



# A Layered Learning Estimation of Distribution Algorithm

Yong Li, Qiang Yang\*, Xu-Dong Gao, Zhen-Yu Lu

School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China.

Jun Zhang

Department of Electrical and Electronic Engineering, Hanyang University, Ansan, Korea.

Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan.

\*Corresponding author: Qiang Yang (mmyq@126.com)

## ABSTRACT

Though estimation of distribution algorithms (EDAs) have witnessed success in problem optimization, most of them suffer from sharp shrink of covariance, which may lead to premature convergence. To alleviate this issue, this paper proposes a layered learning estimation of distribution algorithm (LLEDA) by maintaining multiple probability distribution models. Specifically, LLEDA first separates the population into several layers based on fitness. Then, the mean position of each layer is computed. Subsequently, we let the estimated mean position in each lower layer randomly learn from the one of a randomly selected higher layer, so that the mean positions of lower layers could be promoted to be close to promising areas found in the current population. At last, the covariance of each layer is estimated based on the generated new mean position and the individuals in this layer. By this means, multiple probability models with high quality are maintained and then are used to sample promising and diversified offspring separately. Comparative experiments conducted on a widely used benchmark problem set demonstrate that the proposed LLEDA achieves competitive or even much better performance than several state-of-the-art and representative EDAs.

## CCS CONCEPTS

• **Mathematics of computing** → **Evolutionary algorithm; Bio-inspired optimization;**

## KEYWORDS

Estimation of Distribution Algorithms, Multivariate Gaussian Distribution, Layered Learning, Multiple Distribution Models, Global Numerical Optimization

## ACM Reference format:

Yong Li, Qiang Yang, Xu-Dong Gao, Zhen-Yu Lu and Jun Zhang. 2022. A Layered Learning Estimation of Distribution Algorithm. In *Proceedings of Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion)*, July 9–13, 2022, Boston, MA, USA. ACM, New York, NY, USA, 4 pages.  
<https://doi.org/10.1145/3520304.3528904>

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

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA  
© 2022 Copyright is held by the owner/author(s).

## 1 INTRODUCTION

Estimation of distribution algorithms (EDAs) [1-4] have been regarded as a special branch of evolutionary algorithms (EAs) since it was proposed in 2001 [5]. Different from traditional EAs, like particle swarm optimization (PSO) [6-8], which generate offspring based on individuals, EDAs generate new solutions via a specific probability distribution model, which is estimated based on high-quality solutions in the current population. Due to the random sampling of offspring, EDAs generally preserve high search diversity and strong robustness [9].

In the literature, most EDAs use the Gaussian model to evaluate the probability distribution, and then sample new solutions based on the estimated Gaussian model [10]. These EDAs are usually called Gaussian estimation of distribution algorithms (GEDAs). Based on whether the correlation between variables is considered, existing GEDAs are mainly divided into two categories [9], namely univariate GEDAs (UGEDAs) [10] and multivariate GEDAs (MGEDAs) [11]. Compared with UGEDAs, MGEDAs generally achieve better performance especially on problems with many interacting variables [9].

However, when dealing with complex optimization problems, especially those with many wide and flat local basins [11], most MGEDAs suffer from premature convergence due to the rapid shrink of variances [9]. To alleviate this deficiency, researchers have focused their attention on designing novel techniques to promote the search diversity of EDAs. For instance, in [12], Grahl *et al.* proposed a correlation-triggered adaptive covariance scaling strategy to avoid premature convergence by adaptively multiplying a factor to the estimated variance based on the change of the global best solution. In [13], Bosman *et al.* designed an indicator, named standard-deviation ratio (SDR), to capture the improvements and then used this measure to scale the covariance when the captured improvements are far away from the mean vector. In [14], Cai *et al.* proposed a cross-entropy based adaptive covariance scaling method. In this approach, the covariance scaling factor is computed by minimizing the cross-entropy between the sampled population and the prediction of the probabilistic model. In [2], Liang *et al.* proposed a new GEDA variant, named EDA2, which utilized an archive to store historical information, and then estimated the distribution model through this archive to enlarge the covariance.

