Available online at www.sciencedirect.com

ScienceDirect

Information Sciences 178 (2008) 3000–3023

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

# Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms

Weishan Dong[a,*], Xin Yao[b]

[a] Key Laboratory for Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China

[b] The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, The University of Birmingham, Birmingham B15 2TT, UK

Received 5 June 2007; received in revised form 26 January 2008; accepted 29 January 2008

## Abstract

Multivariate Gaussian models are widely adopted in continuous estimation of distribution algorithms (EDAs), and covariance matrix plays the essential role in guiding the evolution. In this paper, we propose a new framework for multivariate Gaussian based EDAs (MGEDAs), named eigen decomposition EDA (ED-EDA). Unlike classical EDAs, ED-EDA focuses on eigen analysis of the covariance matrix, and it explicitly tunes the eigenvalues. All existing MGEDAs can be unified within our ED-EDA framework by applying three different eigenvalue tuning strategies. The effects of eigenvalue on influencing the evolution are investigated through combining maximum likelihood estimates of Gaussian model with each of the eigenvalue tuning strategies in ED-EDA. In our experiments, proper eigenvalue tunings show high efficiency in solving problems with small population sizes, which are difficult for classical MGEDA adopting maximum likelihood estimates alone. Previously developed covariance matrix repairing (CMR) methods focusing on repairing computational errors of covariance matrix can be seen as a special eigenvalue tuning strategy. By using the ED-EDA framework, the computational time of CMR methods can be reduced from cubic to linear. Two new efficient CMR methods are proposed. Through explicitly tuning eigenvalues, ED-EDA provides a new approach to develop more efficient Gaussian based EDAs.

*Keywords:* Estimation of distribution algorithm; Eigen analysis; Multivariate Gaussian distribution; Covariance matrix scaling; Eigenvalue tuning

## 1. Introduction

Estimation of distribution algorithms (EDAs) [16,18] have been studied intensively for their use in continuous optimization domains. Such approaches employ population-based heuristic searching strategies, and have been an active branch of evolutionary computation (EC). The main difference between EDAs and the

* Corresponding author.
*E-mail addresses:* weishan.dong@ia.ac.cn (W. Dong), x.yao@cs.bham.ac.uk (X. Yao).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.01.021

well-known genetic algorithms (GAs) [10] is that a new population is generated neither with crossover nor with mutation. Instead, new individuals are sampled from a probability distribution estimated from selected individuals of previous generations. The interrelations between variables are expressed explicitly in EDAs through joint probability distributions. Multivariate Gaussian models are used in many existing continuous EDAs. These EDAs are essentially Multivariate Gaussian based EDAs, and we will use the abbreviation, MGEDAs, in the remainder of this paper.

In this paper, we establish a fresh perspective on eigen analysis of covariance matrix for EDAs. A new unified framework for MGEDAs named eigen decomposition EDA (ED-EDA) is proposed. All existing MGEDAs can be unified with this framework. More importantly, the ED-EDA framework reveals that different performances of MGEDAs rely on nothing but the eigenvalues of covariance matrix. ED-EDA has a replaceable module whose function is tuning eigenvalues. Different strategies applied in this module correspond to different central ideas of existing MGEDAs. Proper eigenvalue tuning strategies show high efficiency in solving problems with small population sizes. While with small populations, classical EDAs that use maximum likelihood to estimate Gaussian without tuning usually fail. By adopting new eigenvalue tuning strategy in ED-EDA, more efficient EDAs can be developed. Furthermore, two new methods for repairing negative eigenvalues derived from Covariance Matrix Repairing (CMR) [8] are proposed by ED-EDA. The computation time of the repairing can be sufficiently reduced from $O(k^3)$ to $O(k)$, where $k$ denotes the problem size.

The remainder of this paper is organized as follows. In Section 2, we review MGEDAs and covariance matrix repairing (CMR). Then we present the ED-EDA framework in Section 3. In Section 4, we unify all existing MGEDAs from the perspective of ED-EDA, and point out that their fundamental differences lie in the eigenvalues. In Section 5, we analyze the effects of CMR within ED-EDA and propose two new repairing methods, ECMR and ECMR0. Experimental studies are given in Section 6 to show the reliability and necessity of ECMR0, and the efficiency of ED-EDA with eigenvalue tuning strategies. Our final conclusions are drawn in Section 7.

## 2. Multivariate Gaussian based EDAs

EDAs were proposed originally for combinatory optimizations. Research on EDAs has been extended from discrete domains to continuous optimization and a large amount of progress has been made. A survey of EDAs and their applications to continuous optimization can be found in [16,20,4,9]. In this paper, we focus on Multivariate Gaussian based EDAs (MGEDAs) and related methods for tuning the Gaussian models.

### *2.1. Existing MGEDAs*

- EMNA: EMNA (estimation of multivariate normal algorithm) [16] employs a multivariate Gaussian density. EMNA has two versions, $\text{EMNA}_{\text{global}}$ and $\text{EMNA}_{a}$. In this paper, we use $\text{EMNA}_{\text{global}}$ as an example as it is easier to analyze. In $\text{EMNA}_{\text{global}}$, the sample mean and covariance matrix are computed by maximum likelihood estimates.
- RECEDA: RECEDA (real-coded estimation of distribution algorithm) [19] is essentially identical to $\text{EMNA}_{\text{global}}$, except that the sampling method from multivariate Gaussian with Cholesky decomposition is explicitly outlined. Since EMNA was proposed earlier, and the sampling method RECEDA used is a common technique in random variate generation, RECEDA can be seen as a specific implementation of $\text{EMNA}_{\text{global}}$.
- IDEA with normal density: IDEA is a framework in which different pdfs (probabilistic density functions) can be used [3–5]. If a normal pdf is used, sample mean and covariance matrix are estimated in the same way as in $\text{EMNA}_{\text{global}}$. In [1,12], adaptive variance scaling (AVS) and correlation-triggered AVS (CT-AVS) were proposed to be used along with the normal pdf in IDEA. AVS/CT-AVS helps to increase the area of Gaussian's exploration by scaling the maximum likelihood estimated covariance matrix.

- EEDA: EEDA (eigenspace EDA) [22] also uses multivariate Gaussian model, but the covariance matrix's minimum eigenvalue is to be reset to the value of the maximum eigenvalue after performing the maximum likelihood estimates. EEDA can be seen as an extension to EMNA, but its performance has not been investigated comprehensively in the literature.

All of the approaches above share a common characteristic that essentially they use a multivariate Gaussian model. Generally speaking, after estimating the covariance matrix by maximum likelihood, there are three different strategies to tune it:

1. Leave the covariance matrix alone and directly use the maximum likelihood estimates for sampling, as in EMNA, RECEDA, and IDEA with a normal pdf.
2. Uniformly scale the covariance matrix by a positive coefficient to expand or shrink the variances of current Gaussian, as in IDEA with AVS/CT-AVS. The coefficient AVS/CT-AVS used is self-adaptive.
3. Non-uniformly scale the covariance matrix along selected directions of its eigenvectors, as in EEDA.

Obviously, the first strategy is the most direct one. Common multivariate Gaussian can be seen as a rotation of the simple univariate marginal Gaussian, whose covariances between each pair of different variables are all zeros. Although univariate model has some limitations, it has elegant property of convergence: After sufficient generations, the mean of Gaussian becomes stable and the variance converges towards zero [11,13]. Although not having been rigorously proved, a rotated multivariate Gaussian does have similar convergence property in practice. The reason for stressing the convergence of maximum likelihood estimated Gaussian here is because unconverged variance may lead to strange behaviors in the algorithm, for instance, traversing around the search space but never settling down anywhere. Thus convergence is a merit of the first strategy. However, its drawback is also obvious. When the sample size (population size in EDA) is too small compared with the problem size to establish an adequate initial distribution, the exploring effectiveness will be fast deteriorating and pre-mature convergence will arise [1].

The second strategy can overcome the drawback of the first. Exploring effectiveness is achieved by uniformly scaling the covariance matrix during evolution. When the scaling is performed in a multiplicative manner, it can symmetrically shrink and expand. Although strict theoretical analysis has not been developed, the convergence property of this uniform scaling strategy still approximately holds in practice. However, a counter example of non-convergence will be illustrated in our experiments.

The last strategy, non-uniform scaling of the covariance matrix, can be regarded as resetting eigenvalues along selected eigenvector directions. So far, strict theoretical analysis of convergence for such an operation has not been developed either. Compared with the uniform scaling, its advantage is that the scaling of variances can be performed only in the necessary directions. Unnecessary exploration can be avoided. Of course, how to select the directions is a new problem. EEDA adopts the simplest way to scale only in the direction of the eigenvector corresponding to the minimum eigenvalue of covariance matrix.

### 2.2. Covariance matrix repairing

Covariance matrix repairing (CMR) [8] is a technique for repairing an ill-posed covariance matrix which is not positive semi-definite. A covariance matrix is always positive semi-definite by its definition. But because of numerical errors caused by the finite precision of computation capability of electronic computers, sometimes the covariance matrix we get has negative eigenvalues, which indicates the matrix is not positive semi-definite. In practice, we also have the experience that the smaller the sample size is relatively to the problem size (dimensionality, or number of variables), the more probable such an ill-posed covariance matrix emerges. In other words, when the sample data are sparse in the search space, the effect of computational error is more likely to be amplified. In addition, the randomness of stochastic evolutionary procedure leads to the fact that the computational error cannot be totally avoided just by luck. In [8], some functions that have higher risks to produce ill-posed covariance matrices have been observed.

When facing an ill-posed covariance matrix, MGEDAs' primary obstacle is that sampling new individuals becomes impossible. To sample new individuals from a given $k \times k$ covariance matrix $\Sigma$, the key step is to first decompose $\Sigma$ into $\Sigma = HH^{\mathrm{T}}$ [7]. This requires the positive definiteness of $\Sigma$. More precisely, to sample a vector $Y$ from $\mathcal{N}(\mu, \Sigma)$, we should:

1. Decompose $\Sigma$ to find $H$, s.t. $HH^{\mathrm{T}} = \Sigma$.
2. Generate $k$ independent normal variates $x_1, \ldots, x_k$, where $x_i \sim \mathcal{N}(0,1), i = 1, \ldots, k$.
3. Return $Y = \mu + HX$.

The first step of decomposing $\Sigma$ cannot be accomplished when $\Sigma$ has negative eigenvalues. An example of $\Sigma$ with a negative eigenvalue can be found in [8]. In this case, it is impossible to sample new individuals. This is a deadlock for MGEDAs, because an ill-posed covariance matrix is caused intrinsically by the finite precision of computers, it is also a universal problem of all Gaussian based EDAs using a full covariance matrix. Similar EDAs such as EGNA [15,16] and IDEA with a normal pdf were also analyzed in [8].

To solve this problem, CMR adds a scaled identity matrix to the ill-posed $\Sigma$ to repair the error. Such kind of repairing is also a common technique in the experiments of statistical learning field [25]. But in machine learning, pattern recognition, signal processing, etc., people usually add positive values to the diagonal of covariance matrix that may serve as regularization and lead to better estimates [21]. The more commonly used routine in statistical learning is to shrink the covariance matrix towards an identity matrix $I$ [17] for better generalization performance of $\Sigma$. An optimal value exists to be added so that the estimated $\Sigma$ does not over-fit the data, or deviate from the estimate too much. Some people arbitrarily add a 'small' positive value, e.g., 0.001 to the diagonal elements of $\Sigma$ without any serious consideration as long as it could make $\Sigma$ positive semi-definite. If such a value is not enough, 0.01 could be tried again until $\Sigma$ is positive semi-definite. No systematic methods have been proposed in the literature.

