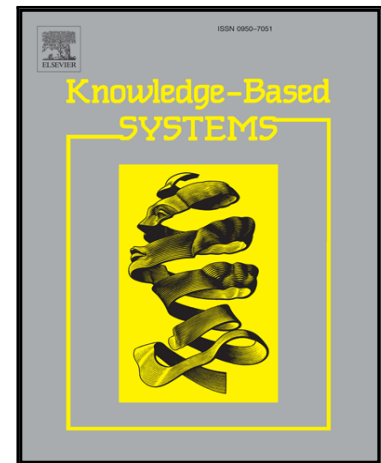


## Accepted Manuscript

Anisotropic Adaptive Variance Scaling for Gaussian Estimation of Distribution Algorithm

Zhigang Ren , Yongsheng Liang , Lin Wang , Aimin Zhang ,  
Bei Pang , Biying Li

PII: S0950-7051(18)30052-2  
DOI: [10.1016/j.knosys.2018.02.001](https://doi.org/10.1016/j.knosys.2018.02.001)  
Reference: KNOSYS 4209



To appear in: *Knowledge-Based Systems*

Received date: 12 September 2017  
Revised date: 25 January 2018  
Accepted date: 1 February 2018

Please cite this article as: Zhigang Ren , Yongsheng Liang , Lin Wang , Aimin Zhang , Bei Pang , Biying Li , Anisotropic Adaptive Variance Scaling for Gaussian Estimation of Distribution Algorithm, *Knowledge-Based Systems* (2018), doi: [10.1016/j.knosys.2018.02.001](https://doi.org/10.1016/j.knosys.2018.02.001)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

[Title Page]

## **Anisotropic Adaptive Variance Scaling for Gaussian Estimation of Distribution Algorithm**

Zhigang Ren<sup>a</sup>, Yongsheng Liang<sup>a</sup>, Lin Wang<sup>b</sup>, Aimin Zhang<sup>a</sup>, Bei Pang<sup>a</sup>, Biying Li<sup>a</sup>

<sup>a</sup> Department of Automation Science and Technology, School of Electronic and Information Engineering,  
Xi'an Jiaotong University, Xi'an, China

<sup>b</sup> School of Information Science and Technology, Northwest University, Xi'an, China

### **Correspondence information:**

**Corresponding author name:** Zhigang Ren

**Affiliation:** Department of Automation Science and Technology, School of Electronic and Information  
Engineering, Xi'an Jiaotong University

**Permanent address:** No.28 Xianning West Road, Xi'an Shaanxi, 710049, P.R. China

**Email address:** renzg@mail.xjtu.edu.cn

**Telephone number:** +86 136 3023 7679

## Anisotropic Adaptive Variance Scaling for Gaussian Estimation of Distribution Algorithm

Zhigang Ren<sup>a,\*</sup>, Yongsheng Liang<sup>a</sup>, Lin Wang<sup>b</sup>, Aimin Zhang<sup>a</sup>, Bei Pang<sup>a</sup>, Biying Li<sup>a</sup>

<sup>a</sup> Department of Automation Science and Technology, School of Electronic and Information Engineering,  
Xi'an Jiaotong University, Xi'an, China

<sup>b</sup> School of Information Science and Technology, Northwest University, Xi'an, China

### Abstract

Traditional Gaussian estimation of distribution algorithms (EDAs) are confronted with issues that the variable variances decrease fast and the main search direction tends to become perpendicular to the improvement direction of the fitness function, which reduces the search efficiency of Gaussian EDAs (GEDAs) and makes them subject to premature convergence. In this paper, a novel anisotropic adaptive variance scaling (AAVS) technique is proposed to improve the performance of traditional GEDAs and a new GEDA variant named AAVS-EDA is developed. The advantages of AAVS over the existing variance scaling strategies lie in its ability for tuning the variances and main search direction of GEDA simultaneously, which are achieved by anisotropically scaling the variances along different eigendirections based on corresponding landscape characteristics captured by a simple topology-based detection method. Besides, AAVS-EDA also adopts an auxiliary global monitor to ensure its convergence by shrinking all the variances if no improvement is achieved in a generation. The evaluation results on 30 benchmark functions of CEC2014 test suite demonstrate that AAVS-EDA possesses stronger global optimization efficiency than traditional GEDAs. The comparison with other state-of-the-art evolutionary algorithms also shows that AAVS-EDA is efficient and competitive.

**Keywords:** Gaussian estimation of distribution algorithm; premature convergence; search direction; anisotropic adaptive variance scaling

---

\* Corresponding author. E-mail address: renzg@mail.xjtu.edu.cn

## 1. Introduction

As a special class of evolutionary algorithm (EA) [1], estimation of distribution algorithm (EDA) [2-4] is characterized by the way of generating new solutions, i.e., sampling solutions according to a probability distribution, but not through crossover and mutation operators as other kinds of EAs. The probability distribution employed in each generation is generally estimated from the relatively high-quality solutions obtained in previous generations. It can capture the structure of the problem being solved to a certain extent and consequently guide the algorithm to more promising solution regions. During the past few decades, EDAs attracted much research effort and achieved great success in both combinatorial and continuous domains [5,6]. In this paper, EDAs for continuous domain are studied.

Continuous EDAs usually adopt Gaussian probability model [3,4] and histogram model [7] as the basic probability model. According to the way in representing variable dependencies, Gaussian models for EDAs can be further categorized into three kinds. The simplest one is the univariate model which neglects all the variable dependencies. A representative algorithm with this type of model is the univariate marginal distribution algorithm (UMDA<sub>c</sub>) [3,4]. A slightly more sophisticated model is the one that just considers some important variable dependencies. To identify these dependencies, Bayesian factorization is usually employed [3,8]. The multivariate model takes all the variable dependencies into account. A representative algorithm of this type is estimation of multivariate normal density algorithm (EMNA<sub>g</sub>) [4].

Although possessing clear physical concept, traditional Gaussian EDAs (GEDAs) often suffer from premature convergence [8,11-15]. Early improvement studies attributed this defect to the rapid shrink of variances and developed many variance scaling strategies. Ocenasek et al. [8] proposed a variance adaption operator for the mixed Bayesian optimization algorithm [9] based on the well-known 1/5-success-rule [10]. Yuan and Gallagher [11] claimed that the performance of GEDA could be improved on certain problems by compulsively keeping the variances at a value of at least 1. Pošík [12] suggested enlarging variances by a

constant factor. Grahl et al. [13] proposed an adaptive variance scaling (AVS) strategy which increases the variances when the best solution improves, otherwise reduces them. Nevertheless, AVS does not directly tune variances in each generation unless it identifies that the algorithm is traversing a slope. To achieve this, Grahl and his coworkers successively developed two identification strategies, i.e., the strategies based on correlation triggering [13] and standard deviation ratio (SDR) [14]. Cai et al. [15] developed a different type of variance scaling method named cross entropy adaptive variance scaling, which calculates the scaling factor by minimizing the cross entropy between the current probability model and the predicted model for the next generation.

Besides directly tuning variances, some researchers achieved variance scaling by modifying the eigenvalues of the estimated covariance matrix. Wagner et al. [16] proposed an eigenspace EDA (EEDA) which adjusts variances by replacing the minimum eigenvalue with the maximum one. Dong et al. [17] developed an eigen decomposition framework for multivariate GEDA and claimed that most variance scaling methods by then could be unified within their framework by applying different eigenvalue tuning strategies. Liu et al. [18] introduced principal component analysis into EDA (PCA-EDA) and tried to avoid premature convergence by regulating the maximum eigenvalue.

It is easily comprehensible that the efficiency of GEDA depends not only on its search scope, but also on its search directions. Unfortunately, it has been shown that, without fine intervention, the main search direction of GEDA tends to become perpendicular to the fitness improvement direction [15,19], which greatly reduces its search efficiency. To remedy this defect, some researchers made beneficial attempts by improving the estimation method for the covariance matrix. The covariance matrix adaptation evolution strategy (CMA-ES) [20], which can be considered as a special EDA, employs a sophisticated covariance matrix estimation method, where the rank- $\mu$ -update operator updates the covariance matrix using the weighted high-quality solutions in the current generation and the corresponding mean in the last generation. By this means, the variance along the gradient direction can be increased. Bosman et al. [19] proposed an

anticipated mean shift (AMS) operator which estimates the covariance matrix by shifting part of selected solutions along the anticipated gradient direction such that the main search direction can be corrected to a certain extent. Bosman et al. integrated AVS, SDR and AMS together and developed a powerful EDA variant known as AMaLGaM [19]. Ren et al. [21] improved the original AMS operator by directly shifting the mean of selected solutions and taking the shifted mean as the center when estimating the covariance matrix. Liang et al. [22] recently reported an enhanced GEDA, in which the inferior solutions in current generation are repaired and utilized to estimate the covariance matrix such that the search directions could be adaptively adjusted.

In addition to scaling variances and improving the covariance matrix estimation method, extensive efforts have been made to enhance the performance of EDA. Xu [23] combined EDA with chaos perturbation operator for the purpose of enhancing the population diversity. Chen et al. [24] proposed a fast interactive EDA which extracts user's preference on the decision variables from historical information to reduce the initial search space to a preferred subspace such that the search process can be accelerated. Fang et al. [25] developed a mean shift strategy to speed up the convergence of EDA. Zhou et al. [7] suggested combining EDA with cheap and expensive local search. Auger and Hansen [26] developed a restart CMA-ES with increasing population size (IPOP-CMAES). Although IPOP-CMAES was developed a decade years ago, it was recently reported that IPOP-CMAES is still competitive with many other state-of-the-art EAs proposed in recent years [27]. Karshenas et al. [28] investigated the effect of regularization method on the model learning process of GEDA. Santana et al. [29] tried to improve EDA with the help of new selection strategies. Instead of using Gaussian and histogram model, [30-32] adopted particle filter, Copula theory and probabilistic graphical model, respectively, to capture the distribution of good solutions. Aiming at seeking multiple solutions, the techniques of detecting promising areas [33] and niching [34,35] were introduced to EDA to enhance its performance on multimodal problems. Moreover, EDAs have been integrated with other EAs like particle swarm optimization (PSO) [36] and differential evolution (DE) [37] to fuse their

advantages together. Theoretical researches have also been done to characterize the behaviors of EDA. Rastegar [38] analyzed the convergence probability of two univariate EDAs and showed a sufficient condition for the convergence. Echegoyen et al. [39] comprehensively studied the relationship between the behaviors of EDA and the solution space of optimization problems.