In addition, some researchers have also proposed some significant improvements in the mean vector estimation. For instance, Bosman *et al.* [15] designed an anticipated mean shift (AMS) scheme to shift the mean vector of the Gaussian distribution,

and then used the shifted mean vector along with the selected solutions to estimate the covariance matrix. In this way, the search direction could be corrected to a certain extent. In [16], Ren *et al.* proposed to utilize different mean vector shifts according to the difference between the mean vector estimated in the current generation and the one in the previous generation.

Even though a large number of significant EDA variants have been developed and achieved good performance in problem optimization, they all maintain only one distribution model during the evolution. Besides, the maintained distribution model is estimated only on a very small number of high-quality solutions with the others discarded. This will lead to serious loss of evolutionary information and heavy sacrifice of search diversity.

To alleviate this predicament, this paper devises a layered learning EDA (LLEDA). Different from existing MGEDAs with only one distribution model in the population, LLEDA maintains multiple probability distribution models for the population in each generation. To be specific, the current parent population is first sorted based on the fitness from the best to the worst, and then it is divided into several layers. Subsequently, each layer uses its own individuals to construct a distribution model. To improve the quality of sampled individuals, a learning method is introduced to let the mean vectors in the lower layers learn from the one randomly selected from higher layers. In this way, the distribution models in lower layers are expectedly promoted and thus the sampled offspring could be improved. With multiple distribution models, the search diversity of LLEDA is expectedly amplified largely, which is beneficial for EDA to escape from local basins.

To validate the effectiveness of LLEDA, we conduct experiments on the commonly used CEC 2014 benchmark set [17] by comparing LLEDA with several state-of-the-art EDA variants.

## 2 PROPOSED LLEDA

Given that  $NP$  individuals are maintained in the population, the proposed LLEDA first sorts all individuals based on their fitness from the best to the worst. Then, the  $NP$  individuals are divided into  $NL$  layers equally with each layer containing  $h_i = NP/NL$  ( $1 \leq i \leq NL$ ) individuals (if  $NP\%NL \neq 0$ , the remaining  $NP\%NL$  individuals are equally placed into the first  $NP\%NL$  layers). For better

---

### Algorithm 1: The overall procedure of LLEDA

---

**Input:** population size  $NP$ , Number of layers  $NL$ , Learning step  $F$ ;

1: Initialize  $NP$  individuals randomly, evaluate their fitness, and  $FES = NP$ ;

2: Obtain the global best solution  $Gbest$ ;

3: **While** ( $FES < FES_{max}$ )

4: Divide the population into  $NL$  layers and calculate the mean vector of each layer based on Eq. (1);

5: Except for the first layer, let each layer learn from a randomly selected higher layer based on Eq. (2);

6: Estimate the covariance matrix  $C_i$  of each layer according to Eq. (3);

7: Each layer separately samples  $h_i$  offspring randomly, calculates their fitness, and sets  $FES = FES + h_i$ ;

8: Combine the offspring in the last generation and the ones in the current generation to select the  $NP$  best individuals to form the parent population;

9: Update the global best solution  $Gbest$ ;

10: Execute the local search method 2 times on  $Gbest$ , and  $FES = FES + 2$ ;

11: **End While**

**Output:** the global best solution  $Gbest$ ;

---

understanding, we denote each layer by  $l_i$  ( $1 \leq i \leq NL$ ) and the smaller the layer index is, the better the individuals in this layer are.

After the partition of the population, each layer first separately estimates its own probability distribution with respect to the mean vector of the Gaussian distribution as follows:

$$\mu_i = \frac{1}{h_i} \sum_{j=1}^{h_i} S_i^j \quad (1 \leq i \leq NL) \quad (1)$$

where  $\mu_i$  denotes the mean vector of the  $i$ th layer,  $S_i$  represents the set of individuals in this layer,  $S_i^j$  is the  $j$ th individual in the  $i$ th layer, and  $h_i$  is the number of individuals in this layer.