In EDA, especially in its converging stage, where $\Sigma$ almost converges to a zero matrix, a seemingly 'small' value like 0.001 can be quite 'large' to an estimated $\Sigma$. Adding a relatively big value can inflate the nearly converged variance of a variable, and because of the iterative mechanism of EDA, it will also disperse the sample points in that dimension in all the following generations. In other words, such repairing to $\Sigma$ in the context of EDA needs to be very precise. That which is used in machine learning cannot be applied effectively here. CMR tries to repair $\Sigma$ to a positive semi-definite one and at the same time makes the least modification to it to hold all the learnt properties of the distribution. [8] presented some experimental comparisons between EDAs with CMR and EDAs with improper repairing, in which the lost precision brought by improperly added value strongly deteriorated the convergence rates of algorithms.

In practice, for a given problem, with a larger population size, the probability of covariance matrix becoming ill-posed is lower than those cases with smaller population sizes. But it is still not completely avoidable. It was shown in [8] that EMNA with a relatively large population size = 1000 still needs CMR on some of the 30d functions.

[8] also showed that the EMNA with only a few CMR activations significantly outperforms the original EMNA and other EMNA based algorithms that are without (or, with improper) error repairing. In other words, we have to clarify whether a bad result is brought by incompetency of the model or just by negative impacts of computational errors. By using CMR, we can remove the effects of error and reveal the real ability of Gaussian model.

Sometimes, activations of CMR come along with early convergence of algorithms as we will see in this paper. The reason for this is that both of them have connections to the population size. Small population sizes easily produce ill-posed covariance matrices [8], and they also easily cause early convergence of EDAs [1]. On the other hand, because the computational error is uncontrollable, when a covariance matrix needs repairing, it does not mean that the algorithm has converged prematurally.

As we have analyzed in [8], CMR is necessary for MGEDAs. It can be restated as one of the non-uniform covariance matrix scaling strategies. Further explanations are to be shown later.

## 3. A unified framework for MGEDAs: eigen decomposition EDA

### 3.1. Eigen analysis on MGEDAs

The decomposition of a real square matrix $\Sigma$ into eigenvalues and eigenvectors is known as eigen decomposition. Assuming matrix $\Sigma$ has nondegenerate eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ and corresponding linearly independent eigenvectors $v_1, v_2, \ldots, v_k$, we have

$$\Sigma = PDP^{\mathrm{T}}, \tag{1}$$

where matrices $P$ and $D$ are composed of eigenvectors

$$P = [v_1, v_2, \ldots, v_k] \tag{2}$$

and eigenvalues

$$D = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & \lambda_k \end{bmatrix}. \tag{3}$$

We also have $P' = P^{\mathrm{T}}, PP' = PP^{\mathrm{T}} = I$.

Furthermore, if $\Sigma$ is a sample covariance matrix, it is not only a square matrix but also symmetric. There are as many eigenvalues as rows (or columns) in $\Sigma$. Conceptually, an eigenvalue can be considered to measure the strength (relative range of distributed data) of an axis whose orientation in space is determined by the corresponding eigenvector. Thus the shape of current Gaussian distribution can be completely described by the eigen decomposition of $\Sigma$. Analysis of eigenvalues and eigenvectors is widely used in the field of multivariate analysis [6]. Eigen analysis also plays a crucial role in principal components analysis (PCA) [14], which is used in the field of machine learning for the purpose of feature extraction and reducing the dimensionality of data without significant loss of information.

Returning to MGEDAs, the purpose of decomposing $\Sigma$ is to sample new individuals. By using some decomposition methods, e.g. Cholesky decomposition adopted by RECEDA, covariance matrix $\Sigma$ can be decomposed into matrices $H$ and $H^{\mathrm{T}}$, s.t. $HH^{\mathrm{T}} = \Sigma$, and new data can be generated using $H$ by following the three steps described in Section 2.2. If we use eigen decomposition, we can easily transform $P$ and $D$ to $H$:

$$\Sigma = PDP^{\mathrm{T}} = P \cdot D^{1/2}D^{1/2} \cdot P^{\mathrm{T}} = PD^{1/2} \cdot D^{1/2}P^{\mathrm{T}} = H \cdot H^{\mathrm{T}}, \tag{4}$$

where $H = PD^{1/2}$. Thus it is easy to sample new individuals by eigen decomposition. Besides, an obvious advantage of eigen decomposition is that we can explicitly tune all the eigenvalues of the covariance matrix. We can move our analysis from the original data space to the eigen space, and understand MGEDAs more profoundly. Eigen decomposition has other advantages too in terms of saving computational time of MGEDAs, which will be introduced later.

### 3.2. Eigen decomposition EDA (ED-EDA)

As discussed before, existing MGEDAs only differ in their manner of tuning/scaling the covariance matrix. We will see that all the three strategies essentially tune/scale nothing but the eigenvalues of the covariance matrix. We propose a unified framework for MGEDAs, named Eigen Decomposition EDA (ED-EDA), to make it clear. The structure of ED-EDA is shown in Fig. 1. Pay attention to the bolded Step 4 in the main loop. By using different strategies in this replaceable module, we are able to reproduce all existing MGEDAs. Even more efficient EDAs can be created using this unified framework.

## 4. Eigenvalues make differences

From ED-EDA's point of view, the three strategies summarized in Section 2.1 are three special cases. The first strategy that is used in EMNA, RECEDA and IDEA with normal pdf just leaves $D$ unchanged. So the

```
ED-EDA

Initialize a population by generating $R$ individuals randomly.
Repeat until a stopping criterion is met.
    1. Select $N \le R$ individuals from the population.
    2. $f(x) = \mathcal{N}(x; \mu, \Sigma) \leftarrow$ Estimate mean $\mu$ and covariance matrix $\Sigma$ of
       the multivariate Gaussian density function by the maximum likelihood
       method from the selected individuals.
    3. Using eigen decomposition $\Sigma = PDP^T$ to obtain $P$ and $D$ as (1).
    4. Tune $D$.
    5. Sample $R$ individuals (the new population) from $f(x)$ by $P$ and $D$: Using
       (4) to obtain $H$ and follow the three steps described in Section 2.2.
```

Fig. 1. Eigen decomposition EDA (ED-EDA).

main idea of these EDAs can be immediately represented within ED-EDA as Fig. 1 by simply removing Step 4 (or designating Step 4 as a null operation). While the second and third strategies apply different techniques to tune $D$ after maximum likelihood estimates.

### *4.1. Uniform eigenvalue scaling*

For the second strategy, uniform covariance matrix scaling, if the scaling coefficient is $c(c > 0)$, and covariance matrix is $\Sigma$, then the scaled covariance matrix should be $c \cdot \Sigma$. Using eigen decomposition, we have

$$\Sigma_{\text{new}} = c \cdot \Sigma = c \cdot PDP^{\mathrm{T}} = P \cdot cD \cdot P^{\mathrm{T}} = P \cdot \begin{bmatrix} c\lambda_1 & 0 & \dots & 0 \\ 0 & c\lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & c\lambda_k \end{bmatrix} \cdot P^{\mathrm{T}} = PD_{\text{new}}P^{\mathrm{T}}. \tag{5}$$

It means that uniformly scaling $\Sigma$ is identical to uniformly scaling $D$. Note that all eigenvalues are scaled by the same ratio. So we have

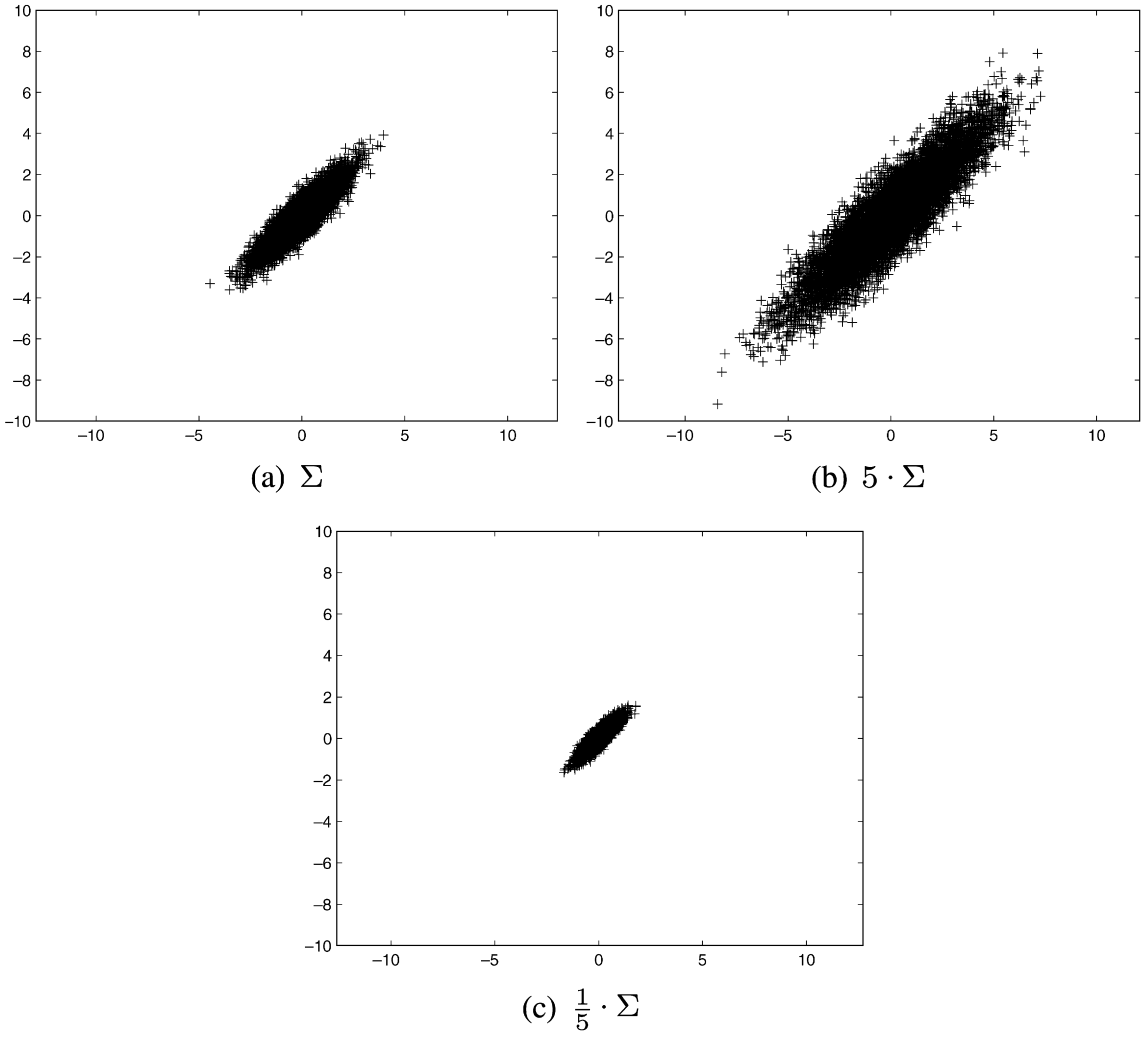
$$\frac{\lambda_{\text{new},i}}{\lambda_{\text{new},j}} = \frac{c\lambda_i}{c\lambda_j} = \frac{\lambda_i}{\lambda_j} \quad \forall i,j = 1,\dots,k, \ \lambda_j \neq 0, \tag{6}$$

which means that the relative strength ratios between different eigenvector axes remain as before. A 2-dimensional demonstration is depicted in Fig. 2. Eight thousand data points are sampled from three Gaussian models with zero means and covariance matrices $\Sigma$, $5\Sigma$ and $\frac{1}{5}\Sigma$, respectively, where

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}. \tag{7}$$

For higher dimensions, the 2d ellipse is generalized to a hyper ellipsoid, which is analogous to the 2d situation. Apparently, uniform scaling expands or shrinks covariance matrices without skewing or twisting them.

Techniques like AVS adopt such a strategy. By using ED-EDA we can represent AVS by re-writing Step 4 in the main loop of ED-EDA as Fig. 3. For CT-AVS with the triggering method, we can also represent it within ED-EDA, which is shown in Fig. 4.



(a) $\Sigma$

(b) $5 \cdot \Sigma$

(c) $\frac{1}{5} \cdot \Sigma$

Fig. 2. Comparison of 2d Gaussian distributions of uniformly scaled covariance matrix.