To sum up, EDAs have been improved significantly in the past decades, but there are still some shortcomings. So far, most existing variance scaling strategies are able to adjust the search scope of GEDA, but can hardly change its search directions. This is one of the key issues that severely restrict the performance of GEDA, but has not been fully recognized and studied. Consequently, few related work was reported in recent years. CMA-ES and AMaLGaM achieve great success by comprehensively regulating their search scopes and directions. However, their algorithmic frameworks are so complex that it is too difficult for practitioners to understand the mechanisms therein, not to mention setting corresponding parameters. As for some other EDA variants, their performance also highly depends on the search efficiency of basic EDA, then it is hopeful that they would be further improved if the efficiency of EDA can be increased.

Taking enhancing EDA with some simple operations as the goal, this paper proposes an anisotropic adaptive variance scaling (AAVS) strategy and develops a novel GEDA variant named as AAVS-EDA. Different from most existing variance scaling strategies which adjust all the variances by a same factor, AAVS first captures the landscape of the problem being solved along each eigendirection with a simple topology-based detection method, then adaptively tunes the variances along different directions according to the corresponding detection results. If a slope is detected along an eigendirection, then AAVS enlarges the corresponding variance. On the contrary, if a valley is detected, then AAVS keeps the corresponding variance unchanged. By this means, the search speed along a slope can be quickened, and the fine search around a valley can be achieved. More importantly, profiting from its anisotropic scaling, AAVS is able to make the main search direction of EDA naturally become consistent with the fitness improvement direction.

To ensure convergence, AAVS-EDA also adopts an auxiliary global monitor which shrinks all the variances if no improvement is achieved in a generation. Thanks to these fine properties, AAVS-EDA shows desirable performance on a variety of benchmark functions.

The remainder of this paper is organized as follows. Section 2 briefly reviews the basic knowledge of GEDA. Section 3 describes AAVS and the resulting AAVS-EDA in detail. Section 4 presents the experiment settings and analyzes the experiment results. The conclusions are finally drawn in Section 5.

## 2. Basic Knowledge of GEDA

As a model-based EA, EDA assumes that good solutions approximately obey a certain probability distribution over the solution space. During the search process, it tries to learn this distribution and generate new solutions according to the learning results [2-4]. The general framework of EDA is outlined in

### Algorithm 1.

EDA usually starts with an initial population which is filled with some randomly generated solutions. After the evaluation, those relatively good solutions are selected generally according to a truncation selection rule. Then a new probability model is built to produce solutions for the next generation. EDA executes this iterative process of evaluation, selection, model building, and solution sampling until meeting the stopping criterion.

---

**Algorithm 1.** The general framework of EDA.

---

- 1: Initialize parameters, set  $t = 0$  and generate initial population  $P^t$ ;
  - 2: Evaluate population  $P^t$  and update the best solution  $b^t$  obtained so far;
  - 3: Output  $b^t$  if the stopping criterion is met;
  - 4: Select promising solutions  $S^t$  from  $P^t$ ;
  - 5: Build a probability model  $G^{t+1}$  based on  $S^t$ ;
  - 6: Generate a new population  $P^{t+1}$  by sampling from  $G^{t+1}$ ;
  - 7: Set  $t \leftarrow t + 1$ , goto step 2.
-



Continuous EDAs generally employ Gaussian model as the basic probability distribution model. The Gaussian probability density function for an  $n$ -dimensional random vector  $\mathbf{x}$  can be parameterized by its mean  $\boldsymbol{\mu}$  and covariance matrix  $\mathbf{C}$  as follows:

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

The mean vector and the covariance matrix totally have  $0.5n^2 + 1.5n$  free parameters, which are generally estimated from the selected solutions using the maximum likelihood (ML) estimation method. For a selected solution set  $\mathbf{S}$ , the 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 Eqs. (2) and (3) takes the dependencies between all pairs of variables into account. It ensures rotation-invariance and is capable of capturing some complex structure characteristics of the solution space [19], hence is widely used by existing GEDAs.

Once  $\bar{\boldsymbol{\mu}}$  and  $\bar{\mathbf{C}}$  are obtained, the search characteristics of GEDA are determined. They can be geometrically described by a probability density ellipsoid (PDE) related with  $\bar{\boldsymbol{\mu}}$  and  $\bar{\mathbf{C}}$  [40]. As shown in Fig. 1, PDE takes  $\bar{\boldsymbol{\mu}}$  as its center which is also the search center of GEDA. The axis directions of PDE are accordant with the eigendirections of  $\bar{\mathbf{C}}$ . They determine the search directions of GEDA. The axis lengths of PDE, i.e., the standard deviations along eigendirections, equal the square roots of corresponding eigenvalues. They determine the search scope of GEDA.

From Fig. 1, it can be seen that the good solutions selected in each generation are mainly located within a semiellipsoid which is the intersection of the current PDE and a function contour surface determined by the selection rule. The major axis of this semiellipsoid tends to parallel to the corresponding function contour surface, so does the major axis of the PDE estimated by the solutions within the semiellipsoid. This means

that the major axis of the new PDE, i.e., the main search direction of GEDA, will gradually become perpendicular to the fitness improvement direction. On the other hand, the solutions within the semiellipsoid are mainly distributed in the region near the center of the current PDE, then the new PDE estimated by these solutions will naturally shrink. With improper search directions and rapid shrinking search scope, traditional GEDA certainly cannot achieve desirable optimization results.

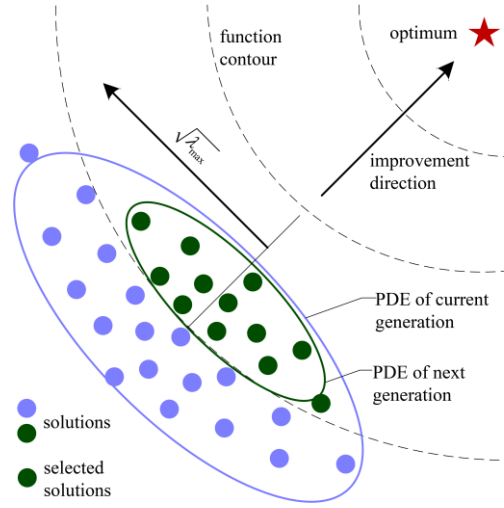


Fig.1. Schematic for the change of PDE in traditional GEDA

Based on the above explanation, it is easy to understand that if we enlarge the search scope with most existing AVS strategies which generally multiply the covariance matrix by a factor [13], the relative magnitudes of eigenvalues remain unchanged, then the ill-shaped PDE cannot be adjusted. EEDA and PCA-EDA achieve variance scaling by regulating the minimum eigenvalue and the maximum one, respectively. They can indeed adjust the shape of PDE, but may waste part of search effort in the futile search directions or make the main search direction too aggressive.

### 3. GEDA with Anisotropic Adaptive Variance Scaling

The proposed AAVS-EDA follows the basic framework of EDA, but contains two distinctive algorithmic strategies, i.e., AAVS and an auxiliary global monitor. This section will present them in sequence and then give the complete procedure of AAVS-EDA.

### 3.1 Anisotropic adaptive variance scaling

From the discussion in section 2, it can be known that the reason for the low efficiency of most existing AVS strategies lies in that they emphasize adjusting the search scope of GEDA, but consider little about its search directions. Different from them, AAVS takes both search elements into account. It is a common view that the local landscape of a problem can be characterized by a slope or a valley [12-14], and if a slope is detected, enlarging variances will facilitate quickening the search process. However, it is improper to brusquely claim an algorithm is searching a slope or a valley. Taking the situation shown in Fig. 1 as an example, GEDA is searching a slope along the minor axis of PDE, but is searching a valley along the major axis since the solutions along this axis become worse as they move away from the center. Therefore, it makes sense that the variances along different eigendirections should be adaptively adjusted according to corresponding landscapes. AAVS achieves this in the following way:

First, eigendirections are obtained by conducting eigendecomposition on the initially estimated covariance matrix  $\bar{C}$ . From Eq. (3), it can be seen that  $\bar{C}$  is a real symmetric matrix, then it is nonnegative definite and can be decomposed as follows:

$$\bar{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T \quad (4)$$

where  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  with each column  $\mathbf{v}_i (i=1, 2, \dots, n)$  representing a eigenvector of  $\bar{C}$ ,  $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  with each  $\lambda_i (i=1, 2, \dots, n)$  representing the eigenvalue corresponding to  $\mathbf{v}_i$ . According to the property of Gaussian distribution, it can be known that  $\lambda_i$  equals the variance along the direction of  $\mathbf{v}_i$ .