By means of the above manner, each layer has its own mean vector, and the higher the layer is, the closer to promising areas its mean vector expectedly is. Since the mean vectors of lower layers are slightly far away from the promising areas, we develop a layered learning strategy for lower layers to promote the quality of their distribution models, so that promising offspring could be sampled. Specifically, we let the mean vector of each lower layer learn from a randomly selected one of a higher layer in the following way:

$$\mu_{i,new} = \mu_i + F * (\mu_{rand} - \mu_i) \quad (1 \leq i \leq NL, 1 \leq rand < i) \quad (2)$$

where  $\mu_i$  represents the mean vector of the  $i$ th layer, while  $\mu_{rand}$  is the mean vector of a higher layer, which is randomly selected from those layers higher than the  $i$ th layer. That is to say,  $rand$  is randomly sampled within  $[1, i]$ .

By means of Equation (2), the mean vector of each lower layer is expectedly closer to promising areas, and thus high-quality offspring are expectedly sampled. It should be mentioned that for the first layer, namely  $l_1$ , since there is no higher layer for it to learn from, we keep its mean vector unchanged.

After the learning of the mean vector of each layer, the covariance of each layer is then estimated by using the updated mean vector and the individuals in this layer in the following way:

$$C_i = \frac{1}{h_i - 1} (S_i - \mu_{i,new})(S_i - \mu_i)^T \quad (1 \leq i \leq NL) \quad (3)$$

With the above manners, multiple high quality probability distribution models are maintained in the population. Therefore, with these distribution models, diversified offspring could be generated and hence high search diversity could be maintained during the evolution, which is beneficial for the population to escape from local areas.

To further enhance the balance between high search diversity and fast convergence speed, we further introduce a cross-generation individual selection method to select individuals for the parent population. Specifically, we combine the offspring in the last generation and the ones in the current generation together and then select the  $NP$  best individuals to form the parent population for the next generation. Subsequently, the parent population is divided into multiple layers and multiple distribution models are estimated.

Since EDAs preserve poor local exploitation ability, we further conduct the following local search method around the global best position ( $Gbest$ ) to improve its accuracy:

$$Ind = Gaussian(Gbest, 1 \times 10^{-4}) \quad (4)$$

In particular, we conduct the local search method in each

TABLE 1. Comparison between LLEDA and the compared state-of-the-art EDA variants on the 100-*D* CEC2014 benchmark functions.

