

Enhance Continuous Estimation of Distribution Algorithm by Variance Enlargement and Reflecting Sampling

Zhigang Ren[†], *Member IEEE*, Chenlong He, Dexing Zhong, Shanshan Huang, and Yongsheng Liang
Department of Automation Science and Technology, School of Electronic and Information Engineering
Xi'an Jiaotong University, Xi'an, China

[†]Email: renzg@mail.xjtu.edu.cn

Abstract—Estimation of distribution algorithm (EDA) is a kind of typical model-based evolutionary algorithm (EA). Although possessing competitive advantages in theoretical analysis, current EDAs may encounter premature convergence due to the rapid shrinkage of the search range and the relatively low sampling efficiency. Focusing on continuous EDAs with Gaussian models, this paper proposes a novel probability density estimator which can adaptively enlarge the variances and thus endow EDA with flexible search behavior. For the estimated probability density, a reflecting sampling strategy which can further improve the search efficiency is put forward. With these two algorithmic strategies, a new EDA variant named EDA_{ve-rs} is developed. Experimental results on a set of benchmark problems demonstrate that EDA_{ve-rs} outperforms conventional EDAs and can produce superior solutions in comparison with some state-of-the-art EAs.

Keywords—estimation of distribution algorithm; premature convergence; probability density; reflecting sampling

I. INTRODUCTION

Estimation of distribution algorithms (EDAs) [1]-[3] refer to a class of evolutionary algorithms (EAs) which produce individuals at each generation according to a certain probability distribution. The probability distribution is generally learnt from the relatively high quality individuals produced at the last generation, thus the structures of the promising solution regions can be captured and consequently employed to guide the search process. During the past few decades, EDAs attracted much research effort, providing a large amount of EDA variants which achieved great success in both combinatorial and continuous domains. In this paper, EDAs for continuous domain are studied.

For continuous EDAs, Gaussian probability models are most commonly used. According to the way in representing the dependences among variables, Gaussian models can be categorized into three types. The simplest one is the univariate model which neglects all the dependences. A representative algorithm with this type of model is the univariate marginal distribution algorithm (UMDAc) [2], [3]. A slightly more sophisticated model is the one that considers some important variable dependencies. To identify these dependencies, Bayesian factorization is usually employed [2], [4]. The full

model takes all the variable dependencies into account. A representative algorithm of this type is estimation of multivariate normal density algorithm (EMNAg) [3].

Despite showing much convenience in theoretical analysis, EDAs with Gaussian models still have some shortcomings. The most significant one is that the estimated variances decrease fast even in slope-like solution regions [4], [5]. This makes the search range shrink rapidly, finally causing premature convergence. To remedy this issue, much research effort has been done. Yuan and Gallagher [6] showed in an initial investigation that, by keeping variances at a value of at least 1, the solution quality of some difficult problems can be improved. Pošík [7] suggested enlarging variances by a constant factor at each generation, while Ocenasek *et al.* [4] developed an adaptive variance scaling (AVS) strategy based on the so-called 1/5-success-rule. Grahl *et al.* [8] proposed another AVS strategy which increases the scaling factor when solution improves, otherwise reduces it. It is notable that the AVS strategy in [8] does not directly scale variances, unless it identifies that the algorithm is traversing a slope. The identification process can be accomplished by some other techniques such as the correlation triggering rule [8] and the standard deviation ratio (SDR) [9]. Based on AVS and SDR, Bosman *et al.* [10] introduced another algorithmic technique named anticipated mean shift (AMS) for EDA. The combination of AVS, SDR, and AMS leads to a powerful EDA variant known as AMaLGaM [10], [11].

Besides scaling variances beyond their estimation values, various other methods have been developed to enhance the performance of EDA. Dong and Yao [12], [13] discussed the effectiveness of modifying the estimated covariance matrix with its eigenvalues. Karshenas *et al.* [14] showed the advantages of the regularization technique in estimating Gaussian model for EDA. The performance of EDAs with multiple sub-models was studied in [15]. The effectiveness of the local search technique in improving EDA was also researched [16]. However, all of these improvements are accompanied by more complicated algorithmic models, more free parameters, and greater computation burden.

Aimed at developing a simple and practical EDA against difficult continuous problems, this paper proposes a novel probability density estimator for EDAs with Gaussian models. The density estimator first improves the AMS technique

This work was supported by the National Natural Science Foundation of China under grant 61105126 and by the Postdoctoral Science Foundation of China under grant 2014M560784.

presented in [10], then provides a variance estimator based on the new mean obtained by AMS. As long as AMS gets a better mean, the variance estimator will adaptively enlarge variances without using an explicit factor. In this way, premature convergence can be avoided to a large extent. Moreover, a reflecting sampling strategy is put forward. It makes use of the previous inferior samples by generating their mirrors *w.s.t* the mean, rather than directly discards them as conventional methods. This further enhances the search efficiency. With the two algorithmic strategies mentioned above, we develop a new EDA variant named EDA_{ve-rs}.