To detect the local landscape of a problem, several detection methods have been developed [33-35,41]. The most widely used one is the topology-based hill-valley method [41]. Its key idea consists in that, for a line segment connecting two solution points  $\mathbf{r}$  and  $\mathbf{l}$ , if there exists a point  $\mathbf{m}$  on the line segment  $\mathbf{r}\mathbf{l}$  whose fitness value is better than those of both  $\mathbf{r}$  and  $\mathbf{l}$ , then a valley along the line segment  $\mathbf{r}\mathbf{l}$  is identified. However, it is impractical to directly apply hill-valley method in AAVS since it requires many sample

points. Considering that the estimated mean  $\bar{\mu}$  lies in the center of the PDE and is the common point connecting all the eigendirections, AAVS examines the solution points on both sides of  $\bar{\mu}$  along each eigendirection instead of examining the interior points connecting two given points. Concretely, for each eigendirection  $\mathbf{v}_i (i=1, 2, \dots, n)$ , AAVS generates two symmetrical points centered on  $\bar{\mu}$  as follows:

$$\begin{cases} \mathbf{l}_i = \bar{\mu} - \delta_i \mathbf{v}_i \\ \mathbf{r}_i = \bar{\mu} + \delta_i \mathbf{v}_i \end{cases} \quad (5)$$

where  $\delta_i$  denotes the sampling step size along  $\mathbf{v}_i$ . Taking  $\bar{\mu}$  as an interior point connecting  $\mathbf{l}_i$  and  $\mathbf{r}_i$ , AAVS considers a slope is identified along  $\mathbf{v}_i$  if  $\mathbf{l}_i$ ,  $\mathbf{r}_i$  and  $\bar{\mu}$  satisfy:

$$\min\{f(\mathbf{l}_i), f(\mathbf{r}_i)\} < f(\bar{\mu}) < \max\{f(\mathbf{l}_i), f(\mathbf{r}_i)\} \quad (6)$$

where  $f(\cdot)$  represents the fitness function of the problem being solved.

According to some preliminary experimental results, it is found that the sampling step size  $\delta_i$  has great influence on the structure learning results and it is difficult to set a fixed value to adapt different eigendirections and different search situations. To alleviate this issue, AAVS sets  $\delta_i$  in a random way, i.e., samples a value for each eigendirection  $\mathbf{v}_i$  from the corresponding single dimensional Gaussian distribution  $N(0, \lambda_i)$ . It is worth to mention that, with this setting, AAVS cannot ensure to provide the right detection results every time, but this setting still makes sense that it facilitate AAVS getting helpful structure information in most cases since AAVS mainly focuses on examining local structure of a single dimensional subspace and the randomly generated  $\delta_i$  can cover the local solution region well.

For each eigendirection  $\mathbf{v}_i$ , if a slope is identified, then AAVS enlarges the corresponding eigenvalue  $\lambda_i$  for the purpose of encouraging further exploration and quickening the search process. On the other hand, if a valley is identified, AAVS keeps the corresponding eigenvalue unchanged, so that the excellent performance of GEDA on valley-like regions can be preserved. Formally, each  $\lambda_i (i=1, 2, \dots, n)$ , i.e., the variance along each eigendirection, is tuned as follows:

$$\lambda_i \leftarrow \begin{cases} \alpha \cdot \lambda_i, & \text{if a slope is detected along } \mathbf{v}_i \\ \lambda_i, & \text{otherwise} \end{cases} \quad (7)$$

where  $\alpha$  is a scaling factor greater than 1. A smaller value of  $\alpha$  makes no significant influence. On the contrary, a larger value of  $\alpha$  encourages faster exploration speed along an eigendirection, but may cause serious imbalance among different directions. We will discuss the influence of  $\alpha$  in detail in section 4.1.

---

**Algorithm 2.** The pseudocode of AAVS

---

- 1: Perform eigendecomposition on  $\bar{\mathbf{C}} : \bar{\mathbf{C}} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ ;
  - 2: Evaluate  $\bar{\boldsymbol{\mu}}$ ;
  - 3: **For** each eigenvector  $\mathbf{v}_i (i = 1, 2, \dots, n)$
  - 4:   Sample step size  $\delta_i$  from  $N(0, \lambda_i)$ ;
  - 5:   Sample two solutions  $\mathbf{l}_i$  and  $\mathbf{r}_i$  according to Eq. (5) and evaluate them;
  - 6:   Detect local landscape based on  $\mathbf{l}_i$ ,  $\mathbf{r}_i$  and  $\bar{\boldsymbol{\mu}}$  and Eq. (6);
  - 7:   **If** a slope is identified
  - 8:      $\lambda_i \leftarrow \alpha \cdot \lambda_i$ ;
  - 9:   **End if**
  - 10: **End for**
- 

**Algorithm 2** lists the pseudocode of AAVS, where three remarks need to be highlighted. First, AAVS tunes all the variances independently and adaptively according to the detection results along corresponding eigendirections. This is also the reason why we name it anisotropic adaptive variance scaling. Second, a single run of AAVS does not change the eigendirections of  $\bar{\mathbf{C}}$ , but running AAVS in successive generations can reshape PDE such that the major axis of PDE would gradually become consistent with the fitness improvement direction. The main reason lies in that slopes provide the main fitness improvement directions, AAVS enlarges the axes along slopes, and the new sampled solutions around these axes are more likely to be selected to estimate the new covariance matrix, which further makes this kind of axes approach to the fitness improvement directions. As a result, the search efficiency of GEDA can be substantially increased. Finally, the excellent performance of AAVS is obtained at the cost of sampling two additional solutions for detecting the local landscape along each eigendirection. Nevertheless, the additional sampling quantity is acceptable compared with the general setting of population size in multivariate GEDA.

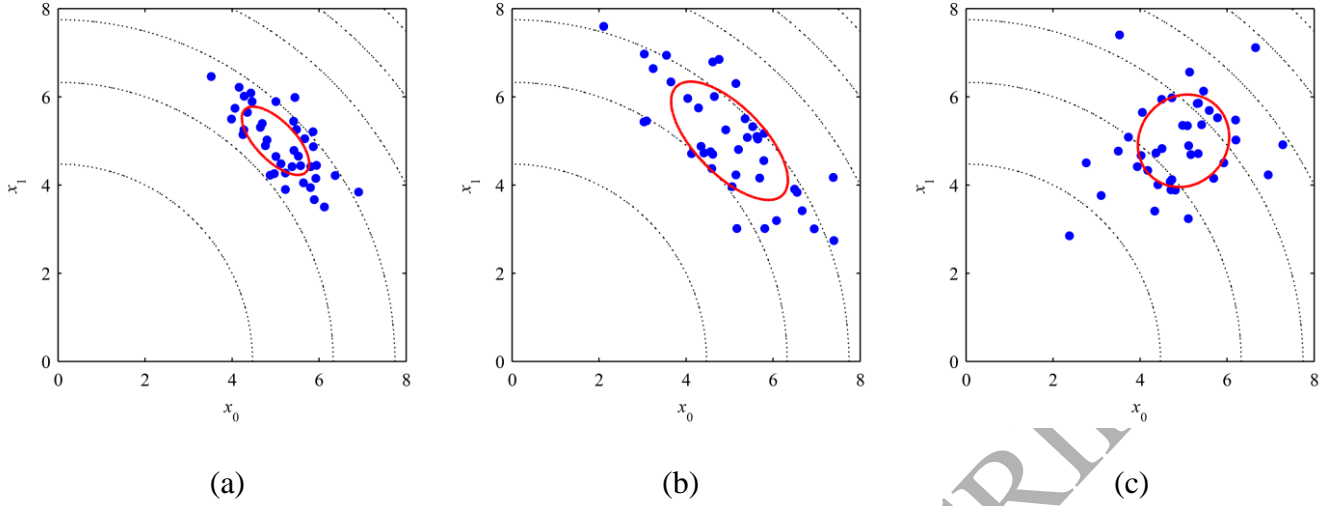


Fig. 2. PDE and the generated solutions: (a) current PDE, (b) PDE tuned by traditional AVS and (c) PDE tuned by AAVS.

To demonstrate the effectiveness of AAVS in adjusting PDE, a schematic experiment is designed. Suppose GEDA is employed to minimize a two dimensional function  $f(\mathbf{x}) = x_0^2 + x_1^2$  and the search range is set to  $[0,8] \times [0,8]$ . Fig. 2(a) shows the contour map of this test function, together with the PDE of the current Gaussian distribution and the solutions generated by it. It is clear that the fitness landscape along the direction  $x_0 = x_1$  is a slope, the one along the direction  $x_0 = -x_1$  is a valley, and the current PDE is ill-shaped. After applying the traditional AVS and the new proposed AAVS, the resulting PDEs and the new solutions generated by them are shown in Fig. 2(b) and Fig. 2(c). It can be seen from Fig. 2(b) that when traditional AVS are employed, variances along both directions are increased. As a result, the exploration range of GEDA is enlarged, but most solutions are generated along the direction  $x_0 = -x_1$  since the PDE is still ill-shaped. When AAVS is applied as shown in Fig. 2(c), only the variance along the direction  $x_0 = x_1$  is increased since a slope can be detected along this direction. As a result, PDE is reshaped to a certain extent and more solutions are generated along the fitness improvement direction.

### 3.2 Global monitor

As illustrated above, AAVS indeed facilitates enhancing the search efficiency of GEDA. However, it is found that if exclusively performed, AAVS may sometimes cause GEDA unable to converge. The reason is twofold: First, AAVS occasionally provides the wrong detection results since it makes the detection decision just based on three solution points. Second, the optimal solutions of some problems are located at the intersection between a slope and the search boundary. For both cases, enlarging variances is ineffectual, especially at the late searching stage.