| <i>F</i>               | Quality | LLEDA       | EDA2                         | EDA <sub>VERS</sub>          | EDA/LS                       | EDA/LS-MS                    | TRA-EDA                      | BUMDA                        |
|------------------------|---------|-------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
| <i>F</i> <sub>1</sub>  | Mean    | 1.47E-06    | 7.24E-13                     | <b>1.22E+06</b> <sup>+</sup> | <b>8.24E+09</b> <sup>+</sup> | <b>4.41E+08</b> <sup>+</sup> | <b>4.62E+08</b> <sup>+</sup> | <b>8.48E+07</b> <sup>+</sup> |
|                        | Std     | 7.85E-06    | 1.70E-13                     | 1.68E+05                     | 8.95E+09                     | 1.03E+09                     | 6.07E+07                     | 8.05E+06                     |
| <i>F</i> <sub>2</sub>  | Mean    | 6.45E-05    | 9.67E-11                     | <b>1.12E+04</b> <sup>+</sup> | 1.54E-09                     | <b>3.87E+01</b> <sup>+</sup> | <b>9.74E+10</b> <sup>+</sup> | <b>7.15E+04</b> <sup>+</sup> |
|                        | Std     | 3.15E-04    | 2.78E-11                     | 5.42E+03                     | 3.04E-09                     | 2.08E+02                     | 3.46E+09                     | 9.73E+04                     |
| <i>F</i> <sub>3</sub>  | Mean    | 1.14E-14    | 0.00E+00                     | <b>6.25E+03</b> <sup>+</sup> | <b>9.02E-01</b> <sup>+</sup> | <b>4.18E+02</b> <sup>+</sup> | <b>1.10E+05</b> <sup>+</sup> | <b>2.34E+04</b> <sup>+</sup> |
|                        | Std     | 2.27E-14    | 0.00E+00                     | 1.48E+03                     | 4.31E+00                     | 1.13E+03                     | 5.11E+03                     | 3.80E+03                     |
| <i>F</i> <sub>4</sub>  | Mean    | 1.86E+02    | <b>1.92E+02</b> <sup>+</sup> | 1.40E+02                     | <b>1.04E+04</b> <sup>+</sup> | <b>4.09E+03</b> <sup>+</sup> | <b>1.27E+04</b> <sup>+</sup> | 1.82E+02                     |
|                        | Std     | 2.44E+01    | 2.69E+01                     | 7.29E+00                     | 3.91E+04                     | 1.31E+04                     | 7.87E+02                     | 2.44E+01                     |
| <i>F</i> <sub>5</sub>  | Mean    | 2.13E+01    | 2.13E+01                     | 2.13E+01                     | 2.00E+01                     | 2.00E+01                     | 2.13E+01                     | 2.13E+01                     |
|                        | Std     | 3.64E-02    | 2.56E-02                     | 2.46E-02                     | 1.47E-02                     | 0.00E+00                     | 2.28E-02                     | 3.62E-02                     |
| <i>F</i> <sub>6</sub>  | Mean    | 6.29E-01    | <b>2.43E+00</b> <sup>+</sup> | <b>1.06E+01</b> <sup>+</sup> | <b>1.56E+02</b> <sup>+</sup> | <b>6.96E+01</b> <sup>+</sup> | <b>3.39E+01</b> <sup>+</sup> | <b>5.05E+00</b> <sup>+</sup> |
|                        | Std     | 6.15E-01    | 1.85E+00                     | 3.51E+00                     | 5.33E+00                     | 2.75E+00                     | 3.54E+00                     | 1.58E+00                     |
| <i>F</i> <sub>7</sub>  | Mean    | 0.00E+00    | 0.00E+00                     | <b>1.00E-12</b> <sup>+</sup> | <b>9.27E-03</b> <sup>+</sup> | <b>2.65E-02</b> <sup>+</sup> | <b>1.10E+03</b> <sup>+</sup> | <b>7.16E-02</b> <sup>+</sup> |
|                        | Std     | 0.00E+00    | 0.00E+00                     | 1.84E-13                     | 2.21E-02                     | 2.82E-02                     | 3.23E+01                     | 5.06E-02                     |
| <i>F</i> <sub>8</sub>  | Mean    | 1.22E+01    | <b>2.37E+01</b> <sup>+</sup> | <b>5.79E+02</b> <sup>+</sup> | <b>1.48E+03</b> <sup>+</sup> | <b>4.93E+02</b> <sup>+</sup> | <b>2.75E+02</b> <sup>+</sup> | 1.33E-01                     |