The reminder of this paper is organized as follows. Section II reviews the continuous EDAs with Gaussian models. Section III describes the proposed EDA_{ve-rs} in detail. Experimental results and analysis are presented in Section IV. Section V concludes the paper.

II. CONTINUOUS EDAS WITH GAUSSIAN MODELS

As a model-based EA, EDA assumes that the good solutions in the solution space obey a certain probability distribution. During the search process, it tries to learn this distribution and generate solutions according to the learning results [1]-[3]. The basic steps of EDA is given in Table I. It starts from an initial population (step 1) which is usually generated in a random way. At each generation, after selecting a subset of solutions according to a selection rule such as truncation (step 3), a probability distribution model is learnt from these selected solutions to encode their characteristics (step 4). The model is then used to generate new solutions for the next generation (step 5). This procedure is repeated until meeting the stop criteria.

TABLE I. BASIC STEPS OF EDA

1.	Generate an initial population \mathbf{P}^0 , set $t = 0$;
2.	Evaluate each solution in \mathbf{P} using the objective function;
3.	Select promising solutions \mathbf{S}' from \mathbf{P} according to a selection rule;
4.	Build a probability model F^t based on \mathbf{S}' ;
5.	Generate a new population \mathbf{P}^{t+1} by sampling from F^t ;
6.	Set $t \leftarrow t + 1$;
7.	Goto step 2 until a stopping criterion is met;
8.	Output the best solution in \mathbf{P} .

Continuous EDAs generally use Gaussian model as the basis of their probability distribution models, since Gaussian model facilitates theoretical analysis and is computationally tractable. The Gaussian probability density function for a n -dimensional random vector \mathbf{X} can be parameterized by a mean vector $\boldsymbol{\mu}$ and a covariance matrix \mathbf{C} as follows:

$$F_{(\boldsymbol{\mu}, \mathbf{C})}(\mathbf{x}) = \frac{(2\pi)^{-n/2}}{(\det \mathbf{C})^{1/2}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu})}. \quad (1)$$

Since the real mean vector and covariance matrix which totally have $0.5n^2 + 1.5n$ free parameters are unknown, we have to estimate them from some samplings. It is well known that, for a selected solution set \mathbf{S} in EDA, the maximum likelihood (ML) estimations of $\boldsymbol{\mu}$ and \mathbf{C} are

$$\bar{\boldsymbol{\mu}} = \frac{1}{|\mathbf{S}|} \sum_{i=1}^{|\mathbf{S}|} \mathbf{S}_i, \quad (2)$$

$$\bar{\mathbf{C}} = \frac{1}{|\mathbf{S}|} \sum_{i=1}^{|\mathbf{S}|} (\mathbf{S}_i - \bar{\boldsymbol{\mu}})(\mathbf{S}_i - \bar{\boldsymbol{\mu}})^T. \quad (3)$$

The Gaussian model estimated by (2) and (3) takes the dependencies between all pairs of variables into account and is used by EMNAg [3]. Theoretically, this model can capture some complex structural characteristics of the solution space. However, it is difficult to be accurately estimated in the context of EDA due to its too many parameters. To overcome this defect, UMDAc adopts a much simpler Gaussian model which neglects all the variable dependencies. It means that the covariance matrix is diagonal and we just need to estimate $2n$ parameters. Unfortunately, the performance of both UMDAc and EMNAg is undesirable. They fail on some benchmark problems where other continuous EAs succeed [4], [5]. The main reason consists in that the estimated mean for Gaussian model can only move a limited distance before converging due to the rapid shrinkage of the estimated variance.

A straightforward way to improve UMDAc and EMNAg is to enlarge the variance adaptively. Along this direction, Bosman and his coworkers did outstanding work [8]-[11]. They suggest enlarging the variance with AVS when EDA obtains better solutions in the slope-like solution regions. By this means, the rapid shrinkage of variances can be alleviated. Bosman *et al.* also introduced the AMS technique into EDA [10], with the aim of making the probability density ellipsoid parallel to the improvement direction of the objective function. The principle of AMS is generating two sets of solutions around two different means, respectively, and estimating variances according to the solutions selected from both sets. The ideas of AVS and AMS are appealing, but give rise to some new issues:

1) It is difficult to decide when to apply AVS, since it is difficult to identify whether EDA is traversing a slope or not. Although the proposed triggering rules, i.e., the one based on correlation [8] and the SDR trigger [9], adhere to our intuition, they may provide us faulty judgments. For example, a small SDR value does not necessarily ensure EDA is not traversing a slope.

2) To re-align the probability density ellipsoid, AMS shifts the mean at each generation, regardless of whether the current mean is better than the one of the last generation, not to mention the quality of the new mean after the shift. Besides, even the new mean improves, AMS still asks to sample some solutions taking the old mean as the center, so that the density ellipsoid can be re-aligned. Therefore, there is plenty of room for performance improvement.