As it is difficult to overcome the defects mentioned above point to point, here we remedy them with a global monitor. Its key idea lies in that no matter under what circumstance the variances are improperly enlarged, GEDA could hardly get better solutions, then we just need to reduce all the variances in these cases. To be specific, the global monitor first calculates the average fitness value of the selected solutions (AFV) in each generation as follows:

$$AFV = \frac{1}{|S|} \sum_{i=1}^{|S|} f(S_i) \quad (8)$$

then shrinks all the variances if AFV is not improved compared with the one in the last generation:

$$\mathbf{D} \leftarrow \begin{cases} \mathbf{D}, & \text{if AFV is improved} \\ \beta \cdot \mathbf{D}, & \text{otherwise} \end{cases} \quad (9)$$

where  $\beta$  is a shrinking factor. Considering that AAVS enlarges the variances along slopes by a factor of  $\alpha$ , then the convergence of GEDE can be guaranteed if  $\beta$  is limited within  $(0, 1/\alpha]$ . We suggest setting  $\beta = 1/\alpha$  for simplicity. It is notable that the selected solutions have been evaluated at the selection phase, thus the calculation of AFV is very straightforward. In addition, compared with AVS that just examines the improvement of the best solution obtained so far, AFV can provide much more stable result. Fig. 3 illustrates the relationship between AAVS and the global monitor. For an decomposed covariance matrix  $\bar{\mathbf{C}}$ , its eigenvalue matrix  $\mathbf{D}$  is first adjusted by AAVS. Then the global monitor checks whether AFV is improved. If it is not, the global monitor shrinks  $\mathbf{D}$  before outputting it.

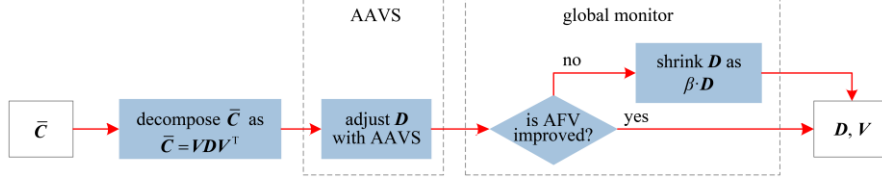


Fig. 3. The relationship between AAVS and the global monitor.

### 3.3 The procedure of AAVS-EDA

AAVS and the global monitor can be easily integrated into the traditional GEDA. The detailed procedure of the resulting EDA variant AAVS-EDA is described in **Algorithm 3**. It should be noted that AAVS-EDA employs the commonly used truncation selection rule in step 5, and just generates  $p-1$  new solutions for a population of size  $p$  in step 10 since it maintains the best solution in each generation to the next generation. It can be known that the computation complexity of AAVS and the global monitor is  $O(n)$  while the computation complexity for estimating the covariance matrix is  $O(n^2)$ . Therefore, the overall computation complexity of AAVS-EDA is not greater than that of the traditional multivariate GEDAs.

---

**Algorithm 3.** The procedure of AAVS-EDA.

---

- 1: Initialize population size  $p$ , truncation ratio  $\tau$  and scaling factor  $\alpha$ ;
  - 2: Set  $t = 0$ , generate initial population  $\mathbf{P}^t$  randomly;
  - 3: Evaluate population  $\mathbf{P}^t$  and update the best solution  $\mathbf{b}^t$  obtained so far;
  - 4: Output  $\mathbf{b}^*$  if the stopping criterion is met;
  - 5: Select the best  $\lfloor \tau p \rfloor$  solutions in  $\mathbf{P}^t$  and store them in  $\mathbf{S}^t$ ;
  - 6: Estimate mean  $\bar{\boldsymbol{\mu}}^{t+1}$ , covariance matrix  $\bar{\mathbf{C}}^{t+1}$  according to  $\mathbf{S}^t$  and Eqs. (2) and (3);
  - 7: Tune  $\bar{\mathbf{C}}^{t+1}$  according to AAVS described by **Algorithm 2**;
  - 8: Tune  $\bar{\mathbf{C}}^{t+1}$  according to the global monitor described by Eqs. (8) and (9);
  - 9: Build a probability model  $G^{t+1}$  based on  $\bar{\boldsymbol{\mu}}^{t+1}$  and  $\bar{\mathbf{C}}^{t+1}$ ;
  - 10: Generate  $p-1$  new solutions by sampling from  $G^{t+1}$  and store them into  $\mathbf{A}^{t+1}$ ;
  - 11: Set  $\mathbf{P}^{t+1} \leftarrow \mathbf{A}^{t+1} \cup \mathbf{b}^t$ ;
  - 12: Set  $t \leftarrow t + 1$ , goto step 3.
-



#### 4. Experimental study

This section aims to experimentally investigate the influences of parameters, the effectiveness of AAVS and the auxiliary global monitor, and the efficiency of AAVS-EDA. To achieve this, CEC2014 test suite was employed [42]. It contains 30 benchmark functions (denoted as  $f_1$ - $f_{30}$ ) which can be classified into four kinds:  $f_1$ - $f_3$  are unimodal functions,  $f_4$ - $f_{16}$  are simple multimodal functions,  $f_{17}$ - $f_{22}$  are hybrid functions and  $f_{23}$ - $f_{30}$  are composition functions. These functions are all single-objective minimization problems and their detailed description can be found in the literature [42]. In our experiments, the dimension of each test function was set to 30, the allowed maximum number of function evaluations (FEs) was conventionally set to 300,000, 25 independent runs were conducted for each function, and the performance of an algorithm on each function was evaluated 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 optimal solution. Note that, for AAVS-EDA, the number of FEs consumed by AAVS was also counted into the allowed FE number, and for all the algorithms tested, the FEVs smaller than  $10^{-8}$  were seen as zero.

##### 4.1 Influences of parameters

There are three parameters in AAVS-EDA, including population size  $p$ , selection ratio  $\tau$  and the enlarging factor  $\alpha$ . As for  $p$  and  $\tau$ , they are common parameters of most GEDAs and their influences have been extensively studied [4,19]. In AAVS-EDA, we directly set them as  $p = 1000$  and  $\tau = 0.35$  which are conventional values adopted by other GEDAs. This section mainly focuses on investigating the influence of enlarging factor  $\alpha$ . We tested AAVS-EDA on a variety of benchmark functions with different  $\alpha$  values varying from 1.5 to 2.5 at an interval of 0.1. Taking unimodal function  $f_2$ , multimodal function  $f_6$  and hybrid function  $f_{18}$  as examples, Fig. 4 shows the performance variation of AAVS-EDA with respect to  $\alpha$ .

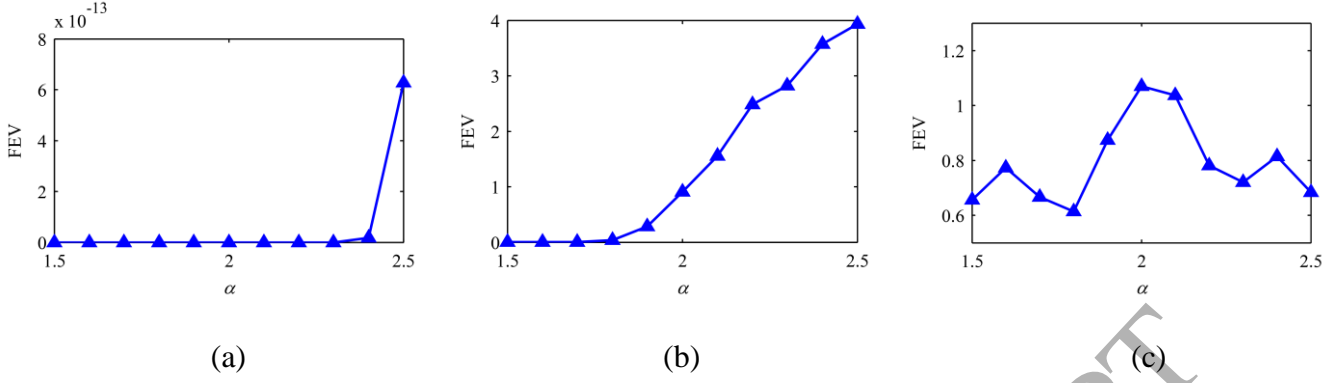


Fig. 4. Performance of AAVS-EDA with different  $\alpha$  values on (a)  $f_2$ , (b)  $f_6$  and (c)  $f_{18}$

From Fig. 4, three observations can be made: 1) When  $\alpha$  changes from 1.5 to 2.5, the variations of average FEVs obtained by AAVS-EDA on each function are all within an order of magnitude, which means that the performance of AAVS-EDA has certain robustness to the change of  $\alpha$ ; 2) AAVS-EDA performs pretty well on  $f_2$  and  $f_6$  within a wide range of  $\alpha$ , but performs slightly worse when  $\alpha$  becomes larger than 2.3 and 1.8, respectively; 3) The performance of AAVS-EDA on the hybrid function  $f_{18}$  does not show significant variation trend when  $\alpha$  varies. In order to balance the performance of AAVS-EDA on different kinds of functions, we suggest setting  $\alpha$  within the range (1.5, 1.8). In the experiments described below, it was set to 1.7.

#### 4.2 Effectiveness of AAVS and the global monitor

To verify the effectiveness of AAVS and the auxiliary global monitor, we compared AAVS-EDA with EMNA<sub>g</sub> [4] which is a representative of traditional multivariate GEDA. Both algorithms share the same algorithmic components except for AAVS and the auxiliary global monitor. To achieve a fair comparison, the population size and the selection ratio of EMNA<sub>g</sub> were all set the same as those of AAVS-EDA. Besides FEV, another two indicators were adopted to show the performance differences between AAVS-EDA and EMNA<sub>g</sub>, including the length of PDE's major axis and the cosine value of the acute angle (denoted as  $\theta$ ) between PDE's major axis and the steepest descent direction of the test function. Here the steepest descent

direction is defined as the direction from PDE's center to the global optimal solution. These two new indicators reflects the search scope and the main search direction of GEDA to a great extent, respectively.

Figs. 5 ~ 7 present the corresponding evolution curves, where functions  $f_2$ ,  $f_6$  and  $f_{18}$  are taken as examples.