|                        | Std     | 3.02E+00    | 4.46E+00                     | 2.94E+02                     | 3.00E+02                     | 1.31E+02                     | 1.62E+01                     | 3.38E-01                     |
| <i>F</i> <sub>9</sub>  | Mean    | 9.15E+00    | <b>2.08E+01</b> <sup>+</sup> | <b>6.50E+02</b> <sup>+</sup> | <b>1.68E+03</b> <sup>+</sup> | <b>6.27E+02</b> <sup>+</sup> | <b>2.78E+02</b> <sup>+</sup> | 3.26E+00                     |
|                        | Std     | 3.08E+00    | 4.72E+00                     | 2.40E+02                     | 2.62E+02                     | 1.31E+02                     | 1.83E+01                     | 1.46E+00                     |
| <i>F</i> <sub>10</sub> | Mean    | 1.23E+03    | <b>1.79E+03</b> <sup>+</sup> | <b>2.78E+04</b> <sup>+</sup> | <b>2.20E+04</b> <sup>+</sup> | <b>8.58E+03</b> <sup>+</sup> | <b>6.55E+03</b> <sup>+</sup> | 1.17E+02                     |
|                        | Std     | 4.85E+02    | 7.29E+02                     | 7.77E+02                     | 7.13E+03                     | 2.27E+03                     | 7.46E+02                     | 1.16E+02                     |
| <i>F</i> <sub>11</sub> | Mean    | 1.79E+03    | <b>1.85E+03</b> <sup>+</sup> | <b>2.87E+04</b> <sup>+</sup> | <b>2.29E+04</b> <sup>+</sup> | <b>8.34E+03</b> <sup>+</sup> | <b>5.66E+03</b> <sup>+</sup> | 1.23E+03                     |
|                        | Std     | 4.62E+02    | 4.59E+02                     | 9.63E+02                     | 6.21E+03                     | 1.97E+03                     | 4.87E+02                     | 5.13E+02                     |
| <i>F</i> <sub>12</sub> | Mean    | 3.90E+00    | 3.99E+00                     | 3.97E+00                     | 1.19E+00                     | 7.28E-01                     | 3.93E+00                     | 3.96E+00                     |
|                        | Std     | 2.34E-01    | 2.21E-01                     | 2.00E-01                     | 5.49E-01                     | 2.81E-01                     | 3.04E-01                     | 2.10E-01                     |
| <i>F</i> <sub>13</sub> | Mean    | 1.94E-01    | 1.35E-01                     | <b>3.23E-01</b> <sup>+</sup> | <b>2.39E+00</b> <sup>+</sup> | <b>8.72E-01</b> <sup>+</sup> | <b>5.54E+00</b> <sup>+</sup> | 5.85E-02                     |
|                        | Std     | 1.92E-02    | 1.85E-02                     | 2.55E-02                     | 3.94E+00                     | 1.17E+00                     | 7.78E-02                     | 9.94E-03                     |
| <i>F</i> <sub>14</sub> | Mean    | 3.12E-01    | <b>3.86E-01</b> <sup>+</sup> | <b>3.63E-01</b> <sup>+</sup> | <b>2.40E+02</b> <sup>+</sup> | <b>4.26E-01</b> <sup>+</sup> | <b>3.27E+02</b> <sup>+</sup> | <b>4.50E-01</b> <sup>+</sup> |
|                        | Std     | 3.53E-02    | 4.86E-02                     | 1.83E-02                     | 2.63E+02                     | 1.50E-01                     | 7.41E+00                     | 2.05E-02                     |
| <i>F</i> <sub>15</sub> | Mean    | 1.07E+01    | 1.03E+01                     | <b>6.01E+01</b> <sup>+</sup> | <b>2.48E+07</b> <sup>+</sup> | <b>2.48E+05</b> <sup>+</sup> | <b>1.90E+05</b> <sup>+</sup> | <b>1.22E+01</b> <sup>+</sup> |
|                        | Std     | 7.70E-01    | 8.26E-01                     | 1.66E+01                     | 7.66E+07                     | 1.33E+06                     | 3.06E+04                     | 1.07E+00                     |
| <i>F</i> <sub>16</sub> | Mean    | 4.39E+01    | <b>4.59E+01</b> <sup>+</sup> | <b>4.53E+01</b> <sup>+</sup> | 4.64E+01                     | 2.26E+01                     | 4.33E+01                     | 4.26E+01                     |
|                        | Std     | 4.36E-01    | 2.25E-01                     | 3.31E-01                     | 9.07E-01                     | 4.77E-01                     | 4.56E-01                     | 5.66E-01                     |