3) AVS, SDR, and AMS introduce many new parameters, including variance increase factor, variance decrease factor, percentile of generated solutions to apply AMS, and factor of mean shift for moving generated solutions, etc. [11]. Part of these parameters can be derived under some ideal assumptions, while the other part of them are difficult to set.

III. EDA WITH VARIANCE ENLARGEMENT AND REFLECTING SAMPLING: EDA_{VE-RS}

EDA_{VE-RS} improves the existing AMS technique, develops a brand new variance estimator and a novel reflecting sampling strategy. It will be shown that, even with the simple univariate Gaussian model, EDA_{VE-RS} can achieve excellent performance.

A. Improved AMS

Recall that the mean of Gaussian probability distribution decides the search center of EDA and the difference between the means in two subsequent generations indicates the search direction. Then the real goal of estimating the mean is not to rigidly obtain the mathematical mean of the selected solutions, but to get a new promising search center. Keeping this principle in mind, we estimate the mean as follows. First, calculate an initial mean using the selected solutions:

$$\bar{\mu}^t = \frac{\sum_{i=1}^{|S^t|} [\log(|S^t|+1) - \log i] S_{i:|S^t|}^t}{\sum_{j=1}^{|S^t|} [\log(|S^t|+1) - \log j]}, \quad (4)$$

where $S_{i:|S^t|}^t$ denotes the i th best solution in the selected set S^t .

It can be seen from (4) that the initial mean $\bar{\mu}^t$ is a weighted average of the selected solutions, and the weights are logarithmically proportional to the ranks of the selected solutions. Once $\bar{\mu}^t$ is obtained, we attempt to shift it according to its difference from $\hat{\mu}^{t-1}$ which denotes the real mean used for sampling at the last generation:

$$\Delta \bar{\mu}^t = \bar{\mu}^t - \hat{\mu}^{t-1}, \quad (5)$$

$$\hat{\mu}^t = \begin{cases} \bar{\mu}^t + 2\Delta \bar{\mu}^t, & \text{if } f(\bar{\mu}^t + 2\Delta \bar{\mu}^t) < f(\bar{\mu}^t) < f(\hat{\mu}^{t-1}) \\ \bar{\mu}^t - 0.5\Delta \bar{\mu}^t, & \text{if } f(\bar{\mu}^t) > \max\{f(\hat{\mu}^{t-1}), f(\bar{\mu}^t - 0.5\Delta \bar{\mu}^t)\} \\ \bar{\mu}^t, & \text{otherwise} \end{cases}, \quad (6)$$

where $f(\cdot)$ is the objective function to be minimized. When $f(\bar{\mu}^t)$ is less than $f(\hat{\mu}^{t-1})$, Eq. (6) updates $\bar{\mu}^t$ with $\bar{\mu}^t + 2\Delta \bar{\mu}^t$, providing that $f(\bar{\mu}^t + 2\Delta \bar{\mu}^t)$ is less than $f(\bar{\mu}^t)$. The purpose of this operation is twofold: One is to exploit the inertia of the direction of $\Delta \bar{\mu}^t$, and the other is to adaptively enlarge variances, which will be explained in the next subsection. On the contrary, if $\bar{\mu}^t$ is worse than $\hat{\mu}^{t-1}$, then (6) updates $\bar{\mu}^t$ with $\bar{\mu}^t - 0.5\Delta \bar{\mu}^t$ under the condition that the latter is better than the former.

Remark: Some other continuous EAs, such as CMA-ES [17], also use the weighted average of the selected solutions to estimate the mean for Gaussian model, but they do not shift the estimated mean, therefore no inertia of the promising direction is exploited. As discussed at the end of section II, although the existing AMS technique performs the mean shift operation, it considers little about the quality of the mean, as its purpose is just to re-align the probability density ellipsoid.

B. Variance Estimator

The covariance matrix of Gaussian probability distribution affects the search range and direction of EDA. It can be

geometrically described by an ellipsoid in the hyperspace [10]. It is hoped that the major axis of this ellipsoid parallels to the descent direction of the objective function and does not shrink too rapidly. However, the opposite situation shown in Fig. 1 often happens, if we primarily calculate the covariance matrix using the ML estimator. This phenomenon was first noted in [18] and was also discussed in [10]. The underlying reason for this phenomenon is that Gaussian model generates more solutions near its mean, while the good solutions which are commonly with shorter distances from the optimum but farther away from the mean tend to be selected. Then the selected solution points form a relatively small cone in the hyperspace. Consequently, the new estimated density ellipsoid is likely to contract with its major axis almost perpendicular to the descent direction of the objective function.