Step 4: Tuning $D$

(1) Compute the coefcient $c^{AVS}$ as in [12].

(2) $D \leftarrow c^{AVS} \cdot D$.

Fig. 3. Tuning *D* using AVS.

### *4.2. Non-uniform eigenvalue scaling*

For non-uniform covariance matrix scaling, i.e. the third strategy, we have more flexibility to attach different importance to each of the eigenvalues. EEDA and CMR can be regarded as two specific approaches of this strategy.

Step 4: Tuning $D$

(1) Compute the coefcient $c^{AVS}$ and the ranked correlation coefficient $r$ as in [12].

(2) If $r > \theta^{corr}$, the correlation trigger threshold, then $D \leftarrow c^{AVS} \cdot D$.

Fig. 4. Tuning $D$ using CT-AVS.

#### 4.2.1. EEDA: minimum eigenvalue resetting

Among all directions of eigenvectors, the one corresponding to the smallest eigenvalue indicates the direction that has the least variance. When the mean of current Gaussian is away from the global optimum, such a direction is an approximation of the negative gradient of the fitness function [22]. In other words, this direction should be paid more attention to than the others. Searching along this direction should be more efficient in terms of approaching the optimum. Suppose the current covariance matrix is $\Sigma$; the main idea of EEDA is

$$\Sigma_{\text{new}} = \Sigma + (\lambda_{\max} - \lambda_{\min}) v_{\min} v_{\min}^{\mathrm{T}}, \tag{8}$$

where $\lambda_{\min} = min\{\lambda_i\}, \lambda_{\max} = max\{\lambda_i\}, i = 1, 2, \ldots, k$, and $v_{\min}$ is the eigenvector corresponding to $\lambda_{\min}$. Using eigen analysis, we have the following equations derived from (8):

$$\begin{aligned}
\Sigma_{\text{new}} &= \Sigma + (\lambda_{\max} - \lambda_{\min}) v_{\min} v_{\min}^{\mathrm{T}} = PDP^{\mathrm{T}} + (\lambda_{\max} - \lambda_{\min}) v_{\min} v_{\min}^{\mathrm{T}} \\
&= \sum_{i=1}^{k} \lambda_i v_i v_i^{\mathrm{T}} - \lambda_{\min} v_{\min} v_{\min}^{\mathrm{T}} + \lambda_{\max} v_{\min} v_{\min}^{\mathrm{T}} = P \begin{bmatrix} \lambda_1 & 0 & & \cdots & & 0 \\ 0 & \ddots & & & & \\ \vdots & & & \lambda_{\min} - \lambda_{\min} + \lambda_{\max} & & \vdots \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & \lambda_k \end{bmatrix} P^{\mathrm{T}} \\
&= P \begin{bmatrix} \lambda_1 & 0 & \cdots & & 0 \\ 0 & \ddots & & & \\ \vdots & & \lambda_{\max} & & \vdots \\ & & & \ddots & 0 \\ 0 & & \cdots & 0 & \lambda_k \end{bmatrix} P^{\mathrm{T}} = PD_{\text{new}}P^{\mathrm{T}}.
\end{aligned} \tag{9}$$

It means that in EEDA, the minimum eigenvalue is replaced by the maximum eigenvalue to construct $D_{\text{new}}$. Unlike uniform scaling, only one selective eigenvalue is enlarged while the others remain unchanged. Thus the shape of distribution changes mostly in the direction of $v_{\min}$. A specific 2d demonstration of data sampled from two Gaussian models with zero means and covariance matrices $\Sigma$ in (7) and scaled $\Sigma_{\text{new}}$ are shown in Fig. 5.

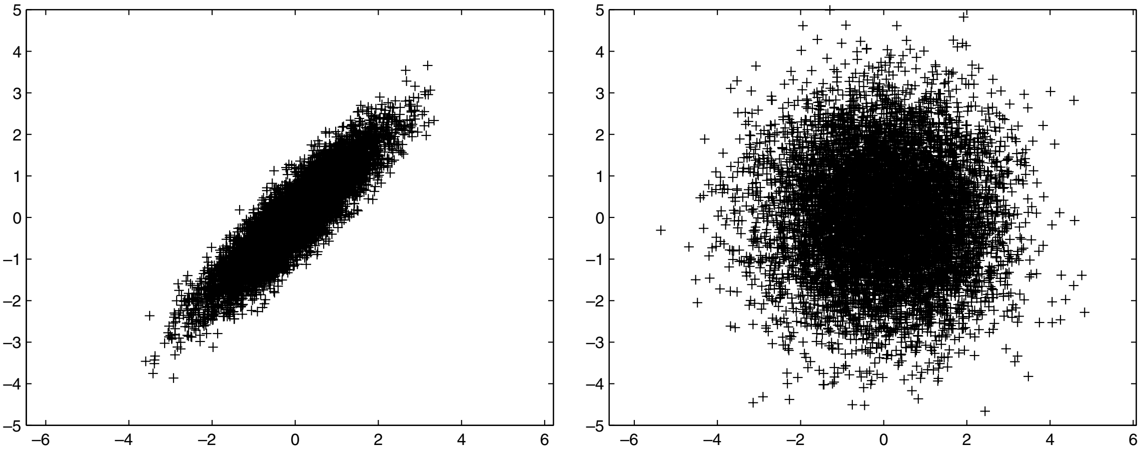
EEDA can also be represented in ED-EDA by substituting the procedure shown in Fig. 6 for Step 4 in Fig. 1.

#### 4.2.2. CMR: eigenvalue shifting

CMR adds a scaled identity matrix to an ill-posed $\Sigma$ as follows:

$$\Sigma_{\text{new}} = \Sigma + |\lambda_{\min}| \cdot K \cdot I, \tag{10}$$

where $\lambda_{\min} < 0$. $K$ is a self-adaptive coefficient to compensate rounding error of calculating eigenvalues next time. It can be generalized to



(a) $\Sigma$

(b) $\Sigma_{new}$

Fig. 5. Comparison of 2d Gaussian distributions before and after the covariance matrix is scaled by EEDA.

Step 4: Tuning $D$

(1) Find $\lambda_{max}$ and the position of $\lambda_{min}$ (marked as $D(j,j)$) in $D$.

(2) $D(j,j) \leftarrow \lambda_{max}$.

Fig. 6. Tuning $D$ using the idea of EEDA.

$$\Sigma_{\text{new}} = \Sigma + \alpha I, \quad \alpha \geqslant 0. \tag{11}$$

Although eigen decomposition can be carried out regardless whether $\Sigma$ is ill-posed or not, when transforming $P$ and $D$ to $H$ by (4), $D^{1/2}$ requires all eigenvalues to be non-negative. So CMR will still be necessary for ED-EDA. (11) can be re-written in the manner of eigen analysis as

$$\Sigma_{\text{new}} = \Sigma + \alpha I = PDP^{\mathrm{T}} + \alpha \cdot PP^{\mathrm{T}} = PDP^{\mathrm{T}} + P \cdot \alpha I \cdot P^{\mathrm{T}} = P \begin{bmatrix} \lambda_1 + \alpha & 0 & \dots & 0 \\ 0 & \lambda_2 + \alpha & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_k + \alpha \end{bmatrix} P^{\mathrm{T}} = PD_{\text{new}}P^{\mathrm{T}}. \tag{12}$$

Thus CMR can be regarded as shifting all eigenvalues with identical magnitude $\alpha$ to make all eigenvalues non-negative. Apparently, such shifting is non-uniform eigenvalue scaling, and we have two observations below:

1. When $\alpha$ is sufficiently small, the shape of the current distribution only has subtle changes which can be ignored:

$$\frac{\lambda_{\text{new},i}}{\lambda_{\text{new},j}} = \frac{\lambda_i + \alpha}{\lambda_j + \alpha} \approx \frac{\lambda_i}{\lambda_j} \quad \forall i,j = 1,\dots,k, \ \lambda_j + \alpha \neq 0, \ \lambda_j \neq 0. \tag{13}$$

In CMR, indeed $\alpha$ is small, usually less than 1e−10 in experienced observations. This is consistent with the aim of CMR of making as least modifications to $\Sigma$ as possible, as long as all eigenvalues are non-negative.

2. When $\alpha$ is sufficiently large, the ratio approximates 1:

```
Step 4: Tuning $D$
(1) Obtain $\lambda_{min}$ from $D$.
(2) If $\lambda_{min} < 0$, then $D \leftarrow D + |\lambda_{min}| \cdot K \cdot I$.
```

Fig. 7. Tuning $D$ using CMR.

$$\frac{\lambda_{\text{new},i}}{\lambda_{\text{new},j}} = \frac{\lambda_i + \alpha}{\lambda_j + \alpha} \approx \frac{\alpha}{\alpha} \approx 1 \quad \forall i, j = 1, \ldots, k, \;\; \lambda_j + \alpha \neq 0. \tag{14}$$

which means $D_{\text{new}}$ is not only enlarged but also crushed towards an amplified identity matrix $\alpha I$.

The second observation has not been made use of in MGEDAs so far. Intuitively, this tuning strategy covers both the advantages of AVS and EEDA: The exploring area is encouraged by a sufficiently large $\alpha$ as in AVS, and simultaneously the directions of eigenvectors corresponding to smaller eigenvalues are paid more attention to than those corresponding to larger eigenvalues, i.e., smaller eigenvalues are scaled up with higher ratio as in EEDA. An interesting research issue here is the convergence of such an ED-EDA.

CMR can be represented within the ED-EDA framework by re-writing the eigenvalue tuning module, which is shown in Fig. 7. Actually, coefficient $K$ (not the problem size $k$) needn't be updated and can be omitted in ED-EDA. We will analyze it in the next section.

A good property of tuning $D$ in ED-EDA is that because $D$ is diagonal, no matter which strategy is used, the computational cost is just $O(k)$ ($k$ is the problem size, not the coefficient $K$ in CMR). Because decomposing $\Sigma$ for sampling always costs $O(k^3)$ which is inevitable, if eigen decomposition is used instead of other decomposition methods, e.g., Cholesky decomposition, original CMR's primary time cost of $O(k^3)$ in calculating eigenvalues can be completely saved. Thus the time cost of CMR can be reduced from $O(k^3)$ to $O(k)$ in ED-EDA.

For original AVS/CT-AVS and EEDA, scaling $\Sigma$ costs $O(k^2)$, and then model factorization[1] or decomposition costs $O(k^3)$, so the primary computational complexity is $O(k^3)$. In ED-EDA, eigen decomposition costs $O(k^3)$, then scaling $D$ costs $O(k)$, and lastly transforming $P$ and $D$ to $H$ costs $O(k^2)$. The primary computational complexity remains the same as $O(k^3)$.

Generally speaking, by using only different strategies in the eigenvalue tuning step in ED-EDA, we can unify all MGEDAs with a unified framework. In the following experimental studies we will study the capability of each eigenvalue tuning strategy. Before that, we propose two new versions of CMR within the ED-EDA framework.

## 5. ECMR and ECMR0: from CMR to eigenvalue repairing

The self-adaptive coefficient $K$ in CMR is introduced because adding $|\lambda_{\min}|$ sometimes will not change the minimum eigenvalue of $\Sigma$ at the required precision level. But this only occurs when calculating eigenvalues again with the repaired $\Sigma$. If we use ED-EDA, re-calculating eigenvalues iteratively can be avoided, thus $K$ can be discarded as well. A new efficient version of CMR can be implemented by ED-EDA, which is called Efficient CMR (ECMR), whose structure is shown in Fig. 8.

It can be easily proved that by adding $|\lambda_{\min}| \cdot I$ to $D$, all eigenvalues in $D_{\text{new}}$ can be guaranteed to be nonnegative. We do not need to re-calculate all the eigenvalues any more and will not be perturbed by the rounding errors. ECMR avoids the repeatedly testing steps of original CMR. ECMR repairs ill-posed covariance matrices by directly repairing negative eigenvalues.