It can be observed from Fig. 5 that, for all the three functions, the length of the major axis derived from AAVS-EDA is always larger than that of  $EMNA_g$  in their early search stages, which means that AAVS indeed endows AAVS-EDA with stronger exploration ability. This facilitates AAVS-EDA keeping larger search scope and locking the global optimal solution region. For functions  $f_2$  and  $f_6$ , AAVS-EDA rapidly shrinks its search scope in the middle search stage since no slope can be detected any more, which enables AAVS-EDA to conduct fine search within the locked solution regions and finally get desirable solutions. From Fig. 5(c), it can be seen that AAVS-EDA keeps a larger search scope throughout the entire search process for the hybrid function  $f_{18}$ . The main reason lies in that the fitness landscape of  $f_{18}$  is so complex that AAVS-EDA cannot find a definitely promising solution region with relatively limited number of FEs. Nevertheless, the final best solution it obtained is still much better than the one obtained by  $EMNA_g$ .

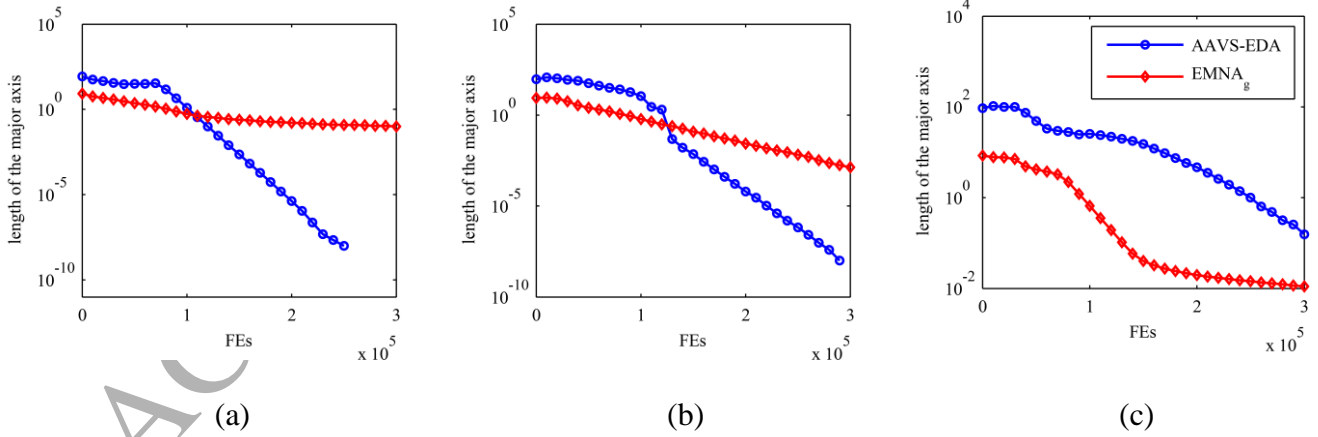


Fig. 5. Evolution of the length of PDE's major axis derived from AAVS-EDA and  $EMNA_g$  on (a)  $f_2$ , (b)  $f_6$  and (c)  $f_{18}$ .

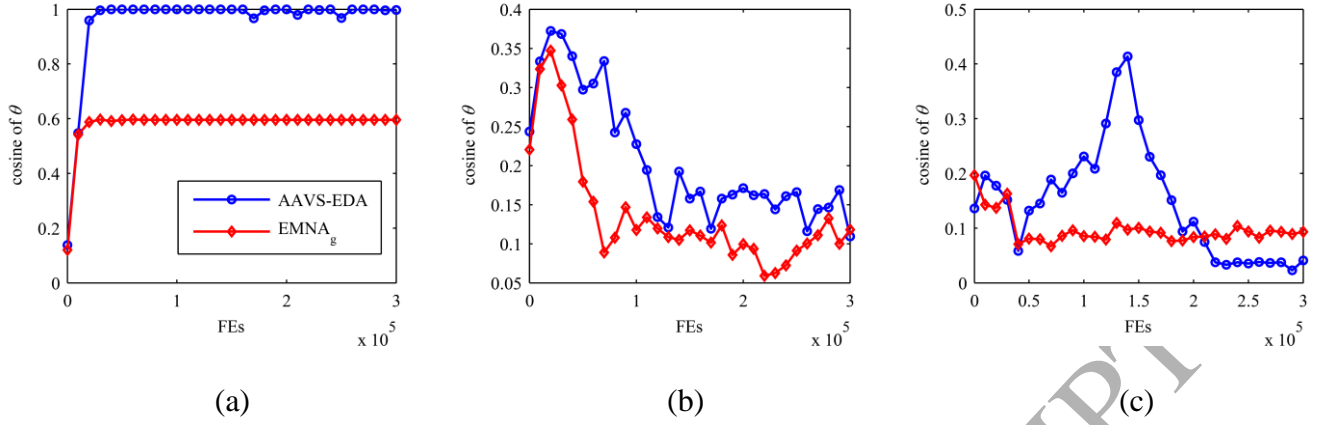


Fig. 6. Evolution of the cosine value of  $\theta$  derived from AAVS-EDA and EMNA<sub>g</sub> on (a)  $f_2$ , (b)  $f_6$  and (c)  $f_{18}$ .

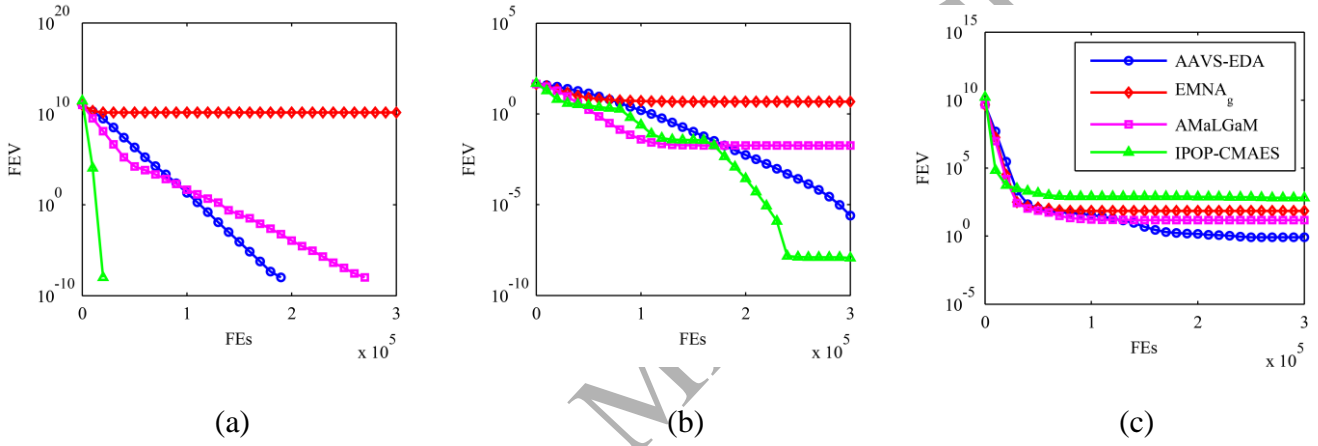


Fig. 7. Evolution of FEV derived from AAVS-EDA, EMNA<sub>g</sub>, AMaLGaM and IPOP-CMAES on (a)  $f_2$ , (b)  $f_6$  and (c)  $f_{18}$ .

Fig. 6 shows the variation of  $\cos(\theta)$  during the evolution process. It is clear that, for all the three functions, AAVS-EDA achieves larger cosine values than EMNA<sub>g</sub> in most cases, which means that AAVS indeed enables AAVS-EDA to identify better search direction. Especially for  $f_2$ , AAVS-EDA finds the steepest descent direction after only a few generations. As for  $f_{18}$ , AAVS-EDA also detects better search direction than EMNA<sub>g</sub> in the early and middle search stages. However, in the late search stage, both algorithms are trapped into local optimal regions, where the main search direction becomes less meaningful.

Fig. 7 shows the evolution curve of FEV, from which it can be observed that AAVS-EDA significantly outperforms EMNA<sub>g</sub> on all the three functions. Besides EMNA<sub>g</sub> and AAVS-EDA, Fig. 7 also presents the variation of FEVs obtained by AMaLGaM[19] and IPOP-CMAES[26]. As mentioned in section 1, AMaLGaM is an efficient EDA variant which synthetically adopts the strategies of AMS, AVS and SDR. IPOP-CMAES is an improved CMA-ES variant, which is competitive to many state-of-the-art EAs [27]. To ensure the fairness of the comparison, the AMaLGaM version which adopts the multivariate Gaussian model and a fixed population size was employed in our experiment. Its population size was set the same as that of AAVS-EDA, and all the other parameters were set the same as in [19]. As for IPOP-CMAES, we employed the source code and the default parameter settings provided by the authors of [43, 44]. It can be seen from Fig. 7(a) that AAVS-EDA, AMaLGaM and IPOP-CMAES all perform pretty well on  $f_2$ , where IPOP-CMAES finds the global optimum with much fewer FEs. As for  $f_6$ , it can be seen from Fig. 7(b) that AMaLGaM gets stuck earlier; IPOP-CMAES improves AMaLGaM and obtains the best solution within the allowed maximum number of FEs, but it still falls into a local optimum finally; Different from these two algorithms, AAVS-EDA keeps an improvement tendency during the whole evolution process. Fig. 7(c) presents the results on hybrid function  $f_{18}$ , from which it can be observed that all the four algorithms are trapped into local optimal solutions. In spite of this, AAVS-EDA demonstrates satisfying performance since it gets stuck at the latest and finds the best solution among these algorithms tested. It is notable that AAVS-EDA still finds better solutions for  $f_{18}$  after consuming about half of FEs. The reason is that it has strong exploration ability and identifies a relatively good search direction at that time, which can be clearly illustrated by Fig. 6(c).