Recall that the covariance matrix of a random vector equals its mixed central moment of the second order. Then a straightforward way to re-align its probability density ellipsoid is to change the center, i.e., its mean. Benefiting from the improved AMS technique, this can be achieved very conveniently in EDA_{VE-RS} by replacing the mean $\bar{\mu}^t$ with $\hat{\mu}^t$. Since EDA_{VE-RS} adopts the univariate Gaussian model, here we just give the variance estimator:

$$\hat{\Delta}^t = \frac{1}{|S^t|} \sum_{i=1}^{|S^t|} (S_i^t - \hat{\mu}^t) \circ (S_i^t - \hat{\mu}^t), \quad (7)$$

where the symbol \circ denotes dot product, and $\hat{\Delta}^t$ can be interpreted as the estimation of the second order moment *w.r.t.* $\hat{\mu}^t$. It is easy to imagine that $\hat{\mu}^t$ will elongate the density ellipsoid along the direction of $(\hat{\mu}^t - \bar{\mu}^t)$ as shown in Fig. 2. In addition, the estimated variances would be greater than the ones obtained with ML estimator, provided that $\hat{\mu}^t$ does not equal $\bar{\mu}^t$. This property can prevent the rapid contraction of the density ellipsoid.

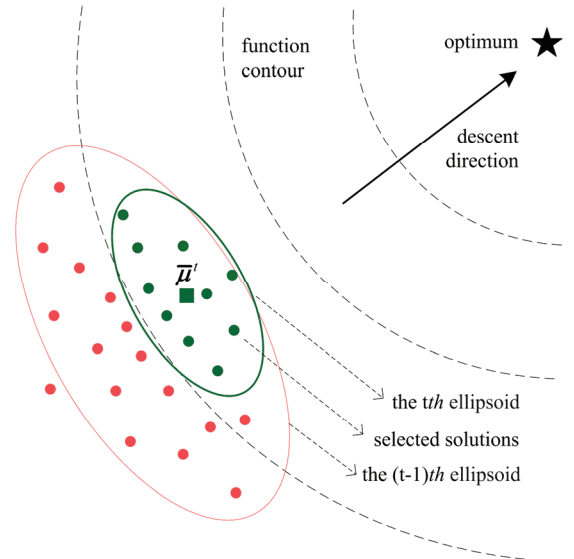


Fig. 1. Schematic for the change of the density ellipsoid estimated by the conventional ML estimator.

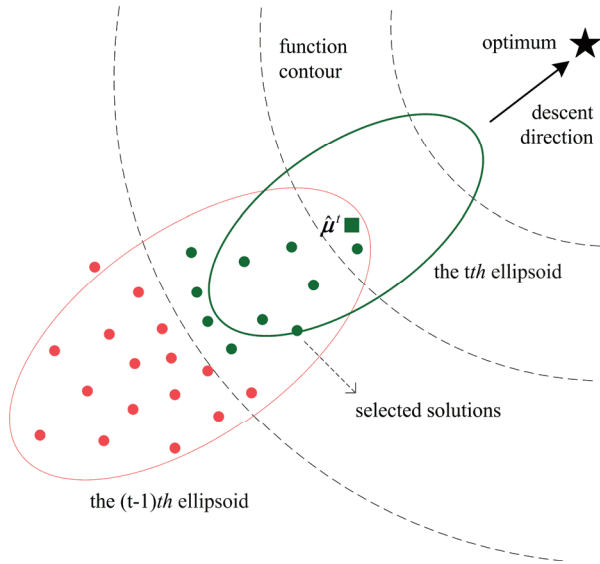


Fig. 2. Schematic for the change of the density ellipsoid estimated by the new variance estimator.

C. Reflecting Sampling

From Figs. 1-2, it can be seen that, no matter the density ellipsoid is re-aligned or not, it is partitioned into two parts by the objective function contour across the mean point. This implies that part of sampled solutions are better than the mean, and the other part are not. According to the conventional sampling strategy, those worse solutions will be directly discarded or be chosen into the selection set, which will deteriorate the search efficiency.

The reflecting sampling strategy proposed in this paper tries to exploit those worse solutions for the purpose of increasing the sampling efficiency. The idea behind it is very intuitive and elegant. If a solution x is worse than the mean $\hat{\mu}$, it implies that x is outside the objective function contour across $\hat{\mu}$, then the mirror of x w.r.t. $\hat{\mu}$, i.e., $x' = 2\hat{\mu} - x$, is much more likely to be located inside the contour. This phenomenon can be visually illustrated by Figs. 1-2. As a result, it is beneficial to sample the reflected mirror x' next time. This new sample is not necessarily superior to the mean, but it can outperform an independently sampled solution in the sense of probability. Therefore, the reflecting sampling strategy can provide more good solutions than the conventional one for a given probability density and a certain sampling number. Its pseudo-code is listed in Table II, where the role of the boolean variable $flag$ is to avoid repeatedly sample a same solution.