[1] In IDEA with AVS/CT-AVS, a factorized density is calculated for current Gaussian to sample new individuals. Sampling does not require decomposing covariance matrix $\Sigma$, but the computation of $\Sigma^{-1}$ is needed [1], which also costs $O(k^3)$.

```
Step 4: Tuning $D$

(1)  Obtain $\lambda_{min}$ from $D$.

(2)  If $\lambda_{min} < 0$, then $D \leftarrow D + |\lambda_{min}| \cdot I$.
```

Fig. 8. Efficient CMR (ECMR).

```
Step 4: Tuning $D$

Repeat  for $i = 1, \ldots, k$

    (1)  If $\lambda_i < 0$, then $\lambda_i \leftarrow 0$.
```

Fig. 9. Efficient CMR-zero (ECMR0).

When $\Sigma$ is ill-posed, its negative eigenvalue always has pretty small absolute value in practice [8]. ECMR and CMR can be seen as shifting all eigenvalues by an identical small magnitude. So what about setting all negative eigenvalues to zero? If all negative eigenvalues have small absolute values, only resetting these negative ones can be seen as different magnitude eigenvalue shifting. And such minor changes to $D$ ought not to influence the shape of the distribution much, which is consistent with the purpose of CMR. We propose such an eigenvalue repairing method named Efficient CMR-zero (ECMR0) in Fig. 9.

ECMR and ECMR0 both cost O($k$) only. Similar to CMR analyzed in [8], ECMR and ECMR0 are also able to be extended to Gaussian mixture model based EDAs. The time cost remains linear to the number of Gaussian components using full covariance matrices. Actually, ED-EDA can be generalized to Gaussian mixture based EDAs, but in this paper we only focus on single probabilistic models.

In the next section, we will compare CMR, ECMR and ECMR0 for different problems that easily produce ill-posed covariance matrices (negative eigenvalues) to see whether their performances have any differences. Because such repairing is necessary for MGEDAs, we will combine the best of the three with other eigenvalue tuning strategies such as AVS and EEDA to further analyze their performances.

## 6. Experimental studies

### 6.1. *Which CMR is better?*

We select four functions from [8] that can easily produce ill-posed covariance matrices to compare the effects of CMR, ECMR and ECMR0. These functions include: $f_8, f_{13}$, Rosenbrock and SumCan. Details of these functions can be found in the Appendix. SumCan is a maximization problem while the others are minimization problems. We use 10d SumCan and 30d for the other three in our experiments. Original CMR is combined with $\text{EMNA}_{\text{global}}$, while ECMR and ECMR0 are applied in the ED-EDA framework. As previously discovered, ED-EDA without the eigenvalue tuning step is essentially identical to $\text{EMNA}_{\text{global}}$. So by comparing $\text{EMNA}_{\text{global}}$ + CMR, ED-EDA + ECMR and ED-EDA + ECMR0 we should draw a fair conclusion of comparing CMR, ECMR and ECMR0.

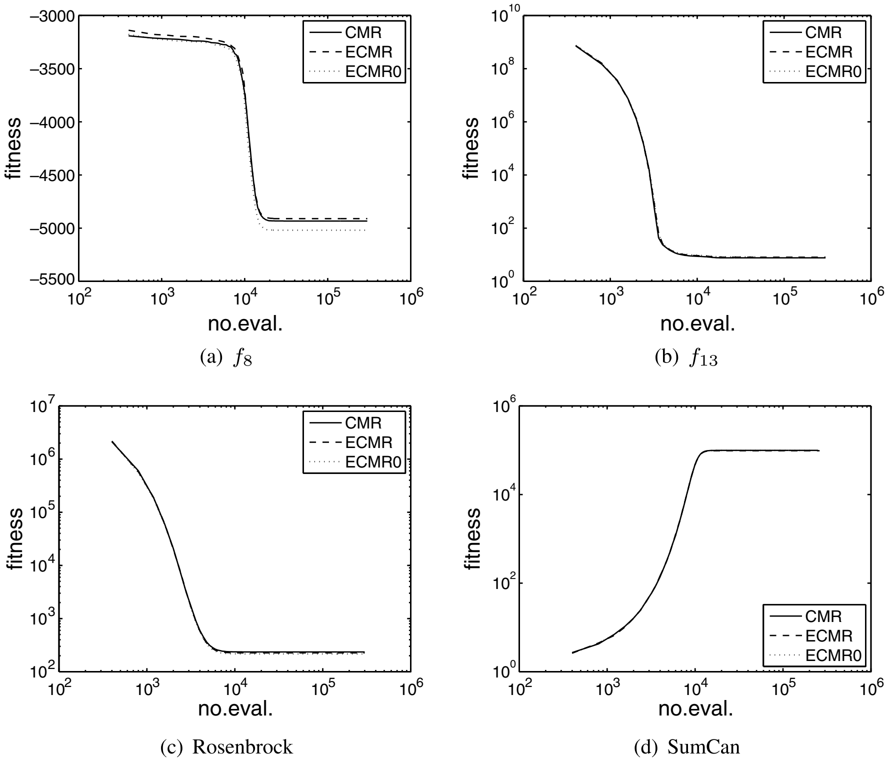
For all three algorithms and all the test functions, the same parameters are used: The population size is 400; truncation selection with selected size 60 is used; elitist approach is used as well. The stop criterion is met when the best fitness is within 1e−6 error of the global optimum, or the number of evaluations reaches 300,000, or the algorithm converges. The convergence is reached when the average fitness of the population between two consecutive generations is smaller than 1e−10. If covariance matrix $\Sigma$ or eigenvalue matrix $D$ is modified by CMR, ECMR or ECMR0, we count it for one activation of repairing.

Table 1 summarizes the average performance of 100 independent runs of each algorithm. The best fitness achieved, the least evaluations and the least activations of repairing are emphasized in bold respectively for each test. All these together imply better results, faster convergence rate and less additional computation. The average best fitness curves are shown in Fig. 10.

Table 1
Comparison among CMR, ECMR and ECMR0

| Function | Algorithm | Best fitness | No. Eval. | No. activations[a] |
|---|---|---|---|---|
| $f_8$ | CMR | −4.933186e+03 ± 6.76e+02 | 58801.6 ± 61689.8 | **49.10 ± 141.4** |
| | ECMR | −4.910472e+03 ± 7.00e+02 | 58869.5 ± 61570.0 | 56.37 ± 153.7 |
| | ECMR0 | **−5.019996e+03 ± 6.83e+02** | **56124.3 ± 56615.0** | 55.08 ± 148.3 |
| $f_{13}$ | CMR | **7.584966 ± 3.42** | 42538.4 ± 37318.9 | 16.67 ± 84.2 |
| | ECMR | 8.038493 ± 4.49 | 45331.4 ± 45256.6 | 24.79 ± 108.2 |
| | ECMR0 | 8.311668 ± 3.88 | **36980.3 ± 4542.5** | **6.67 ± 9.5** |
| Rosenbrock | CMR | 2.354741e+02 ± 1.21e+02 | **34147.4 ± 27167.5** | **7.83 ± 59.1** |
| | ECMR | 2.262437e+02 ± 9.98e+01 | 39142.9 ± 46276.4 | 21.22 ± 110.0 |
| | ECMR0 | **2.162407e+02 ± 8.55e+01** | 44936.4 ± 58989.4 | 37.41 ± 151.1 |
| SumCan | CMR | **9.888832e+04 ± 7.90e+03** | 48814.7 ± 41112.2 | **16.35 ± 51.2** |
| | ECMR | 9.731987e+04 ± 1.43e+04 | 52026.6 ± 46528.8 | 27.07 ± 81.5 |
| | ECMR0 | 9.850797e+04 ± 9.67e+03 | **48623.1 ± 40950.3** | 27.80 ± 87.2 |

[a] Note that all the numbers of activations are not zero, which means MGEDAs without covariance matrix repairing or eigenvalue repairing will definitely break down.



(a) $f_8$

(b) $f_{13}$

(c) Rosenbrock

(d) SumCan

Fig. 10. Average best fitness curves of CMR, ECMR and ECMR0 on $f_8$, $f_{13}$, Rosenbrock and SumCan.

From Table 1 we can see the restricted conditions we set (including the small population size and selected size, and the specific functions we chose) have led to a frequent occurrence of ill-posed covariance matrices/negative eigenvalues. The necessity of covariance matrix/eigenvalue repairing is clearly shown. But none of the three repairing methods show a consistent advantage over the others. Through examining the average best fitness curves, we can see no matter how frequent they are activated respectively, all the three repairing methods do not differ much in terms of influencing the evolution. Therefore, we can say the three methods have approximately identical performance. Considering the execution efficiency when activated, the simplest ECMR0 requires the least operations of all. Thus we will use ECMR0 within the ED-EDA framework in the following experiments.

### 6.2. Comparisons of different eigenvalue tuning strategies

Although an ill-posed covariance matrix/negative eigenvalue is not always encountered in practice, to avoid the effects of computational error, a repairing mechanism must be used in MGEDAs. In order to compare different eigenvalue tuning strategies, we have to keep an eye on repairment to show the true searching ability of a specific eigenvalue tuning strategy. We have combined ECMR0 with maximum likelihood estimated multivariate Gaussian without further tuning, multivariate Gaussian with AVS, and multivariate Gaussian with EEDA, respectively. That is, in Step 4 of the ED-EDA's main loop, we first use ECMR0 to repair possibly 'bad' $D$, and then tune $D$ with different strategies.

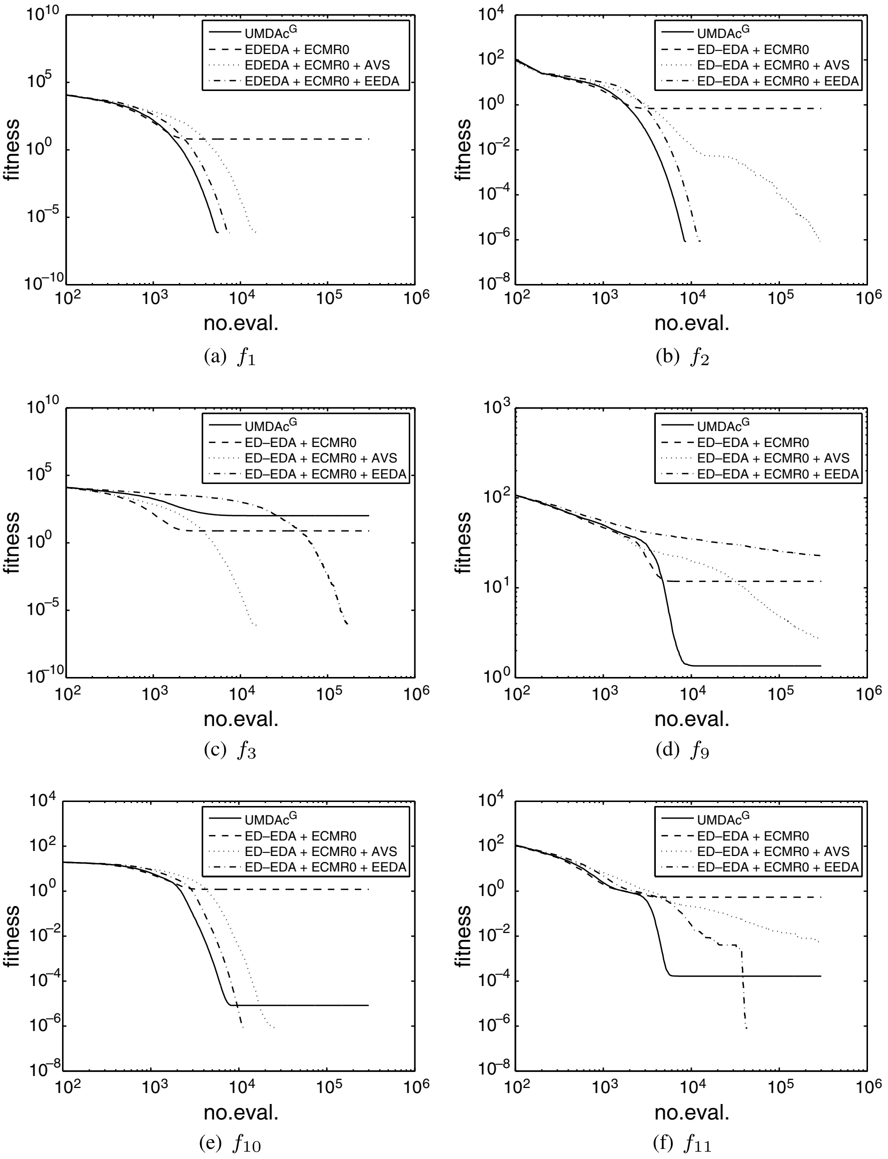
Thus the algorithms to be compared include: ED-EDA + ECMR0, ED-EDA + ECMR0 + AVS, and ED-EDA + ECMR0 + EEDA. Each of them represents an eigenvalue tuning strategy described in Section 2.1. Because the best setting of correlation trigger threshold $\theta^{\text{corr}}$ in CT-AVS is problem-dependent, we use AVS rather than CT-AVS for the eigenvalue uniform scaling strategy in our experiments. Generally speaking, ED-EDA + ECMR0 can be seen as a representative of $\text{EMNA}_{\text{global}}$, RECEDA and IDEA with normal pdf.