| <i>F</i> <sub>17</sub> | Mean    | 1.54E+03    | 7.72E+02                     | <b>2.74E+05</b> <sup>+</sup> | <b>1.32E+09</b> <sup>+</sup> | <b>1.19E+08</b> <sup>+</sup> | <b>9.54E+07</b> <sup>+</sup> | <b>3.85E+07</b> <sup>+</sup> |
|                        | Std     | 4.09E+02    | 2.45E+02                     | 5.52E+04                     | 9.08E+08                     | 1.52E+08                     | 1.15E+07                     | 3.48E+06                     |
| <i>F</i> <sub>18</sub> | Mean    | 1.46E+02    | 2.60E+01                     | <b>2.54E+02</b> <sup>+</sup> | <b>1.97E+10</b> <sup>+</sup> | <b>1.41E+09</b> <sup>+</sup> | <b>3.01E+09</b> <sup>+</sup> | <b>1.73E+02</b> <sup>+</sup> |
|                        | Std     | 3.13E+01    | 7.82E+00                     | 1.87E+02                     | 2.58E+10                     | 3.71E+09                     | 4.29E+08                     | 1.43E+02                     |
| <i>F</i> <sub>19</sub> | Mean    | 9.29E+01    | 9.40E+01                     | 9.44E+01                     | <b>2.43E+03</b> <sup>+</sup> | 2.57E+01                     | 5.14E+02                     | 7.21E+01                     |
|                        | Std     | 1.48E+00    | 1.38E+00                     | 1.41E+00                     | 4.59E+03                     | 2.12E+00                     | 6.67E+01                     | 1.71E+01                     |
| <i>F</i> <sub>20</sub> | Mean    | 1.70E+01    | 9.56E+00                     | <b>8.43E+03</b> <sup>+</sup> | <b>2.33E+06</b> <sup>+</sup> | <b>6.75E+04</b> <sup>+</sup> | <b>2.60E+04</b> <sup>+</sup> | <b>1.08E+05</b> <sup>+</sup> |
|                        | Std     | 5.52E+00    | 2.40E+00                     | 2.00E+03                     | 5.76E+06                     | 3.67E+04                     | 2.96E+03                     | 1.89E+04                     |
| <i>F</i> <sub>21</sub> | Mean    | 3.68E+02    | 1.98E+02                     | <b>1.51E+05</b> <sup>+</sup> | <b>8.09E+08</b> <sup>+</sup> | <b>2.02E+07</b> <sup>+</sup> | <b>3.18E+06</b> <sup>+</sup> | <b>2.40E+07</b> <sup>+</sup> |
|                        | Std     | 1.28E+02    | 6.34E+01                     | 3.04E+04                     | 5.30E+08                     | 3.12E+07                     | 1.09E+06                     | 2.95E+06                     |
| <i>F</i> <sub>22</sub> | Mean    | 5.09E+01    | <b>6.56E+01</b> <sup>+</sup> | <b>3.49E+03</b> <sup>+</sup> | <b>2.43E+03</b> <sup>+</sup> | <b>2.43E+03</b> <sup>+</sup> | <b>9.21E+01</b> <sup>+</sup> | <b>2.90E+02</b> <sup>+</sup> |
|                        | Std     | 2.94E+01    | 5.21E+01                     | 2.48E+02                     | 2.59E+05                     | 1.73E+03                     | 4.21E+01                     | 2.09E+02                     |
| <i>F</i> <sub>23</sub> | Mean    | 3.48E+02    | 3.48E+02                     | 3.48E+02                     | <b>6.72E+03</b> <sup>+</sup> | <b>1.44E+03</b> <sup>+</sup> | <b>4.41E+02</b> <sup>+</sup> | <b>3.59E+02</b> <sup>+</sup> |
|                        | Std     | 0.00E+00    | 0.00E+00                     | 0.00E+00                     | 1.60E+03                     | 6.17E+02                     | 1.26E+01                     | 2.91E+00                     |
| <i>F</i> <sub>24</sub> | Mean    | 3.83E+02    | 3.73E+02                     | 3.61E+02                     | <b>5.07E+02</b> <sup>+</sup> | 4.29E+02                     | 3.82E+02                     | 3.82E+02                     |
|                        | Std     | 3.04E+00    | 1.90E+00                     | 4.63E+00                     | 2.74E+01                     | 1.32E+02                     | 1.41E+00                     | 1.59E+00                     |