D. Framework of EDA_{ve-rs}

EDA_{ve-rs} follows the basic framework of UMDac, but employs a new probability density estimator and a new sampling strategy. For the selection step, the truncation selection with a truncation ratio of $\tau < 1$ is used. The best solution at each generation is maintained for the next one. Besides, since EDA_{ve-rs} explicitly evaluates the estimated mean which can also be regarded as a sample, then we just need to produce $p-2$ new solutions for a population of size p at each

generation. Table III provides the pseudo-code of EDA_{ve-rs} . It can be seen that, compared with UMDac, EDA_{ve-rs} does not introduce new free parameters.

TABLE II. PSEUDO-CODE OF REFLECTING SAMPLING

Input: function $f(\cdot)$, probability density $F_{(\hat{\mu}, \hat{\Delta})}(\cdot)$, sampling number m ;
Output: sampled solution set A .
1. Initialize $A = \emptyset$, $i = 0$, $flag = false$;
2. Set $i \leftarrow i + 1$;
3. If $flag = true$ then
4. Sample $A_i = 2\hat{\mu} - A_{i-1}$, make A_i feasible, set $flag = false$;
5. Else
6. Independently sample A_i according to $F_{(\hat{\mu}, \hat{\Delta})}(\cdot)$, make A_i feasible;
7. If $f(A_i) > f(\hat{\mu})$ then set $flag = true$;
8. Set $A \leftarrow A \cup \{A_i\}$;
9. Goto step 2 until $i = m$.

TABLE III. PSEUDO-CODE OF EDA_{ve-rs}

Input: objective function $f(\cdot)$, stopping criterion;
Output: the best solution in the final population.
1. Initialize population size p , truncation ratio τ ;
2. Generate p solutions randomly to initialize P^0 , set $t = 0$;
3. Set $S' \leftarrow$ the best $\lfloor \tau p \rfloor$ solutions in P' (truncation selection);
4. Estimate $\hat{\mu}'$, $\hat{\Delta}'$ according to S' and (4)-(7);
5. Set $A' \leftarrow p-2$ solutions obtained with reflecting sampling strategy;
6. Set $P^{t+1} \leftarrow A' \cup \{S'_{1:S'}\}$, $t \leftarrow t + 1$;
7. Goto step 3 until a stopping criterion is met.

IV. EXPERIMENTAL STUDY

The purpose of this section is twofold: One is to investigate the effectiveness of the new proposed algorithmic strategies, and the other is to evaluate the performance of EDA_{ve-rs} by comparing it with other EDA variants and some state-of-the-art EAs. In the experiment, EDA_{ve-rs} was implemented in Matlab[®] and run on a PC with an Intel i7-4790 CPU under Windows 7 system. The first 12 benchmark functions developed for IEEE CEC2005 [19], including 5 unimodal ones (i.e., $f_1 - f_5$) and 7 multimodal ones (i.e., $f_6 - f_{12}$), were used to test EDA_{ve-rs} . Each function was set with the dimension of 30 and was independently tested 25 times by each algorithm involved. Unless mentioned otherwise, we terminated each algorithm after $3.0E+5$ function evaluations (FEs) and evaluated the algorithm according to the function error value (FEV) of the best solution obtained, i.e., the difference between its objective value and that of the global optima.

A. Comparison with Other EDA Variants

In order to show the effectiveness of the strategies developed for EDA_{ve-rs} , we compared it with UMDac [3] and AMaLGaM [11], where the latter also uses an AMS technique and a variance enlargement strategy. The population sizes and the selection ratios of these three algorithms were all set as $p = 500$, $\tau = 0.35$, and the other parameter settings of

AMaLGaM were the same as in [11]. Besides, AMaLGaM was also required to use the univariate Gaussian model and a single population, and no restart strategy was allowed for achieving a fair comparison.

At each generation, we statistically calculated the FEV of the estimated mean and the average variance of all variables, with the aim of showing the effectiveness of the improved AMS and the new variance estimator, respectively. Figs. 3-4 present the corresponding evolution curves, where the benchmark function f_6 is used as an example. From Fig. 3, it can be observed that AMaLGaM and EDA_{ve-rs} significantly outperform UMDAc, and EDA_{ve-rs} improves AMaLGaM, since UMDAc stagnates much earlier than the other two algorithms and EDA_{ve-rs} keeps a more desirable downward tendency compared with AMaLGaM. This observation is consistent with the result shown in Fig. 4, where UMDAc persistently declines in terms of the average variance, while AMaLGaM and EDA_{ve-rs} keep relatively great variances after a brief decline.

To test the effectiveness of the reflecting sampling strategy, we specially constructed a same algorithm as EDA_{ve-rs} except that no reflecting sampling strategy was used, and denoted it as EDA_{ve} . Fig. 5 presents the evolution of FEVs of the best solutions derived from these two algorithms, where more FEs ($1.5E+6$) were allowed in order to show their performance difference more obviously. It can be seen that, equipped with the reflecting sampling strategy, EDA_{ve-rs} always obtains better solutions. Fig. 5 once again verifies that EDA_{ve-rs} has strong exploration ability and it can achieve much better solutions when given more computation amount.