#### 4.3 Comparison with other state-of-the-art EAs

To evaluate the performance of AAVS-EDA, this section compares it with EMNA<sub>g</sub> [4], AMaLGaM [19], IPOP-CMAES [26], CPI-JADE [45] and HHSPSO-GDS [46]. CPI-JADE is a new variant of DE, it

improves the performance of classic JADE [47] by implementing the crossover operator in both the original coordinate space and the eigen coordinate space, where the latter is established by decomposing the covariance matrix which is estimated from the cumulative population distribution information with the covariance matrix adaptation strategy developed in CMA-ES. In this sense, CPI-JADE has suitable comparability with AAVS-EDA. HHSPSO-GDS integrates the harmony search algorithm and the global dimension selection strategy into PSO to enhance its exploration and exploitation abilities. Experimental results revealed that HHSPSO-GDS possesses superior performance over many other PSO variants [46]. In short, AMaLGaM, IPOP-CMAES, CPI-JADE and HHSPSO-GDS can be viewed as the state-of-the-art of four kinds of EAs for continuous optimization, i.e., EDA, ES, DE and PSO, respectively. Table 1 reports the mean and standard deviation of FEVs obtained by the 6 algorithms on 30 benchmark functions, where the results of CPI-JADE and HHSPSO-GDS are directly taken from their original papers. Cohen's  $d$  effect size [48] is adopted to judge the performance difference between AAVS-EDA and its five competitors. Cohen's  $d$  effect size is a popular statistical method for quantifying the difference between two groups of data. It is independent of the sample size and is generally considered "small", "medium", and "large" if its absolute value belongs to  $[0.2, 0.3)$ ,  $[0.3, 0.8)$  and  $[0.8, +\infty)$ , respectively. From Table 1, it can be observed that:

1) Unimodal functions  $f_1$ - $f_3$ : AAVS-EDA obtains the global optima for three unimodal functions, so do AMaLGaM, IPOP-CMAES and CPI-JADE. The results show that the combination of AAVS and the global monitor results in fine search performance for AAVS-EDA. As for the other two algorithms, HHSPSO-GDS shows better performance than EMNA<sub>g</sub>, but they both are outperformed by AAVS-EDA on all the three test functions.

2) Simple multimodal functions  $f_4$ - $f_{16}$ : Among these 13 functions, AAVS-EDA has an edge over AMaLGaM, IPOP-CMAES, CPI-JADE and HHSPSO-GDS on 9, 4, 4 and 7 functions, respectively, but is defeated by them on 2, 7, 8 and 6 functions, respectively. As a whole, AAVS-EDA performs a little worse

than IPOP-CMAES and CPI-JADE on this group of functions, but achieves better performance than EMNA<sub>g</sub>, AMaLGaM and HHSPSO-GDS.

3) Hybrid functions  $f_{17}$ - $f_{22}$ : It is obvious that AAVS-EDA performs best among six algorithms on this group of test functions. The results it obtained are all not worse than the ones obtained by the other five algorithms except that HHSPSO-GDS achieves a better solution for  $f_{19}$ . Especially, AAVS-EDA reduces the mean of FEVs for  $f_{18}$  by at least two orders of magnitude.

Table 1. The mean and the standard deviation (mean $\pm$ standard deviation) of FEVs obtained by 6 algorithms over 25 independent runs on 30 CEC2014 benchmark functions with 30D

| Fun.            | EMNA <sub>g</sub>                 | AMaLGaM                           | IPOP-CMAES                        | CPI-JADE                          | HHSPSO-GDS                        | AAVS-EDA                |
|-----------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-------------------------|
| $f_1$           | 1.43E+08 $\pm$ 3.30E+07 –         | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 8.60E+03 $\pm$ 4.67E+03 –         | 0.00E+00 $\pm$ 0.00E+00 |
| $f_2$           | 1.39E+10 $\pm$ 1.71E+09 –         | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 1.14E+02 $\pm$ 1.25E+02 –         | 0.00E+00 $\pm$ 0.00E+00 |
| $f_3$           | 1.95E+04 $\pm$ 3.48E+03 –         | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 4.64E+02 $\pm$ 3.83E+02 –         | 0.00E+00 $\pm$ 0.00E+00 |
| $f_4$           | 1.53E+03 $\pm$ 2.45E+02 –         | 4.94E+00 $\pm$ 1.19E+01 –         | 0.00E+00 $\pm$ 0.00E+00 +         | 0.00E+00 $\pm$ 0.00E+00 +         | 5.92E+00 $\pm$ 4.01E+00 –         | 1.08E–02 $\pm$ 4.41E–02 |
| $f_5$           | 2.09E+01 $\pm$ 5.19E–02 $\approx$ | 2.09E+01 $\pm$ 2.79E–02 $\approx$ | 2.09E+01 $\pm$ 4.97E–02 $\approx$ | 2.03E+01 $\pm$ 3.68E–02 +         | 1.13E+01 $\pm$ 3.79E–02 +         | 2.09E+01 $\pm$ 4.66E–02 |
| $f_6$           | 4.62E+00 $\pm$ 1.13E+00 –         | 1.83E–02 $\pm$ 9.18E–02 –         | 1.18E–08 $\pm$ 1.23E–08 +         | 3.44E+00 $\pm$ 3.57E+00 –         | 1.20E+01 $\pm$ 2.45E+00 –         | 2.11E–06 $\pm$ 5.82E–06 |
| $f_7$           | 1.65E+02 $\pm$ 1.91E+01 –         | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 0.00E+00 $\pm$ 0.00E+00 $\approx$ | 2.31E–04 $\pm$ 2.30E–04 –         | 0.00E+00 $\pm$ 0.00E+00 |
| $f_8$           | 4.14E+01 $\pm$ 5.94E+00 –         | 4.17E+00 $\pm$ 1.69E+00 –         | 9.21E–01 $\pm$ 9.01E–01 +         | 0.00E+00 $\pm$ 0.00E+00 +         | 0.00E+00 $\pm$ 0.00E+00 +         | 1.15E+00 $\pm$ 1.13E+00 |
| $f_9$           | 3.01E+01 $\pm$ 7.28E+00 –         | 3.66E+00 $\pm$ 1.36E+00 –         | 8.74E–01 $\pm$ 1.06E+00 +         | 2.24E+01 $\pm$ 5.33E+00 –         | 1.35E+01 $\pm$ 3.61E+01 –         | 2.46E+00 $\pm$ 9.98E–01 |
| $f_{10}$        | 1.02E+03 $\pm$ 3.70E+02 –         | 5.29E+02 $\pm$ 2.44E+02 –         | 2.74E+02 $\pm$ 2.32E+02 –         | 3.83E–01 $\pm$ 7.47E–02 +         | 2.39E+00 $\pm$ 1.30E+00 +         | 1.36E+02 $\pm$ 1.83E+02 |
| $f_{11}$        | 1.16E+03 $\pm$ 3.43E+02 –         | 5.66E+02 $\pm$ 3.24E+02 –         | 2.22E+02 $\pm$ 1.66E+02 +         | 1.77E+03 $\pm$ 2.55E+02 –         | 1.96E+03 $\pm$ 5.78E+02 –         | 3.02E+02 $\pm$ 3.68E+02 |
| $f_{12}$        | 2.49E+00 $\pm$ 2.26E–01 –         | 2.51E+00 $\pm$ 7.91E–01 –         | 2.49E+00 $\pm$ 2.26E–01 –         | 3.95E–01 $\pm$ 8.64E–02 +         | 1.06E–01 $\pm$ 1.34E–02 +         | 5.21E–01 $\pm$ 7.56E–01 |
| $f_{13}$        | 4.09E+00 $\pm$ 1.90E–01 –         | 5.85E–02 $\pm$ 1.39E–02 –         | 6.98E–02 $\pm$ 2.01E–02 –         | 2.04E–01 $\pm$ 3.38E–02 –         | 1.49E–01 $\pm$ 1.34E–01 –         | 2.03E–02 $\pm$ 3.37E–03 |
| $f_{14}$        | 7.92E+01 $\pm$ 9.35E+00 –         | 4.25E–01 $\pm$ 3.77E–02 –         | 3.80E–01 $\pm$ 5.24E–02 –         | 2.32E–01 $\pm$ 3.35E–02 +         | 1.51E–01 $\pm$ 2.04E–01 +         | 2.83E–01 $\pm$ 2.73E–02 |
| $f_{15}$        | 4.67E+01 $\pm$ 7.41E+01 –         | 3.21E+00 $\pm$ 4.70E–01 +         | 2.50E+00 $\pm$ 3.81E–01 +         | 3.26E+00 $\pm$ 3.78E–01 +         | 2.13E+01 $\pm$ 7.46E+00 –         | 6.73E+00 $\pm$ 1.98E+00 |
| $f_{16}$        | 1.32E+01 $\pm$ 1.59E–01 –         | 1.06E+01 $\pm$ 7.76E–01 +         | 9.47E+00 $\pm$ 1.82E+00 +         | 9.70E+00 $\pm$ 2.79E–01 +         | 1.06E+00 $\pm$ 6.62E–01 +         | 1.13E+01 $\pm$ 3.65E–01 |
| $f_{17}$        | 1.35E+06 $\pm$ 9.72E+05 –         | 1.97E+02 $\pm$ 1.17E+02 –         | 1.89E+03 $\pm$ 5.03E+02 –         | 1.16E+03 $\pm$ 3.81E+02 –         | 1.15E+05 $\pm$ 7.94E+04 –         | 1.20E+02 $\pm$ 8.76E+01 |
| $f_{18}$        | 7.04E+01 $\pm$ 1.54E+01 –         | 1.94E+01 $\pm$ 9.09E+00 –         | 6.33E+02 $\pm$ 2.16E+02 –         | 9.47E+01 $\pm$ 3.42E+01 –         | 1.21E+01 $\pm$ 9.78E+01 –         | 8.02E–01 $\pm$ 2.79E–01 |
| $f_{19}$        | 1.40E+01 $\pm$ 5.01E+00 –         | 3.97E+00 $\pm$ 7.05E–01 –         | 6.77E+00 $\pm$ 1.67E+00 –         | 4.89E+00 $\pm$ 7.64E–01 –         | 1.69E–01 $\pm$ 2.18E–01 +         | 3.35E+00 $\pm$ 4.33E–01 |
| $f_{20}$        | 2.70E+03 $\pm$ 1.24E+03 –         | 3.53E+00 $\pm$ 1.47E+00 –         | 2.44E+02 $\pm$ 1.93E+00 –         | 1.12E+01 $\pm$ 5.24E+00 –         | 2.09E+01 $\pm$ 1.19E+01 –         | 1.65E+00 $\pm$ 2.22E–01 |
| $f_{21}$        | 2.80E+04 $\pm$ 4.47E+04 –         | 1.10E+02 $\pm$ 9.45E+01 $\approx$ | 2.16E+03 $\pm$ 9.67E+02 –         | 3.33E+02 $\pm$ 1.54E+02 –         | 1.07E+05 $\pm$ 9.07E+04 –         | 1.05E+02 $\pm$ 2.75E+01 |
| $f_{22}$        | 1.28E+02 $\pm$ 5.61E+01 –         | 8.27E+01 $\pm$ 6.13E+01 $\approx$ | 2.36E+02 $\pm$ 1.48E+02 –         | 9.99E+01 $\pm$ 6.09E+01 $\approx$ | 1.27E+02 $\pm$ 2.32E+02 $\approx$ | 9.20E+01 $\pm$ 5.08E+01 |
| $f_{23}$        | 3.57E+02 $\pm$ 1.00E+01 –         | 3.15E+02 $\pm$ 2.07E–13 $\approx$ | 3.14E+02 $\pm$ 1.16E–13 +         | 3.15E+02 $\pm$ 4.01E–13 $\approx$ | 3.15E+02 $\pm$ 1.06E–04 $\approx$ | 3.15E+02 $\pm$ 7.03E–13 |
| $f_{24}$        | 2.30E+02 $\pm$ 2.96E+00 –         | 2.22E+02 $\pm$ 4.78E+00 $\approx$ | 2.22E+02 $\pm$ 2.70E+01 $\approx$ | 2.24E+02 $\pm$ 2.93E+00 –         | 2.00E+02 $\pm$ 0.00E+00 +         | 2.22E+02 $\pm$ 6.66E+00 |
| $f_{25}$        | 2.08E+02 $\pm$ 3.70E–01 –         | 2.02E+02 $\pm$ 4.73E–02 $\approx$ | 2.00E+02 $\pm$ 1.42E–04 +         | 2.03E+02 $\pm$ 5.77E–01 –         | 2.01E+02 $\pm$ 6.42E+00 $\approx$ | 2.02E+02 $\pm$ 6.83E–02 |
| $f_{26}$        | 1.05E+02 $\pm$ 6.76E+00 –         | 1.00E+02 $\pm$ 7.68E–02 $\approx$ | 1.00E+02 $\pm$ 2.39E–02 $\approx$ | 1.00E+02 $\pm$ 2.92E–02 $\approx$ | 1.00E+02 $\pm$ 4.68E–02 $\approx$ | 1.00E+02 $\pm$ 4.75E–03 |
| $f_{27}$        | 5.60E+02 $\pm$ 4.86E+01 –         | 3.02E+02 $\pm$ 1.26E+01 –         | 3.10E+02 $\pm$ 3.32E+01 –         | 3.53E+02 $\pm$ 5.03E+01 –         | 2.11E+02 $\pm$ 1.96E+02 +         | 3.00E+02 $\pm$ 5.97E–07 |
| $f_{28}$        | 9.63E+02 $\pm$ 1.43E+02 –         | 9.02E+02 $\pm$ 2.54E+01 –         | 3.66E+02 $\pm$ 4.14E–01 +         | 8.02E+02 $\pm$ 4.34E+01 –         | 2.36E+02 $\pm$ 3.08E+01 +         | 7.03E+02 $\pm$ 2.06E+02 |
| $f_{29}$        | 1.52E+03 $\pm$ 8.70E+02 –         | 6.43E+02 $\pm$ 1.59E+02 –         | 2.07E+02 $\pm$ 1.55E+00 –         | 8.13E+02 $\pm$ 7.12E+01 –         | 2.57E+02 $\pm$ 4.03E+02 –         | 1.00E+02 $\pm$ 5.19E–03 |
| $f_{30}$        | 3.86E+03 $\pm$ 1.08E+03 –         | 1.35E+03 $\pm$ 6.48E+02 –         | 6.63E+02 $\pm$ 2.02E+02 –         | 1.40E+03 $\pm$ 7.24E+02 –         | 1.12E+03 $\pm$ 1.32E+03 –         | 4.70E+02 $\pm$ 8.64E+01 |
| –/ $\approx$ /+ | 29/1/0                            | 17/11/2                           | 13/7/10                           | 15/7/8                            | 16/4/10                           | —                       |

