015**	-0.012	3.657e-11	0	1
3,90	-0.001	-0.001	**-0.033**	-0.033	0.003	0	0.47
4,120	-0.001	**-0.001**	0.011	0.019	0.000	0.927	0.73
5,150	-0.001	**-0.001**	0.167	0.173	0.125	1	0.61
2,200	-0.001	-0.001	**-0.016**	-0.016	0.112	0	0.97
3,300	-0.001	-0.001	**-0.035**	-0.036	0.000	0	0.00
4,400	-0.001	**-0.001**	0.070	0.111	0.082	1	1.00
5,500	-0.001	**-0.001**	0.518	0.512	0.000	1	0.41
2,600	-0.001	-0.001	**-0.017**	-0.017	0.009	0	1.00
3,900	-0.001	-0.001	-0.009	**-0.026**	0.385	0.028	0.00
4,1200	-0.001	**-0.001**	0.230	0.165	9.548e-05	1	0.00
5,1500	-0.001	**-0.001**	0.799	0.782	0.752	1	0.22
2,2000	-0.001	-0.001	-0.006	**-0.020**	0.205	0	0