Table 2
Results of small population-low dimension

| Function | Algorithm | Best fitness | No. Eval. | No. ECMR0 |
|---|---|---|---|---|
| $f_1$ | $\text{UMDA}_c^G$ | **7.324486e−07 ± 1.72e−07** | **5240.1 ± 124.0** | N/A |
| | ED-EDA + ECMR0 | 6.267411e+00 ± 1.21e+01 | 300070.0 ± 0.0 | 5.52 ± 44.4 |
| | ED-EDA + ECMR0 + AVS | 7.378096e−07 ± 1.88e−07 | 11857.2 ± 1233.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 7.263456e−07 ± 1.80e−07 | 6969.6 ± 204.3 | 0.00 ± 0.0 |
| $f_2$ | $\text{UMDA}_c^G$ | **8.334779e−07 ± 1.21e−07** | **8304.1 ± 157.1** | N/A |
| | ED-EDA + ECMR0 | 6.905211e−01 ± 6.35e−01 | 300070.0 ± 0.0 | 410.14 ± 90.8 |
| | ED-EDA + ECMR0 + AVS | 8.594290e−07 ± 1.07e−07 | 22560.1 ± 29872.3 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 8.610971e−07 ± 1.16e−07 | 11865.2 ± 308.2 | 0.00 ± 0.0 |
| $f_3$ | $\text{UMDA}_c^G$ | 1.039245e+02 ± 9.75e+01 | 300070.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | 7.691274e+00 ± 1.28e+01 | 300070.0 ± 0.0 | 3.07 ± 2.4 |
| | ED-EDA + ECMR0 + AVS | **7.572571e−07 ± 1.73e−07** | **11820.6 ± 1234.7** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + EEDA | 8.040550e−07 ± 1.50e−07 | 136813.1 ± 18972.7 | 0.00 ± 0.0 |
| $f_9$ | $\text{UMDA}_c^G$ | **1.352954e+00 ± 1.06** | **250763.0 ± 109498.4** | N/A |
| | ED-EDA + ECMR0 | 1.180018e+01 ± 6.84 | 300070.0 ± 0.0 | 184.73 ± 612.9 |
| | ED-EDA + ECMR0 + AVS | 2.706852e+00 ± 2.10 | 284419.1 ± 58468.5 | 602.44 ± 575.9 |
| | ED-EDA + ECMR0 + EEDA | 2.285066e+01 ± 2.96 | 300070.0 ± 0.0 | 0.00 ± 0.0 |
| $f_{10}$ | $\text{UMDA}_c^G$ | 8.301565e−06 ± 7.46e−05 | 10970.2 ± 29202.4 | N/A |
| | ED-EDA + ECMR0 | 1.209118e+00 ± 7.82e−01 | 300070.0 ± 0.0 | 237.11 ± 667.0 |
| | ED-EDA + ECMR0 + AVS | 8.551754e−07 ± 1.07e−07 | 18170.5 ± 1789.6 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | **8.660051e−07 ± 1.08e−07** | **10860.3 ± 247.6** | **0.00 ± 0.0** |
| $f_{11}$ | $\text{UMDA}_c^G$ | 1.663107e−04 ± 1.66e−03 | 9117.9 ± 29390.5 | N/A |
| | ED-EDA + ECMR0 | 5.408410e−01 ± 2.77e−01 | 300070.0 ± 0.0 | 929.78 ± 1298.4 |
| | ED-EDA + ECMR0 + AVS | 5.496360e−03 ± 2.29e−02 | 69070.3 ± 96209.5 | 51.07 ± 143.4 |
| | ED-EDA + ECMR0 + EEDA | **7.762815e−07 ± 1.66e−07** | **13022.5 ± 3767.7** | **0.00 ± 0.0** |

ED-EDA+CMR0 + AVS represents AVS-IDEA [12]. And ED-EDA+CMR0 + EEDA represents EEDA. The differences between our algorithms and original algorithms include: (1) the application of error repairing; (2) different selection approaches (tournament selection, truncation selection, etc.); (3) different selection intensities and (4) different initialization of population. In order to compare different eigenvalue tuning strategies under fair circumstances, we implement them within the same framework of ED-EDA by using the same selection method, same selection intensity and same initialization method. In the literature of AVS and EEDA, results were drawn mainly based on non-uniformly initialized populations in the search space. Our experiments will extend previous results by applying the common uniformly initialization of population in all the experiments. In addition, we include another widely used EDA, $\mathrm{UMDA}_c^G$ [16] which adopts simple univariate marginal Gaussian model in the comparison. Model parameters of $\mathrm{UMDA}_c^G$ are also computed by maximum likelihood estimates.

So far, all kinds of existing Gaussian based continuous EDAs have been included in our studies. Their probabilistic distributions include univariate marginal Gaussian, multivariate Gaussian, and multivariate



(a) $f_1$

(b) $f_2$

(c) $f_3$

(d) $f_9$

(e) $f_{10}$

(f) $f_{11}$

Fig. 11. Average best fitness curves of small population-low dimension.

Gaussian with different covariance matrix scaling approaches. ED-EDA provides us a good environment to study all these algorithms under a unified framework.

### 6.2.1. Test functions and experiment setup

The test function suite is from [24]. We select six functions: unimodal functions $f_1, f_2, f_3$, and multimodal functions $f_9, f_{10}, f_{11}$, from the twenty-three benchmark functions. Details of these functions are in Appendix. We set up a low dimensional test suite in which all the six functions are 10d, and a high dimensional test suite in which the functions are 50d. The population sizes and selected sizes are set with two levels: a small population size of 100 with a selected size of 50, and a large population size of 2000 with a selected size of 1000. We use truncation selection in all the experiments. Therefore we have four different testing environments:

- Small population-low dimension.
- Small population-high dimension.
- Large population-low dimension.
- Large population-high dimension.

No free lunch theorem [23] indicates that under certain assumptions no single search algorithm is best on average for all problems. We will analyze which algorithm performs well under which environment(s). Other parameter settings are the same as the CMR comparison experiment in the previous subsection except that no convergence criterion is used here. Each algorithm on each test is evaluated on 100 runs. The activation numbers of ECMR0 are also recorded. Values of additional parameters of AVS are adopted from [12], i.e., $\eta^{\mathrm{DEC}} = 0.9, \eta^{\mathrm{INC}} = 1/\eta^{\mathrm{DEC}}, c^{\mathrm{AVS\text{-}MAX}} = 10.0, c^{\mathrm{AVS\text{-}MIN}} = 1/c^{\mathrm{AVS\text{-}MAX}}$, and initial value of $c^{\mathrm{AVS}}$ is 1. The results are shown in four groups respectively:

Table 3
Results of small population-high dimension

| Function | Algorithm | Best fitness | No. Eval. | No. ECMR0 |
|---|---|---|---|---|
| $f_1$ | UMDA$_c^G$ | 4.575777e−01 ± 2.02e+00 | 260157.2 ± 99422.2 | N/A |
| | ED-EDA + ECMR0 | 3.362485e+03 ± 8.95e+02 | 300070.0 ± 0.0 | 540.01 ± 355.9 |
| | ED-EDA + ECMR0 + AVS | 3.382982e+03 ± 8.23e+02 | 300070.0 ± 0.0 | 3022.53 ± 6.7 |
| | ED-EDA + ECMR0 + EEDA | **9.848507e−07 ± 1.54e−08** | **183813.3 ± 7075.4** | **1841.44 ± 71.7** |
| $f_2$ | UMDA$_c^G$ | **5.659719e−03 ± 3.44e−02** | **69642.6 ± 104810.5** | N/A |
| | ED-EDA + ECMR0 | 3.282926e+01 ± 4.04e+00 | 300070.0 ± 0.0 | 982.66 ± 256.9 |
| | ED-EDA + ECMR0 + AVS | 2.744233e+01 ± 3.51e+00 | 300070.0 ± 0.0 | 3026.35 ± 1.4 |
| | ED-EDA + ECMR0 + EEDA | 1.441234e+01 ± 3.27e+00 | 300070.0 ± 0.0 | 1267.73 ± 126.8 |
| $f_3$ | UMDA$_c^G$ | 1.354130e+04 ± 3.64e+03 | 300070.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | 5.150689e+03 ± 2.12e+03 | 300070.0 ± 0.0 | 348.47 ± 52.0 |
| | ED-EDA + ECMR0 + AVS | 4.356499e+03 ± 1.49e+03 | 300070.0 ± 0.0 | 3024.27 ± 5.9 |
| | ED-EDA + ECMR0 + EEDA | **2.903753e+02 ± 1.65e+02** | **300070.0 ± 0.0** | **3019.89 ± 2.6** |
| $f_9$ | UMDA$_c^G$ | **1.854362e+01 ± 3.70e+00** | **300070.0 ± 0.0** | N/A |
| | ED-EDA + ECMR0 | 2.934226e+02 ± 2.67e+01 | 300070.0 ± 0.0 | 669.14 ± 724.5 |
| | ED-EDA + ECMR0 + AVS | 2.044667e+02 ± 2.65e+01 | 300070.0 ± 0.0 | 3024.23 ± 7.2 |
| | ED-EDA + ECMR0 + EEDA | 3.220488e+02 ± 1.38e+01 | 300070.0 ± 0.0 | 3018.26 ± 2.7 |
| $f_{10}$ | UMDA$_c^G$ | **8.497928e−02 ±2.33e−01** | **297301.0 ± 27690.3** | N/A |
| | ED-EDA + ECMR0 | 9.323666e+00 ± 6.42e−01 | 300070.0 ± 0.0 | 513.63 ± 594.3 |
| | ED-EDA + ECMR0 + AVS | N/A[a] | 139913.7 ± 117164.7 | 1405.32 ± 1179.9 |
| | ED-EDA + ECMR0 + EEDA | 3.757219e+00 ± 6.95e−01 | 300070.0 ± 0.0 | 3013.39 ± 3.3 |
| $f_{11}$ | UMDA$_c^G$ | 1.027040e−01 ± 1.98e−01 | 285815.0 ± 62449.3 | N/A |
| | ED-EDA + ECMR0 | 2.935355e+01 ± 6.51e+00 | 300070.0 ± 0.0 | 503.03 ± 351.6 |
| | ED-EDA + ECMR0 + AVS | 3.049473e+01 ± 8.30e+00 | 300070.0 ± 0.0 | 3021.45 ± 8.5 |
| | ED-EDA + ECMR0 + EEDA | **6.628387e−03 ± 8.67e−03** | **300070.0 ± 0.0** | **3014.95 ± 2.9** |