“–”, “+” and “ $\approx$ ” respectively denote that the performance of the corresponding algorithm is worse than, better than or similar to that of AAVS-EDA according to Cohen’s  $d$  effect size.

4) Composition functions  $f_{23}$ - $f_{30}$ : The components of these composition functions are so sophisticated that it is hard to find even the near optimal solutions. In spite of this, AAVS-EDA demonstrates superior performance. It performs not worse than EMNA<sub>g</sub>, AMaLGaM and CPI-JADE on all the eight functions, and is defeated by IPOP-CMAES and HHSPSO-GDS on only three functions. It is notable that the solutions for  $f_{29}$  and  $f_{30}$  obtained by AAVS-EDA are not only better but also more stable than the ones obtained by the other five algorithms.

The last row of Table 1 lists the number of functions on which AAVS-EDA performs better than, similarly to or worse than each of its competitors. This result demonstrate that AAVS-EDA shows significant advantages over EMNA<sub>g</sub>, AMaLGaM and CPI-JADE, and slight advantages over IPOP-CMAES and HHSPSO-GDS. In addition, we analyzed the differences between AAVS-EDA and the other five algorithms with Friedman test. Table 2 lists the ranking of the six algorithms obtained by the Friedman test, from which it can be concluded that AAVS-EDA performs best, followed by IPOP-CMAES; HHSPSO-GDS, AMaLGaM and CPI-JADE show similar performance to each other, whereas EMNA<sub>g</sub> is definitely defeated by the other five algorithms. This conclusion is basically consistent with the one drawn according to Cohen's  $d$  effect size.

Table 2. Ranking of 6 algorithms according to the Friedman test.

| Algorithms        | Ranking |
|-------------------|---------|
| AAVS-EDA          | 2.48    |
| IPOP-CMAES        | 2.95    |
| HHSPSO-GDS        | 3.23    |
| AMaLGaM           | 3.28    |
| CPI-JADE          | 3.38    |
| EMNA <sub>g</sub> | 5.67    |

As a whole, AAVS-EDA is very competitive compared with its competitors which contain four state-of-the-art EAs. This mainly profits from the effectiveness of AAVS which can adaptively adjust the search



scope and directions of AAVS-EDA. Considering AAVS-EDA possesses a simple algorithmic framework and only three parameters which are easy to set, it can be viewed as an efficient and practical EA.

## 5. Conclusions

This paper presents a new EDA variant named AAVS-EDA for continuous optimization problems. Its main advantages lie in three aspects. First, it explicitly detects the landscape of an optimization problem in eigenspace with a simple topology-based detection method, and reveals that an optimization problem may have different landscape characteristics along different eigendirections. Second, it independently and adaptively adjusts the variable variances along different eigendirections according to the landscape detection results. As a result, both its search scope and search direction can fit well with the characteristics of the problem being solved. This shows great superiority to existing variance scaling strategies since they can only adjust the search scope. Finally, AAVS-EDA introduces an auxiliary global monitor which shrinks variable variances if the average fitness of the selected high-quality solutions does not improve. This helps AAVS-EDA to converge to a promising solution region. The performance of AAVS-EDA was fully tested on a set of 30 benchmark functions. The comparison results with several state-of-the-art EAs verify the efficiency and superiority of AAVS-EDA.

The superiority of AAVS-EDA is obtained at the cost of sampling additional solutions for detecting the landscape in eigenspace. For future research, it is of great significance to study landscape detection method by employing historical solutions. Since landscape information is beneficial to all kinds of EAs, then it is interesting to introduce the idea of AAVS into other EAs like DE and PSO. Moreover, we will extensively evaluate the performance of AAVS-EDA on some challenging optimization problems, including constrained optimization problem, combinatorial optimization problem, and some real world problems [49-52].

## Acknowledgements

This work was partially supported by the National Natural Science Foundation of China [grant numbers 61105126, 61503300]; and the Postdoctoral Science Foundation of China [grant numbers 2014M560784, 2016T90922].