Table IV reports the mean and the standard deviation of FEVs of the best solutions derived from UMDAc, AMaLGaM, and EDA_{ve-rs} over 25 independent runs. The t -test values between the former two algorithms and EDA_{ve-rs} are also given. It can be observed that EDA_{ve-rs} surpasses UMDAc on 11 out of 12 functions and improves AMaLGaM on nine functions. For functions f_8 , f_9 , and f_{10} , although the t -test values indicate EDA_{ve-rs} gets worse results than UMDAc or AMaLGaM, the mean FEVs of the three algorithms are still on the same order of magnitude. To sum up, EDA_{ve-rs} performs best among three EDA variants compared, which mainly profits from the new developed algorithmic strategies.

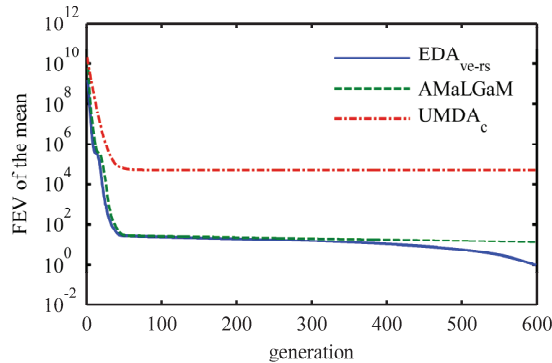


Fig. 3. Evolution of FEVs of the means derived from UMDAc, AMaLGaM, and EDA_{ve-rs} on f_6 .

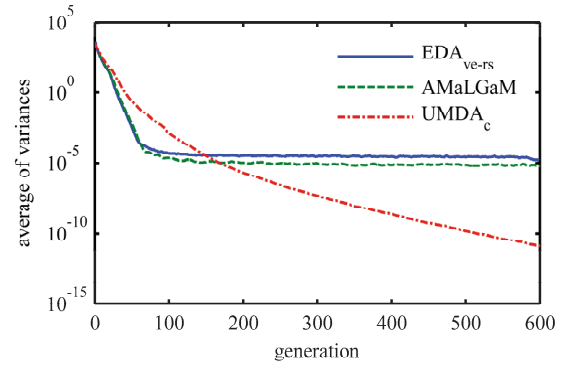


Fig. 4. Evolution of the average variance of all variables derived from UMDAc, AMaLGaM, and EDA_{ve-rs} on f_6 .

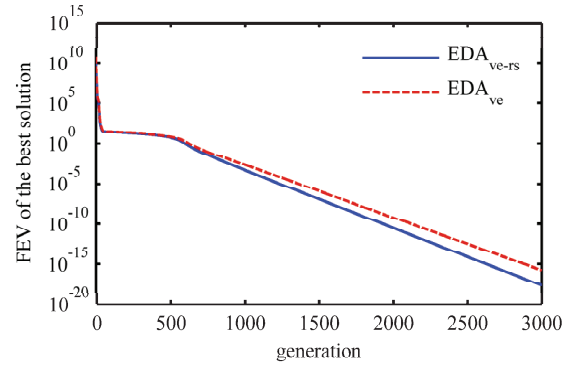


Fig. 5. Evolution of FEVs of the best solutions derived from EDA_{ve-rs} and EDA_{ve} on f_6 .

B. Comparison with Other State-of-the-Art EAs

We also compared EDA_{ve-rs} with other three EAs, including CMA-ES [17], CLPSO [20], and SaDE [21]. The reason for selecting these three algorithms for comparison lies in that they respectively represent the state-of-the-art of three EAs for continuous optimization, i.e., ES, PSO, and DE. By setting the parameters of CMA-ES, CLPSO, and SaDE as the same in their original papers, the corresponding experimental results shown in Table IV can be obtained. To make the comparison fair and alleviate the complexity of the experiment, the results of these three algorithms are directly taken from [22].

From the last four columns of Table IV, it is evident that EDA_{ve-rs} performs best among the four compared algorithms in a statistically significant fashion. Concretely, EDA_{ve-rs} outperforms CMA-ES on seven functions and is defeated by CMA-ES on the other five functions. This result is appealing, taking into account the fact that CMA-ES adopts a complex multivariate Gaussian model, while EDA_{ve-rs} just adopts a simple univariate one. The performance of EDA_{ve-rs} is also significantly better than that of CLPSO on nine functions. CLPSO uniquely achieves the best solution for function f_6 among all the algorithms tested, while the solutions it obtains for functions f_1 and f_8 are similar to the ones obtained by EDA_{ve-rs} from a practical point of view. Compared with SaDE, EDA_{ve-rs} shows significantly better and worse performance on eight functions and four functions, respectively.

It is worth to mention that EDA_{ve-rs} can generate solutions very close to the optima for some functions such as f_1, f_2 , and f_7 , but it can hardly get the real optima, since it samples according to a certain probability distribution. All the model-based EAs,

including CMA-ES, AMaLGaM, and UMDAc, satisfy this property. Even so, the above comparisons clearly demonstrate that EDA_{ve-rs} performs significantly better than the three competitors.