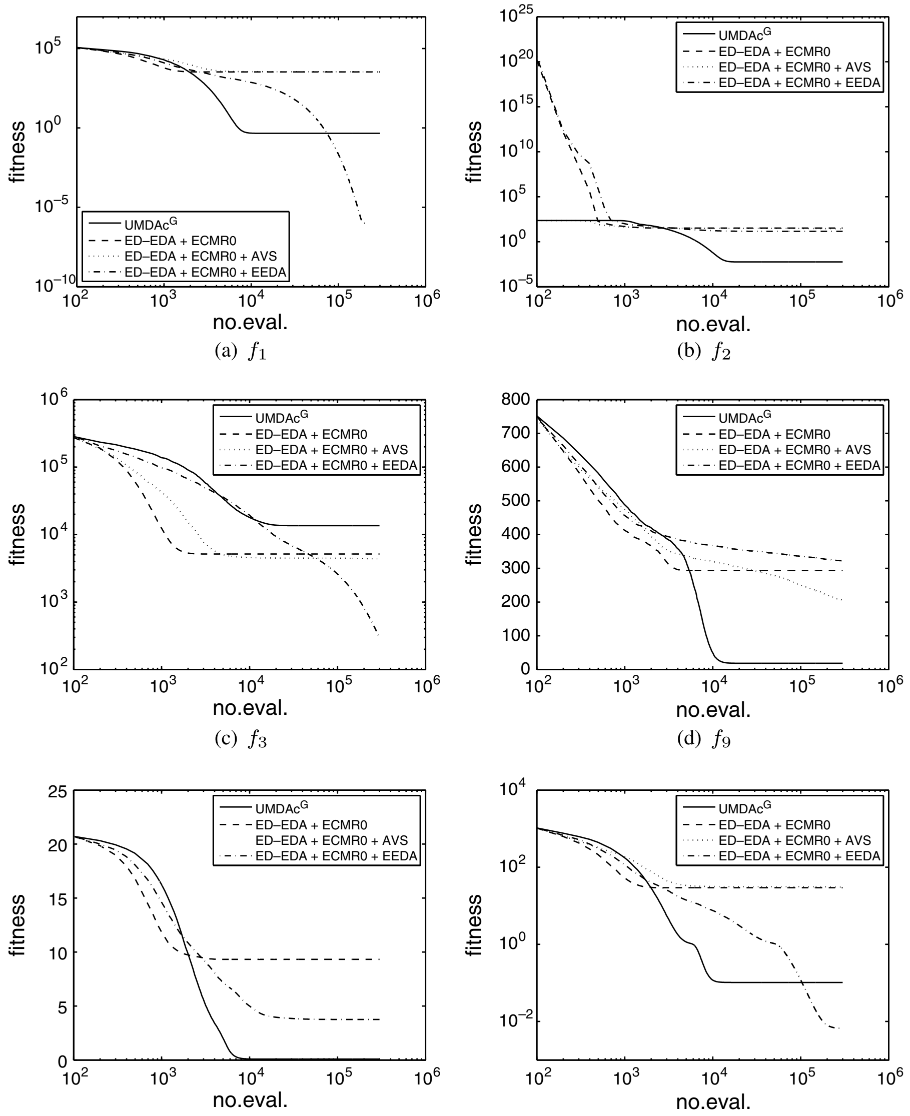
[a] 78 runs out of 100 break down because of data overflow. Thus the average best fitness is not calculated.

- Small population-low dimension: Table 2 and Fig. 11.
- Small population-high dimension: Table 3 and Fig. 12.
- Large population-low dimension: Table 4 and Fig. 13.
- Large population-high dimension: Table 5 and Fig. 14.

In all the tables, the best result on each problem is in bold. First we consider the best fitness achieved. If the best fitness values are the same, the algorithm that costs least evaluations is selected to be the best result. The axes of all figures are scaled either by linear or logarithmic scales in order to distinguish the four algorithms as much as possible.

### 6.2.2. Summary of experimental results

First, let us consider the activations of ECMR0 in the four tests. The sample sizes of covariance matrices in all the algorithms are identical to the selected sizes. We know that when the sample size is relatively small to



(a) $f_1$

(b) $f_2$

(c) $f_3$

(d) $f_9$

(e) $f_{10}$(*)

(f) $f_{11}$

Fig. 12. Average best fitness curves of small population-high dimension. (*) The curve of ED-EDA + ECMR0 + AVS on $f_{10}$ cannot be plotted because of data overflow.

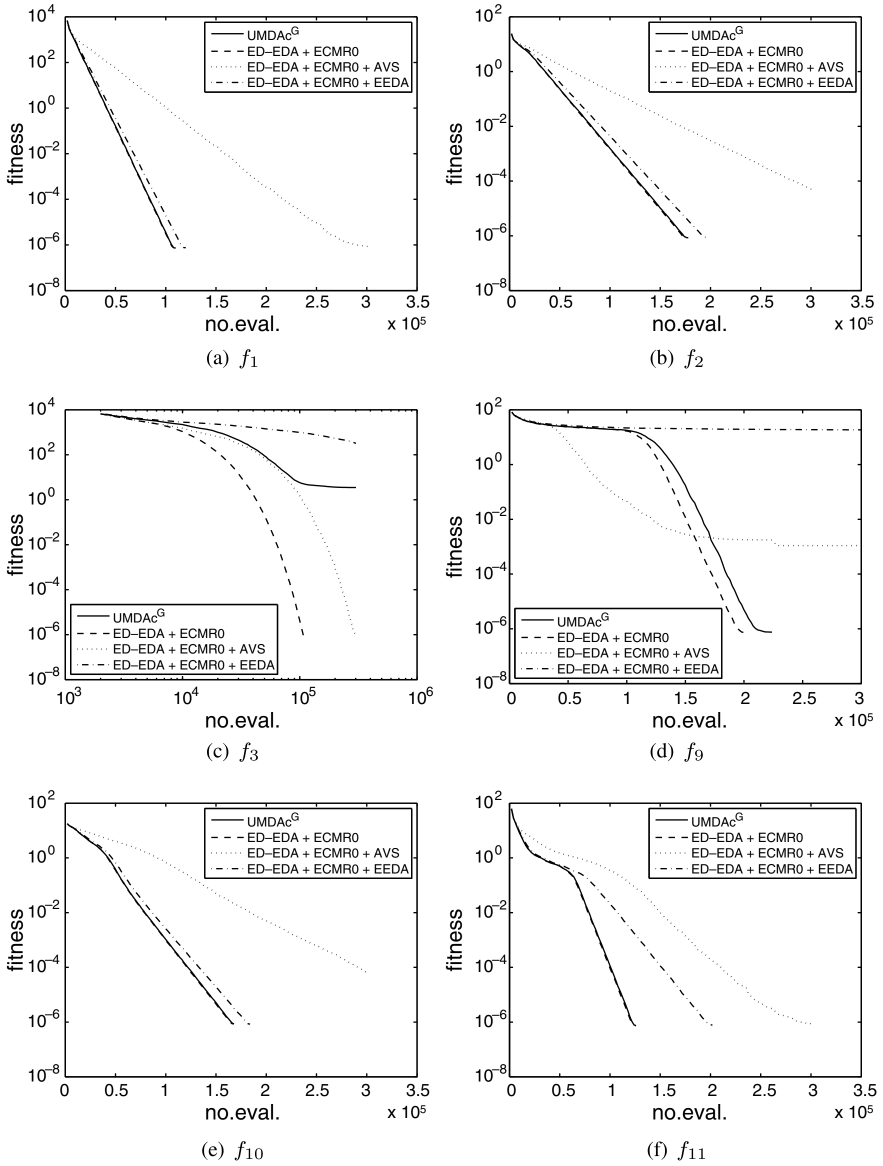
Table 4
Results of large population-low dimension

| Function | Algorithm | Best fitness | No. Eval. | No. ECMR0 |
|---|---|---|---|---|
| $f_1$ | UMDA$_c^G$ | 7.276811e−07 ± 1.72e−07 | 106687.6 ± 1573.1 | N/A |
| | ED-EDA + ECMR0 | **7.468662e−07 ± 1.72e−07** | **106087.9 ± 1455.8** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 8.571609e−07 ± 3.52e−07 | 249096.4 ± 29296.5 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 7.560365e−07 ± 1.60e−07 | 116102.9 ± 1648.9 | 0.00 ± 0.0 |
| $f_2$ | UMDA$_c^G$ | 8.426540e−07 ± 1.14e−07 | 174433.7 ± 1742.7 | N/A |
| | ED-EDA + ECMR0 | **8.558247e−07 ± 1.09e−07** | **172874.5 ± 1716.1** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 4.763670e−05 ± 7.34e−05 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 8.616955e−07 ± 9.65e−08 | 193184.4 ± 1763.4 | 0.00 ± 0.0 |
| $f_3$ | UMDA$_c^G$ | 3.499000e+00 ± 3.13e+00 | 301850.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | **7.604064e−07 ± 1.69e−07** | **107407.3 ± 1444.7** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 1.017514e−06 ± 1.31e−06 | 252114.9 ± 27847.6 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 3.234649e+02 ± 1.49e+02 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_9$ | UMDA$_c^G$ | 7.539530e−07 ± 1.80e−07 | 202599.6 ± 6604.8 | N/A |
| | ED-EDA + ECMR0 | **7.193811e−07 ± 1.88e−07** | **192824.5 ± 4284.7** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 1.093871e−03 ± 1.09e−02 | 209216.3 ± 30209.9 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 1.852634e+01 ± 2.50e+00 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_{10}$ | UMDA$_c^G$ | 8.544855e−07 ± 1.14e−07 | 165878.0 ± 1606.8 | N/A |
| | ED-EDA + ECMR0 | **8.559863e−07 ±1.13e−07** | **164978.5 ± 1433.5** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 6.245679e−05 ± 6.61e−05 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 8.418291e−07 ± 1.11e−07 | 180930.5 ± 1692.8 | 0.00 ± 0.0 |
| $f_{11}$ | UMDA$_c^G$ | 7.375143e−07 ± 1.83e−07 | 122899.5 ± 1692.4 | N/A |
| | ED-EDA + ECMR0 | **7.472051e−07 ± 1.67e−07** | **122059.9 ± 1577.3** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 8.638529e−07 ± 1.05e−06 | 236142.9 ± 32914.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 7.768699e−07 ± 1.63e−07 | 195843.0 ± 3282.3 | 0.00 ± 0.0 |

the problem size, the ill-posed covariance matrix/negative eigenvalue easily emerges. So ECMR0 should be activated more frequently in these cases. In other words, the smaller the ratio of $\frac{\text{selected size}}{\text{problem size}}$ is, the more probable ECMR0 activates. The ratios corresponding to our experiments are $\frac{50}{10}, \frac{50}{50}, \frac{1000}{10}$, and $\frac{1000}{50}$. Our results are exactly consistent with this judgement: The small population-high dimension test $\left(\frac{\text{selected size}}{\text{problem size}} = \frac{50}{50}\right)$ requires the activations of ECMR0 most, and the small population-low dimension test $\left(\frac{\text{selected size}}{\text{problem size}} = \frac{50}{10}\right)$ follows. The other two tests, large population-low dimension and large population-high dimension, hold sufficiently large value of the ratio, thus the probability of the appearance of negative eigenvalues is very low. Note that [8] presents the result that some MGEDA still needs repairing for some problems where the ratio equals to $\frac{1000}{30}$. That is to say, in our tests we have just avoided the error by luckily selecting 'good' problems. Computational error is inevitable and repairing an ill-posed covariance matrix/negative eigenvalue cannot be omitted in general.

From Table 2 we can see that AVS and EEDA strategies require less eigenvalue repairing and often the activation of ECMR0 comes with poor results. However, in Table 3 we see that a larger number of activations of ECMR0 brings opposite results, ED-EDA + ECMR0 + EEDA surprisingly performs the best in half of the tests. This is because the activation of ECMR0 (and other CMR methods) only depends on computational error, and such error varies with different problems and with different algorithm parameters, all we need to do is to remove the error when we observe it. However, when the covariance matrix needs repairing, it does not mean that the EDA has converged, or that only a poor result can be obtained. As long as the probabilistic model is accurate enough and competent for the current job, CMR methods do help to show the true searching ability of MGEDAs.