## References

- [1] K. De Jong, Evolutionary computation: a unified approach, in: Proc. 14th Annu. Conf. Companion Genet. Evol. Comput., ACM, 2012: pp. 737–750.
- [2] H. Mühlenbein, G. Paass, From recombination of genes to the estimation of distributions I. Binary parameters, Springer Berlin Heidelberg, 1996.
- [3] P. Larranga, R. Etxeberria, J.A. Lozano, J.M. Pena, Optimization by learning and simulation of Bayesian and Gaussian networks, Tech. Rep. EHU-KZAA-IK-4/99, Univ. Basqu. Country, Spain. (1999).
- [4] P. Larrañaga, J.A. Lozano, Estimation of distribution algorithms: A new tool for evolutionary computation, Springer Science & Business Media, 2001.
- [5] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, *Swarm Evol. Comput.* 1 (2011) 111–128.
- [6] J. Shen, L. Wang, S. Wang, A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion, *Knowledge-Based Syst.* 74 (2015) 167–175.
- [7] A. Zhou, J. Sun, Q. Zhang, An estimation of distribution algorithm with cheap and expensive local search methods, *IEEE Trans. Evol. Comput.* 19 (2015) 807–822.
- [8] J. Ocenasek, S. Kern, N. Hansen, P. Koumoutsakos, A mixed bayesian optimization algorithm with variance adaptation, in: *Int. Conf. Parallel Probl. Solving from Nat.*, Springer, 2004: pp. 352–361.
- [9] J. Ocenasek, J. Schwarz, Estimation of distribution algorithm for mixed continuous-discrete optimization problems, in: *2nd Euro-International Symp. Comput. Intell.*, 2002: pp. 227–232.

- [10] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, 1973.
- [11] B. Yuan, M. Gallagher, On the importance of diversity maintenance in estimation of distribution algorithms, in: *Proc. 7th Annu. Conf. Genet. Evol. Comput.*, ACM, 2005: pp. 719–726.
- [12] P. Pošík, Preventing premature convergence in a simple EDA via global step size setting, *Parallel Probl. Solving from Nature–PPSN X. (2008)* 549–558.
- [13] J. Grahl, P.A.N. Bosman, F. Rothlauf, The correlation-triggered adaptive variance scaling IDEA, in: *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, ACM, 2006: pp. 397–404.
- [14] P.A.N. Bosman, J. Grahl, F. Rothlauf, SDR: A better trigger for adaptive variance scaling in normal EDAs, in: *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, ACM, 2007: pp. 492–499.
- [15] Y. Cai, X. Sun, H. Xu, P. Jia, Cross entropy and adaptive variance scaling in continuous EDA, in: *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, ACM, 2007: pp. 609–616.
- [16] M. R. Wagner, A. Auger, and M. Schoenauer, EEDA: A New Robust Estimation of Distribution Algorithm, *Rapport de Recherche (Research Report) RR-5190*, INRIA, 2004.
- [17] W. Dong, X. Yao, Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms, *Inf. Sci.* 178 (15) (2008) 3000–3023.
- [18] J. Liu, Y. Wang, H. Teng, Variance analysis and adaptive control in intelligent system based on Gaussian model, *Int. J. Model. Identif. Control.* 18 (2013) 26–33.
- [19] P.A.N. Bosman, J. Grahl, D. Thierens, Enhancing the performance of maximum–likelihood Gaussian EDAs using anticipated mean shift, in: *Int. Conf. Parallel Probl. Solving from Nat.*, Springer, 2008: pp. 133–143.
- [20] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evol. Comput.* 9 (2001) 159–195.

- [21] Z. Ren, C. He, D. Zhong, S. Huang, Y. Liang, Enhance continuous estimation of distribution algorithm by variance enlargement and reflecting sampling, in: 2016 IEEE Congr. Evol. Comput. CEC 2016, 2016. doi:10.1109/CEC.2016.7744225.
- [22] Y. Liang, Z. Ren, L. Wang, B. Pang, M.M. Hossain, Inferior solutions in Gaussian EDA: Useless or useful?, 2017 IEEE Congr. Evol. Comput. (2017) 301–307. doi:10.1109/CEC.2017.7969327.
- [23] Q. Xu, Chaos elitism estimation of distribution algorithm, in: Fifth Int. Conf. Intell. Control Inf. Process., 2014: pp. 265–269. doi:10.1109/ICICIP.2014.7010352.
- [24] Y. Chen, X. Sun, D. Gong, Y. Zhang, J. Choi, S. Klasky, Personalized Search Inspired Fast Interactive Estimation of Distribution Algorithm and Its Application, IEEE Trans. Evol. Comput. (2017).
- [25] H. Fang, A. Zhou, G. Zhang, An estimation of distribution algorithm guided by mean shift, 2016 IEEE Congr. Evol. Comput. (2016) 3268–3275. doi:10.1109/CEC.2016.7744203.
- [26] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: Evol. Comput. 2005. 2005 IEEE Congr., IEEE, 2005: pp. 1769–1776.
- [27] C. García-Martínez, P.D. Gutiérrez, D. Molina, M. Lozano, F. Herrera, Since CEC 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis's weakness, Soft Comput. (2017) 1–11.
- [28] H. Karshenas, R. Santana, C. Bielza, P. Larrañaga, Regularized continuous estimation of distribution algorithms, Appl. Soft Comput. 13 (2013) 2412–2432.
- [29] R. Santana, A. Mendiburu, J.A. Lozano, Customized Selection in Estimation of Distribution Algorithms, in: Asia-Pacific Conf. Simulated Evol. Learn., Springer, 2014: pp. 94–105.
- [30] J. Zhang, J. Zeng, Estimation of distribution algorithm based on sequential importance sampling particle filters, Acta Electron. Sin. 38 (2010) 2929–2932.

- [31] B. Qian, Z. Li, R. Hu, A copula-based hybrid estimation of distribution algorithm for m-machine reentrant permutation flow-shop scheduling problem, *Appl. Soft Comput.* 61 (2017) 921–934. doi:10.1016/j.asoc.2017.08.037.
- [32] L. PourMohammadBagher, M.M. Ebadzadeh, R. Safabakhsh, Graphical model based continuous estimation of distribution algorithm, *Appl. Soft Comput.* 58 (2017) 388–400. doi:10.1016/j.asoc.2017.04.066.
- [33] P. Yang, K. Tang, X. Lu, Improving Estimation of Distribution Algorithm on Multimodal Problems by Detecting Promising Areas, *IEEE Trans. Cybern.* 45 (2015) 1438–1449. doi:10.1109/TCYB.2014.2352411.
- [34] X. Li, M.G. Epitropakis, K. Deb, A. Engelbrecht, Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications, *IEEE Trans. Evol. Comput.* 21 (2017) 518–538. doi:10.1109/TEVC.2016.2638437.
- [35] Q. Yang, W.N. Chen, Y. Li, C.L.P. Chen, X.M. Xu, J. Zhang, Multimodal Estimation of Distribution Algorithms, *IEEE Trans. Cybern.* 47 (2017) 636–650. doi:10.1109/TCYB.2016.2523000.
- [36] C.W. Ahn, J. An, J. Yoo, Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs, *Inf. Sci. (Ny)*. 192 (2012) 109–119.
- [37] Y. Wang, B. Li, T. Weise, Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems, *Inf. Sci. (Ny)*. 180 (2010) 2405–2420. doi:10.1016/j.ins.2010.02.015.
- [38] R. Rastegar, On the Optimal Convergence Probability of Univariate Estimation of Distribution Algorithms, *Evol. Comput.* 19 (2011) 225–248. doi:10.1162/EVCO\_a\_00022.
- [39] C. Echegoyen, R. Santana, A. Mendiburu, J.A. Lozano, Comprehensive characterization of the behaviors of estimation of distribution algorithms, *Theor. Comput. Sci.* 598 (2015) 64–86. doi:10.1016/j.tcs.2015.04.015.

- [40] C.M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics), Springer-Verlag New York, Inc., 2006.
- [41] R.K. Ursem, Multinational GAs: Multimodal optimization techniques in dynamic environments, in: *Proc. 2nd Annu. Conf. Genet. Evol. Comput.*, Morgan Kaufmann Publishers Inc., 2000: pp. 19–26.
- [42] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Report*, Nanyang Technol. Univ. Singapore. (2013).
- [43] I. Loshchilov, M. Schoenauer, M. Sebag, Alternative restart strategies for CMA-ES, in: *Int. Conf. Parallel Probl. Solving from Nat.*, Springer, 2012: pp. 296–305.
- [44] I. Loshchilov, CMA-ES with restarts for solving CEC 2013 benchmark problems, in: *Evol. Comput. (CEC)*, 2013 IEEE Congr., Ieee, 2013: pp. 369–376.
- [45] Y. Wang, Z. Liu, J. Li, H. Li, G.G. Yen, Utilizing cumulative population distribution information in differential evolution, *Appl. Soft Comput.* 48 (2016) 329–346.
- [46] H. Ouyang, L. Gao, X. Kong, S. Li, D. Zou, Hybrid harmony search particle swarm optimization with global dimension selection, *Inf. Sci.* 346 (2016) 318–337.
- [47] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, *IEEE Trans. Evol. Comput.* 13 (2009) 945–958.
- [48] J. Cohen, *Statistical power analysis for the behavioral sciences* Lawrence Earlbaum Associates, Hillsdale, NJ. (1988) 20–26.
- [49] D. Bose, S. Kundu, S. Biswas, S. Das, Circular Antenna Array Design Using Novel Perturbation Based Artificial Bee Colony Algorithm, in *Swarm, Evolutionary, and Memetic Computing (SEMCCO)*, 2012 Third International Conference on, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012: pp. 459–466. doi:10.1007/978-3-642-35380-2\_54.

- [50] S. Kundu, S. Das, A. V Vasilakos, S. Biswas, A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks, *Soft Comput.* 19 (2015) 637–659. doi:10.1007/s00500-014-1286-9.
- [51] D. Bose, S. Biswas, A. V Vasilakos, S. Laha, Optimal filter design using an improved artificial bee colony algorithm, *Inf. Sci. (Ny)*. 281 (2014) 443–461. doi: 10.1016/j.ins.2014.05.033.
- [52] A. Trivedi, D. Srinivasan, S. Biswas, T. Reindl, A genetic algorithm-differential evolution based hybrid framework: Case study on unit commitment scheduling problem, *Inf. Sci. (Ny)*. 354 (2016) 275–300. doi: 10.1016/j.ins.2016.03.023.