TABLE IV. THE MEAN AND THE STANDARD DEVIATION (MEAN \pm STD DEV (t -TEST)) OF FEVS OF THE BEST SOLUTIONS DERIVED FROM SIX ALGORITHMS

Fun.	UMDAc	AMaLGaM	CMA-ES	CLPSO	SaDE	EDA_{ve-rs}
f_1	8.37E-25 \pm 9.31E-26 (44.71)	8.64E-25 \pm 9.49E-26 (45.32)	1.58E-25 \pm 3.35E-26 (22.99)	0.00E+00 \pm 0.00E+00 ($-\infty$)	0.00E+00 \pm 0.00E+00 ($-\infty$)	3.96E-27 \pm 8.20E-28
f_2	1.08E+04 \pm 7.82E+02 (69.13)	1.01E+00 \pm 2.92E-01 (17.32)	1.12E-24 \pm 2.93E-25 (-1.41E+15)	8.40E+02 \pm 1.90E+02 (22.11)	8.26E-06 \pm 1.65E-05 (2.50)	8.27E-11 \pm 6.64E-11
f_3	1.94E+07 \pm 5.05E+06 (18.96)	4.68E+05 \pm 9.00E+04 (10.06)	5.54E-21 \pm 1.69E-21 (-8.48+26)	1.42E+07 \pm 4.19E+06 (16.60)	4.27E+05 \pm 2.08E+05 (3.37)	2.87E+05 \pm 5.13E+04
f_4	1.59E+04 \pm 8.76E+02 (78.63)	8.34E+03 \pm 1.34E+03 (23.41)	9.15E+05 \pm 2.16E+06 (2.11)	6.99E+03 \pm 1.73E+03 (14.19)	1.77E+02 \pm 2.67E+02 (-35.66)	2.08E+03 \pm 5.79E+02
f_5	3.71E+03 \pm 1.27E+02 (74.85)	2.41E+03 \pm 1.44E+02 (21.00)	2.77E-10 \pm 5.04E-11 (-1.79+14)	3.86E+03 \pm 4.35E+02 (23.59)	3.25E+03 \pm 5.90E+02 (12.22)	1.81E+03 \pm 1.72E+02
f_6	5.08E+04 \pm 1.08E+05 (2.35)	1.28E+01 \pm 6.02E-01 (98.37)	4.78E-01 \pm 1.32E+00 (-1.75)	4.16E+00 \pm 3.48E+00 (4.62)	5.31E+01 \pm 3.25E+01 (8.02)	9.42E-01 \pm 1.33E-01
f_7	1.45E+02 \pm 2.32E+01 (31.31)	2.27E-03 \pm 4.14E-03 (2.74)	1.82E-03 \pm 4.33E-03 (2.10)	4.51E-01 \pm 8.47E-02 (26.62)	1.57E-02 \pm 1.38E-02 (5.68)	2.80E-16 \pm 1.33E-16
f_8	2.09E+01 \pm 3.81E-02 (-2.76)	2.10E+01 \pm 5.58E-02 (5.38)	2.03E+01 \pm 5.72E-01 (-5.66)	2.09E+01 \pm 4.41E-02 (-5.35)	2.09E+01 \pm 4.95E-02 (-4.78)	2.09E+01 \pm 5.08E-02
f_9	5.77E+00 \pm 1.30E+00 (6.75)	2.43E+00 \pm 1.19E+00 (-6.71)	4.45E+02 \pm 7.12E+01 (30.97)	0.00E+00 \pm 0.00E+00 ($-\infty$)	2.39E-01 \pm 4.33E-01 (-43.66)	4.02E+00 \pm 1.78E+00
f_{10}	7.56E+00 \pm 2.21E+00 (3.60)	3.86E+00 \pm 1.26E+00 (-8.35)	4.63E+01 \pm 1.16E+01 (17.38)	1.04E+02 \pm 1.53E+01 (32.04)	4.72E+01 \pm 1.01E+01 (20.41)	5.97E+00 \pm 1.84E+00
f_{11}	8.67E+00 \pm 8.65E-01 (40.85)	8.67E+00 \pm 8.65E-01 (6.86)	7.11E+00 \pm 2.14E+00 (12.86)	2.60E+01 \pm 1.63E+00 (74.83)	1.65E+01 \pm 2.42E+00 (30.77)	1.61E+00 \pm 1.54E+00
f_{12}	3.97E+04 \pm 1.96E+04 (9.62)	3.27E+03 \pm 3.80E+03 (1.70)	1.26E+04 \pm 1.74E+04 (3.05)	1.79E+04 \pm 5.24E+03 (15.19)	3.02E+03 \pm 2.33E+03 (2.23)	1.98E+03 \pm 2.19E+03
N_b	1	2	5	3	4	—
N_s	0	1	0	0	0	—
N_w	11	9	7	9	8	—

* N_b, N_s, N_w denote the numbers of functions on which the performance of an algorithm is significantly better, similar to, and significantly worse than that of EDA_{ve-rs} , respectively. The performance of an algorithm on a function is significantly better (worse) than that of EDA_{ve-rs} at a level of 0.05 if the corresponding t -test value is less than -2.064 (greater than 2.064), otherwise we say that they have similar performance.