After repairing $D$ to a good condition, different strategies of tuning eigenvalues show different performances. Generally speaking, UMDA$_c^G$ and ED-EDA + ECMR0, which do not tune eigenvalues, perform well when the population size is large enough to establish an adequate initial distribution. In particular, ED-EDA + ECMR0 has almost the fastest converging speed among the four algorithms in both tests with a large population. The



(a) $f_1$

(b) $f_2$

(c) $f_3$

(d) $f_9$

(e) $f_{10}$

(f) $f_{11}$

Fig. 13. Average best fitness curves of large population-low dimension.

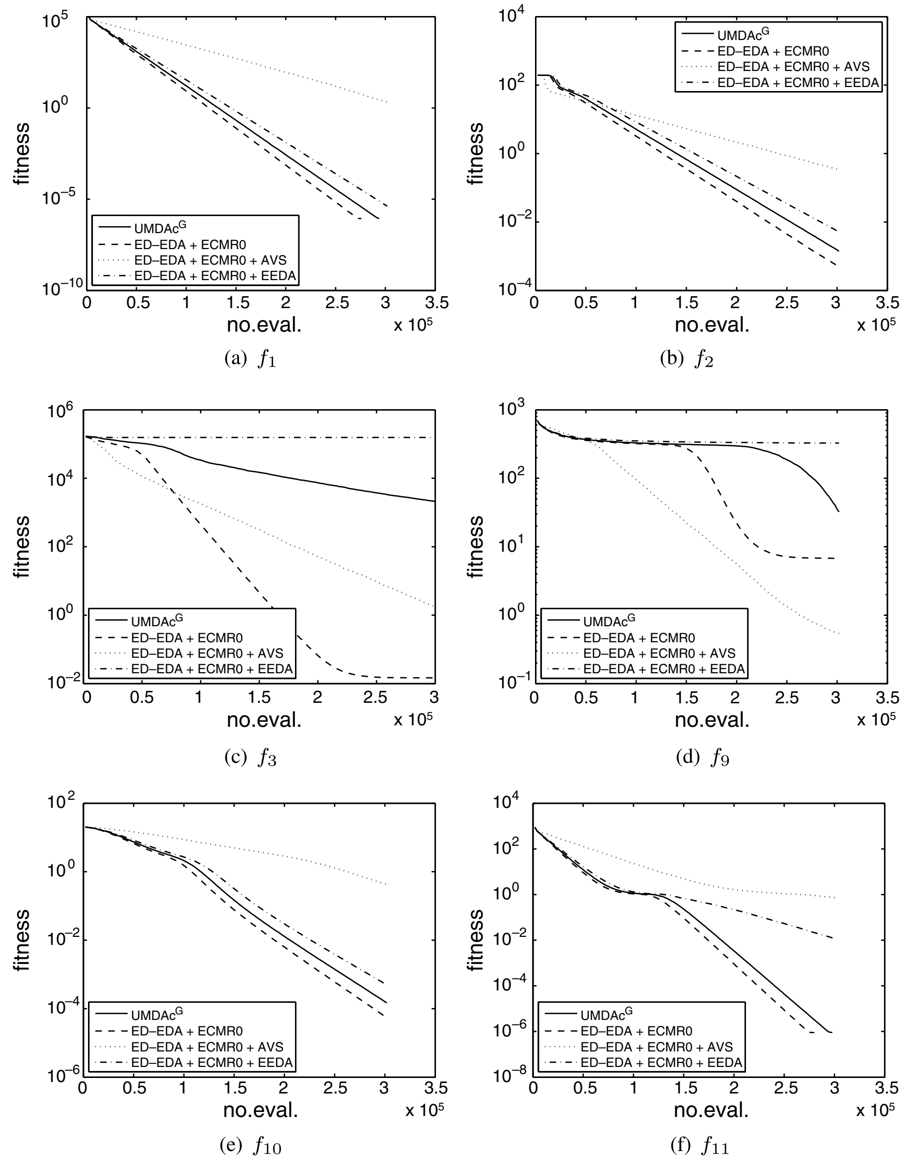
advantage of UMDA$_c^G$ is not apparent on $f_3$ because the quadratic relation between variables is hard to learn for the univariate marginal model. Curves of 50d $f_9$ also show a little exception partially because the 300,000 fitness evaluation limit is not adequate to see their full behaviors. In general, when the population size is large enough, multivariate Gaussian performs more efficiently than univariate marginal Gaussian. And these two maximum likelihood estimated models are more efficient than those using the eigenvalue tuning strategies. Sufficiently large population sizes help the maximum likelihood estimates to be precise and reliable. ED-EDA + ECMR0 + AVS and ED-EDA + ECMR0 + EEDA usually converge slower than UMDA$_c^G$ and ED-EDA + ECMR0 in the large population tests, because AVS and EEDA both enlarge the covariance matrix so that the success rate of sampling better individuals decreases if the current distribution has already been precisely learnt by maximum likelihood estimates. However, maximum likelihood estimated Gaussian's performances strongly depend on the adequacy of the initial distribution. With a sufficiently large population size for a given problem, they perform pretty well, whereas an insufficiently large population size can result in poor results.

Table 5
Results of large population-high dimension

| Function | Algorithm | Best fitness | No. Eval. | No. ECMR0 |
|---|---|---|---|---|
| $f_1$ | $\text{UMDA}_c^G$ | 8.826993e−07 ± 8.61e−08 | 291994.9 ± 1399.3 | N/A |
| | ED-EDA + ECMR0 | **8.720652e−07 ± 7.95e−08** | **271865.0 ± 1448.8** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 2.084492e+00 ± 2.25e+00 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 4.134877e−06 ± 4.53e−07 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_2$ | $\text{UMDA}_c^G$ | 1.406607e−03 ± 8.67e−05 | 301850.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | **4.977453e−04 ± 2.61e−04** | **301850.0 ± 0.0** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 3.347511e−01 ± 2.15e−01 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 5.143610e−03 ± 2.96e−04 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_3$ | $\text{UMDA}_c^G$ | 2.046511e+03 ± 3.44e+02 | 301850.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | **1.472998e−02 ± 6.12e−02** | **301850.0 ± 0.0** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 1.656897e+00 ± 1.97e+00 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 1.556182e+05 ± 2.61e+04 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_9$ | $\text{UMDA}_c^G$ | 3.222134e+01 ± 1.51e+01 | 301850.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | 6.770974e+00 ± 5.01e+00 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + AVS | **5.419324e−01 ± 2.01e+00** | **301850.0 ± 0.0** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + EEDA | 3.260484e+02 ± 9.77e+00 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_{10}$ | $\text{UMDA}_c^G$ | 1.473622e−04 ± 8.12e−06 | 301850.0 ± 0.0 | N/A |
| | ED-EDA + ECMR0 | **5.474890e−05 ± 8.41e−06** | **301850.0 ± 0.0** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 4.129898e−01 ± 3.22e−01 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 4.925867e−04 ± 2.75e−05 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| $f_{11}$ | $\text{UMDA}_c^G$ | 8.982967e−07 ± 7.26e−08 | 294393.7 ± 1499.6 | N/A |
| | ED-EDA + ECMR0 | **8.856177e−07 ± 7.54e−08** | **274243.8 ± 1697.5** | **0.00 ± 0.0** |
| | ED-EDA + ECMR0 + AVS | 7.114941e−01 ± 2.81e−01 | 301850.0 ± 0.0 | 0.00 ± 0.0 |
| | ED-EDA + ECMR0 + EEDA | 1.114116e−02 ± 1.31e−03 | 301850.0 ± 0.0 | 0.00 ± 0.0 |

Eigenvalue tuning shows better robustness to reach the optimum under different populations in all the tests. Especially for the two tests with a small population, AVS and EEDA show exciting exploring efficiency. For example, on $f_3, f_{10}, f_{11}$ in low dimensional tests, and on $f_1, f_3, f_{11}$ in high dimensional tests, for which 50 selected individuals are too sparse in the 10d and 50d search spaces, ED-EDA + ECMR0 + EEDA takes advantage of the approximate negative gradient direction to show the most remarkable and stable performance. ED-EDA + ECMR0 + AVS also performs well but usually needs more evaluations than ED-EDA + ECMR0 + EEDA to converge. This can be explained as that AVS scales a covariance matrix not only in the approximate negative gradient direction but also in every other eigenvector direction. This may waste some attempts in the directions which are not necessary in terms of finding the optimum. Thus its convergence rate is slowed down. Simultaneously, there are also opposite results on $f_3$ in the small population-low dimension test and large population-low dimension test, and $f_9$ in all the four tests. For these cases, ED-EDA + ECMR0 + AVS converges faster than ED-EDA + ECMR0 + EEDA. This can be explained as that on $f_3$ and $f_9$, the best searching direction may not be well approximated only in EEDA manner, while AVS profits from scaling up the trying region sufficiently. However, by checking through all the tests, ED-EDA + ECMR0 + AVS does not perform as stable as the other three algorithms, especially on the two large dimensional tests.

Another interesting observation is that in small population tests, for additive separable functions $f_1, f_2, f_9$ and $f_{10}$, the appropriate model structure of $\text{UMDA}_c^G$ helps it achieve very good results, because its model structure highly fits the problem structure. When the population size is large enough, $\text{UMDA}_c^G$ can converge faster than ED-EDA + ECMR0 + EEDA (see $f_1, f_2$ and $f_9$ in Table 2, $f_2, f_9$ and $f_{10}$ in Table 3), while ED-EDA + ECMR0 (multivariate Gaussian without tuning) fails because the shape of the distribution converges too fast due to inaccurate estimates. The simplicity of $\text{UMDA}_c^G$ here shows its capability: When only a rough estimate can be gained, complicated multivariate Gaussian model may perform even worse than the simple univariate marginal Gaussian. For more generalized cases, i.e., the problem structure is beyond the capacity



(a) $f_1$

(b) $f_2$

(c) $f_3$

(d) $f_9$

(e) $f_{10}$

(f) $f_{11}$

Fig. 14. Average best fitness curves of large population-high dimension.

of the simple model, proper eigenvalue tuning can significantly enhance the searching ability of inaccurately estimated multivariate Gaussian (ED-EDA + ECMR0 + EEDA in our tests).

The efficiency of eigenvalue tuning is also evident by looking at the no. eval. axes in all the figures: ED-EDA + ECMR0 + AVS and ED-EDA + ECMR0 + EEDA are more efficient when the population size is small; for test pairs of same dimensions, in the small population test, the optimum is usually reached by the fastest algorithm with much less evaluations than those in the large population test. That seems to be the most attractive property of ED-EDA with eigenvalue scaling: When a problem can be solved by proper eigenvalue tuning with a small population, it costs much less evaluations. EDA can be more efficient.

In summary, eigenvalue tuning strategies, especially non-uniform eigenvalue scaling in our tests, show promising performances when a multivariate Gaussian suffers from inaccurate estimates that are primarily caused by an insufficient population size. The importance of CMR methods is also demonstrated simultaneously. Only for a large enough population does the classical maximum likelihood estimated Gaussian model have good performance. If the merits of eigenvalue tuning can be exerted in combination with classical Gaussian, we can make EDA solve problems with even smaller population sizes to achieve higher efficiency.

### *6.2.3. Additional discussions on AVS*

There is another remarkable convergence issue of AVS observed in our tests. We have stated the lack of convergence proof of existing eigenvalue scaling strategies in Section 2.1. Compared with EEDA, AVS scales the covariance matrix faster and more intensively. The mechanism of AVS determines that if no improvement is achieved in the current generation, then the covariance matrix should be scaled up. When the population size is too small for the problem size, the attempts of new individuals are likely to be too sparse to find better solutions even if the current distribution is already adequate and accurate. In this case, AVS may be misguided and keep on scaling up the covariance matrix.

We did find such an example of AVS on $f_{10}$ in the small population-high dimension test. When we recorded the average fitness of population and the standard error in ED-EDA + ECMR0 + AVS, 78 runs out of 100 broke down because the sum of fitness and the sum of fitness square used for calculating standard error exceeded the maximum real value that the computer could represent. Therefore the average best fitness of ED-EDA + ECMR0 + AVS in Table 3 was not presented. But the number of evaluations and the number of ECMR0's activations both were recorded using the data before those runs broke down. We plot a typical run just before the breakdown of AVS, i.e. on 50d $f_{10}$ with population size 100 and selected size 50, in Fig. 15.

As we can see, after about 1e4 evaluations, the average fitness curve should have converged to the best fitness curve more closely. But due to not finding new better solutions, AVS scaled up the covariance matrix to explore larger areas. It misled the population back to a high level of average fitness. Note the best fitness curve remained still because we used the elitist approach. Still no better solution was found during the following generations, which resulted in another wrong jump of the average fitness. Some individuals even distributed outside the initial domain boundary of the search space. Just at that time, when calculating the sum of fitness and the sum of fitness square (they kept on increasing rapidly because widely dispersed individuals have huge objective fitness values due to the shape of the function), the data overflew and flipped to negative values in the computer. So the curve of the next huge jump could not be recorded. Such abnormal behavior is partially related to the value of parameters of AVS, the selection intensity of the algorithm and also the shape of objective function.

The latest literature of AVS [2] has presented a new triggering method, standard-deviation-ratio (SDR) trigger instead of correlation-triggered (CT). However the main idea of scaling covariance matrix uniformly by AVS remains the same. SDR-AVS, as well as AVS and CT-AVS, can all be regarded as uniform eigenvalue scaling strategies, but with different triggering methods. Thus our results still make sense.

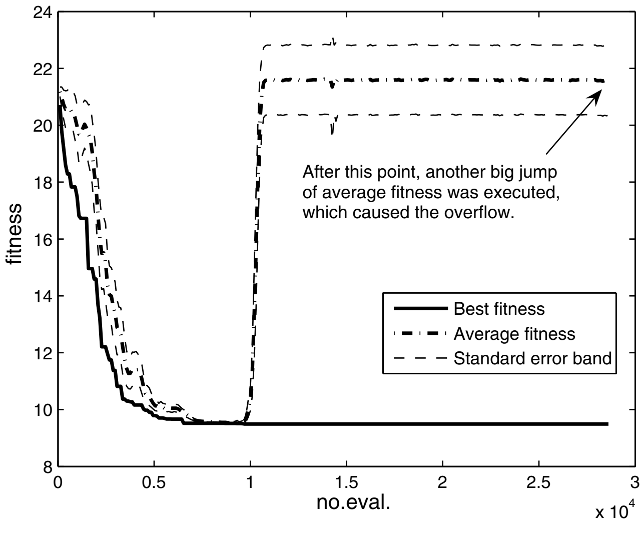


Fig. 15. A typical run just before breakdown of AVS on $f_{10}$.

## 7. Conclusions

We have proposed a unified framework ED-EDA to study Multivariate Gaussian based EDAs (MGEDAs) in this paper. Through eigen analysis, we have shown that the existing MGEDAs only differ from each other on different strategies of tuning eigenvalues. All existing MGEDAs can be unified within the same ED-EDA framework. ED-EDA has the beneficial property of explicitly presenting the eigenvalue tuning step. Different eigenvalue tuning strategies can be efficiently implemented in ED-EDA by substituting/modifying the step of tuning eigenvalue matrix $D$. New covariance matrix/eigenvalue repairing methods are proposed in ED-EDA. ECMR and ECMR0 show reliable abilities to repair negative eigenvalues of ill-posed covariance matrices which are caused by inevitable computational errors. Combinations of ECMR0 with different eigenvalue tuning strategies are successfully applied to benchmark optimization problems.

Our experiments have shown the promising exploring efficiency of eigenvalue tuning strategies. With appropriate eigenvalue tuning, we can reduce the population size when solving a problem, and at the same time spend fewer evaluations in reaching the optimum. For EDAs using classical maximum likelihood estimated Gaussian models without eigenvalue tuning, solving problems with a small population size is quite difficult. We observed the convergence issue of AVS in our experiments, which warrants future theoretical research on eigenvalue tuning techniques. We believe that non-uniform eigenvalue scaling will play a more important role in the future research of MGEDAs. Given our unified ED-EDA framework, other more efficient EDAs can be developed for different classes of optimization problems in the future.

## Acknowledgements

The authors are grateful to reviewers for their constructive comments and to Gongqing Zhang for his help to improve the paper, and to Felicity Simon for her proof-reading. This work is partially supported by a grant from the Chinese Academy of Sciences for Outstanding Young Scholars from Overseas (No. 2F03B01) and a grant from EPSRC (EP/D052785/1).

## Appendix A

All the functions listed below are $k$-dimensional.

### A.1. Functions used in comparing CMR, ECMR and ECMR0

Four functions are included:

$f_8$: Generalized Schwefel's Problem 2.26, minimization problem

$$f_8(x) = -\sum_{i=1}^{k} x_i \sin\left(\sqrt{|x_i|}\right), \tag{15}$$

where $-500 \leqslant x_i \leqslant 500, \min(f_8) = f_8(420.9687, \ldots, 420.9687) = -418.9829k$.

$f_{13}$: Generalized penalized function, minimization problem

$$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{k-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_k - 1)^2[1 + \sin^2(2\pi x_k)]\} + \sum_{i=1}^{k} u(x_i, 5, 100, 4), \tag{16}$$

where $-50 \leqslant x_i \leqslant 50, \min(f_{13}) = f_{13}(1, \ldots, 1) = 0$ and

$$u(x_i, a, j, m) = \begin{cases} j(x_i - a)^m, & x_i > a, \\ 0, & -a \leqslant x_i \leqslant a, \\ j(-x_i - a)^m, & x_i < -a. \end{cases} \tag{17}$$

*Rosenbrock*: Generalized Rosenbrock's function, minimization problem

$$F(x) = \sum_{i=1}^{k-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \tag{18}$$

where $-10 \leqslant x_i \leqslant 10, min(F) = F(1, \ldots, 1) = 0$.

*SumCan*: Summation cancellation function, maximization problem

$$F(x) = \frac{1}{10^{-5} + \sum_{i=1}^{k}|y_i|}, \tag{19}$$

where $y_1 = x_1, y_i = x_i + y_i - 1, i = 2, \ldots, k$, and $-0.16 \leqslant x_i \leqslant 0.16, \max(F) = F(0, \ldots, 0) = 10^5$.

### *A.2. Functions used in comparing eigenvalue tuning strategies*

Six functions are included:

$f_1$: Sphere model, minimization problem

$$f_1(x) = \sum_{i=1}^{k} x_i^2, \tag{20}$$

where $-100 \leqslant x_i \leqslant 100, \min(f_1) = f_1(0, \ldots, 0) = 0$.

$f_2$: Schwefel's Problem 2.22, minimization problem

$$f_2(x) = \sum_{i=1}^{k} |x_i| + \prod_{i=1}^{k} |x_i|, \tag{21}$$

where $-10 \leqslant x_i \leqslant 10, \min(f_2) = f_2(0, \ldots, 0) = 0$.

$f_3$: Schwefel's Problem 1.2, minimization problem

$$f_3(x) = \sum_{i=1}^{k} \left( \sum_{j=1}^{i} x_j \right)^2, \tag{22}$$

where $-100 \leqslant x_i \leqslant 100, \min(f_3) = f_3(0, \ldots, 0) = 0$.

$f_9$: Generalized Rastrigin's function, minimization problem

$$f_9(x) = \sum_{i=1}^{k}[x_i^2 - 10\cos(2\pi x_i) + 10], \tag{23}$$

where $-5.12 \leqslant x_i \leqslant 5.12, \min(f_9) = f_9(0, \ldots, 0) = 0$.

$f_{10}$: Ackley's function, minimization problem

$$f_{10}(x) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{k}\sum_{i=1}^{k} x_i^2}\right) - \exp\left(\frac{1}{k}\sum_{i=1}^{k} \cos 2\pi x_i\right) + 20 + e, \tag{24}$$

where $-32 \leqslant x_i \leqslant 32, \min(f_{10}) = f_{10}(0, \ldots, 0) = 0$.

$f_{11}$: Generalized Griewank function, minimization problem

$$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{k} x_i^2 - \prod_{i=1}^{k} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \tag{25}$$

where $-600 \leqslant x_i \leqslant 600, \min(f_{11}) = f_{11}(0, \ldots, 0) = 0$.

## References

[1] P.A.N. Bosman, J. Grahl, Matching Inductive Search Bias and Problem Structure in Continuous Estimation-of-Distribution Algorithms, Technical Report 03/2005, Dept. of Logistics, Mannheim Business School, 2005.

[2] P.A.N. Bosman, J. Grahl, F. Rothlauf, SDR: A better trigger for adaptive variance scaling in normal EDAs, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2007, ACM Press, New York, NY, USA, 2007, pp. 492–499.

[3] P.A.N. Bosman, D. Thierens, An Algorithmic Framework for Density Estimation based Evolutionary Algorithms, Utrecht University Technical Report UU-CS-1999-46, 1999.

[4] P.A.N. Bosman, D. Thierens, Expanding from discrete to continuous estimation of distribution algorithms: the IDEA, in: Proceedings of the Sixth International Conference on Parallel Problem Solving From Nature – PPSN VI, Springer-Verlag, Berlin, 2000, pp. 767–776.

[5] P.A.N. Bosman, D. Thierens, Continuous iterated density estimation evolutionary algorithms within the IDEA framework, in: Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference, GECCO-2000, Morgan Kauffmann, San Francisco, California, 2000, pp. 197–200.

[6] C. Chatfield, A.J. Collin, Introduction to Multivariate Analysis, Science Paperbacks, 1980.

[7] L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, New York, 1986.

[8] W. Dong, X. Yao, Covariance matrix repairing in Gaussian based EDAs, in: 2007 IEEE Congress on Evolutionary Computation (CEC2007), Singapore, 2007, pp. 415–422.

[9] M.R. Gallagher, M. Frean, Population-based continuous optimization, probabilistic modelling and mean shift, Evol. Comput. 13 (2005) 29–42.

[10] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts, USA, 1989.

[11] C. Gonzlez, J.A. Lozano, P. Larrañaga, Mathematical modelling of UMDAc algorithm with tournament selection. Behaviour on linear and quadratic functions, Int. J. Approxim. Reason. 31 (2002) 313–340.

[12] J. Grahl, P.A.N. Bosman, F. Rothlauf, The correlation-triggered adaptive variance scaling IDEA, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2006, ACM Press, New York, New York, 2006, pp. 397–404.

[13] J. Grahl, S. Minner, F. Rothlauf, Behaviour of UMDAc with truncation selection on monotonous functions, in: Proceedings of the Congress on Evolutionary Computation (CEC2005), IEEE Press, Piscataway, New Jersey, 2005, pp. 2553–2559.

[14] I.T. Jolliffe, Principle Component Analysis, Springer-Verlag, 1986.

[15] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO-2000, ACM Press, New York, NY, USA, 2000, pp. 201–204.

[16] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer, Norwel, MA, 2001.

[17] O. Ledoit, M. Wolf, A well-conditioned estimator for large-dimensional covariance matrices, J. Multivar. Anal. 88 (2004) 365–411.

[18] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, Lecture Notes in Computer Science 1411, in: Parallel Problem Solving from Nature – PPSN IV, Springer-Verlag, London, UK, 1996, pp. 178–187.

[19] T. Paul, H. Iba, Real-coded estimation of distribution algorithm, in: Proceedings of the 5th Metaheuristics International Conference, 2003, pp. 61–66.

[20] M. Pelikan, D. Goldberg, F. Lobo, A survey to optimization by building and using probabilistic models, Comput. Optim. Appl. 21 (2002) 5–20.

[21] V. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1998.

[22] M.R. Wagner, A. Auger, M. Schoenauer, EEDA: A New Robust Estimation of Distribution Algorithm, Rapport de Recherche (Research Report) RR-5190, INRIA, 2004.

[23] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.

[24] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.

[25] J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, J. Mach. Learning Res. 6 (2005) 483–502.