V. CONCLUSION

In this paper, a new EDA variant called EDA_{ve-rs} is presented for continuous optimization problems. EDA_{ve-rs} keeps using Gaussian model, but does not primly estimate the mean and the variance with the conventional ML estimator. Instead, it first shifts a weighted mean along the descent direction of the objective function, then takes the second order moment *w.s.t* the shifted mean as the variance. In this way, the probability density ellipsoid is elongated along the descent direction of the objective function. This facilitates EDA_{ve-rs} sampling solutions in the promising regions on the one hand, and enlarges the variances on the other hand, then premature convergence can be avoided to a great degree. Furthermore, a novel reflecting sampling strategy is proposed for EDA_{ve-rs} . Instead of directly discarding the previous inferior samples, this sampling strategy generates their mirrors *w.s.t* the mean, then better samples can be obtained with a greater probability. Profiting from these new developed algorithmic strategies, EDA_{ve-rs} demonstrates very appealing performance on a set of 12 benchmark functions, and outperforms two conventional EDA variants and other three state-of-the-art EAs. Since this study only discusses the effectiveness of the developed algorithmic strategies on the univariate Gaussian model, our future work will focus on their applications on the multivariate models.

REFERENCES

- [1] H. Mühlenbein, G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem*

Solving from Nature - PPSN IV, ser. Lecture Notes in Computer Science, vol. 1141, Springer, 1996, pp. 178–187.

- [2] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization by learning and simulation of bayesian and gaussian networks," University of the Basque Country, Spain, Technical Report, KZZA-IK-4-99, 1999.
- [3] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer, 2001.
- [4] J. Ocenasek, S. Kern, N. Hansen, and P. Koumoutsakos, "A mixed bayesian optimization algorithm with variance adaptation," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, vol. 3242, Springer, 2004, pp. 352–361.
- [5] J. Grah, S. Minner, and F. Rothlauf, "Behaviour of UMDAc with truncation selection on monotonous functions," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Edinburgh, UK, 2005, pp. 2553–2559.
- [6] B. Yuan, M. Gallagher, "On the importance of diversity maintenance in estimation of distribution algorithms," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Washington, USA, 2005, pp. 719–726.
- [7] P. Pošik, "Preventing premature convergenc in a simple EDA via global step size setting," in *Parallel Problem Solving from Nature - PPSN X*, ser. Lecture Notes in Computer Science, vol. 5199, Springer, 2008, pp. 549–558.
- [8] J. Grah, P. A. N. Bosman, and F. Rothlauf, "The correlation-triggered adaptive variance scaling IDEA," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Seattle, USA, 2006, pp. 397–404.
- [9] P. A. N. Bosman, J. Grah, and F. Rothlauf, "SDR: A better trigger for adaptive variance scaling in normal EDAs," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, London, UK, 2007, pp. 492–499.
- [10] P. A. N. Bosman, J. Grah, and D. Thierens, "Enhancing the performance of maximum-likelihood Gaussian EDAs using anticipated mean shift," in *Parallel Problem Solving from Nature - PPSN X*, ser. Lecture Notes in Computer Science, vol. 5199, Springer, 2008, pp. 133–143.

- [11] P. A. N. Bosman, J. Grahl, and D. Thierens, "Benchmarking parameter-free AMALGAM on functions with and without noise," *Evolutionary Computation*, vol. 21, no. 3, pp. 445–469, 2013.
- [12] W. Dong and X. Yao, "Covariance matrix repairing in Gaussian based EDAs," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, 2007, pp. 415–422.
- [13] W. Dong and X. Yao, "Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms," *Information Sciences*, vol. 178, no. 15, pp. 3000–3023, 2008.
- [14] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Regularized continuous estimation of distribution algorithms," *Applied Soft Computing*, vol. 13, no. 5, pp. 2412–2432, 2013.
- [15] P. Yang, K. Tang, and X. Lu, "Improving estimation of distribution algorithm on multi-modal problems by detecting promising areas," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1438–1449, 2015.
- [16] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 807–822, 2015.
- [17] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [18] Y. Cai, X. Sun, H. Xu, and P. Jia, "Cross entropy and adaptive variance scaling in continuous EDA," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, London, UK, 2007, pp. 609–616.
- [19] P. N. Suganthan *et al.*, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. #2006005, 2005.
- [20] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [21] A. K. Qin, V. L. Huang, P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [22] Y. Wang, H. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, no. 1, pp. 232–247, 2014.