| <i>F</i> <sub>25</sub> | Mean    | 2.16E+02    | 2.16E+02                     | 2.00E+02                     | <b>1.37E+03</b> <sup>+</sup> | <b>4.20E+02</b> <sup>+</sup> | <b>2.42E+02</b> <sup>+</sup> | <b>2.23E+02</b> <sup>+</sup> |
|                        | Std     | 5.77E-01    | 8.62E-01                     | 0.00E+00                     | 2.04E+02                     | 5.42E+01                     | 3.06E+00                     | 9.36E+00                     |
| <i>F</i> <sub>26</sub> | Mean    | 2.00E+02    | 2.00E+02                     | 2.00E+02                     | <b>8.64E+02</b> <sup>+</sup> | 1.25E+02                     | <b>2.00E+02</b> <sup>+</sup> | 2.00E+02                     |
|                        | Std     | 0.00E+00    | 3.97E-04                     | 7.67E-03                     | 4.77E+02                     | 6.80E+01                     | 2.31E-03                     | 4.88E-02                     |
| <i>F</i> <sub>27</sub> | Mean    | 3.11E+02    | <b>4.96E+02</b> <sup>+</sup> | <b>4.15E+02</b> <sup>+</sup> | <b>5.50E+03</b> <sup>+</sup> | <b>2.43E+03</b> <sup>+</sup> | <b>1.78E+03</b> <sup>+</sup> | <b>3.23E+02</b> <sup>+</sup> |
|                        | Std     | 1.04E+01    | 1.13E+02                     | 5.12E+01                     | 3.00E+02                     | 1.17E+02                     | 1.05E+02                     | 2.48E+01                     |
| <i>F</i> <sub>28</sub> | Mean    | 2.45E+03    | <b>2.74E+03</b> <sup>+</sup> | 2.17E+03                     | <b>2.98E+04</b> <sup>+</sup> | <b>1.05E+04</b> <sup>+</sup> | <b>4.45E+03</b> <sup>+</sup> | 8.14E+02                     |
|                        | Std     | 1.57E+02    | 5.59E+02                     | 5.72E+01                     | 5.35E+03                     | 1.89E+03                     | 1.10E+03                     | 1.85E+01                     |
| <i>F</i> <sub>29</sub> | Mean    | 5.93E+06    | 7.85E+02                     | 1.80E+03                     | <b>5.60E+09</b> <sup>+</sup> | <b>9.62E+08</b> <sup>+</sup> | 1.69E+05                     | 4.54E+02                     |
|                        | Std     | 2.22E+07    | 1.55E+02                     | 4.30E+01                     | 1.23E+09                     | 2.57E+08                     | 1.26E+05                     | 3.73E+02                     |
| <i>F</i> <sub>30</sub> | Mean    | 9.03E+03    | 4.71E+03                     | 8.82E+03                     | <b>3.11E+08</b> <sup>+</sup> | <b>1.38E+07</b> <sup>+</sup> | <b>8.01E+04</b> <sup>+</sup> | 1.88E+03                     |
|                        | Std     | 8.17E+02    | 1.21E+03                     | 6.54E+02                     | 1.03E+08                     | 5.63E+06                     | 2.63E+04                     | 4.86E+02                     |
| w/t/l                  |         | -           | 11/8/11                      | 19/6/5                       | 26/1/3                       | 24/1/5                       | 25/2/3                       | 15/5/10                      |
| Rank                   |         | <b>2.42</b> | 2.75                         | 3.87                         | 6.10                         | 4.57                         | 5.00                         | 3.30                         |

generation for two times with a greedy replacement strategy. That is to say, once the sampled solution by the local search method in the first run is better than *Gbest*, *Gbest* is replaced by the sampled solution and then the updated *Gbest* is used in Equation (4) to execute another run of the local search method.

To summarize, the overall procedure of LLEDA is shown in Algorithm 1, where the novel contribution steps are shaded in grey.

### 3 EXPERIMENTS

This section is to validate the effectiveness of LLEDA by conducting experiments on the widely used CEC 2014 benchmark set. To make comparisons, we select six state-of-the-art and representative EDA variants, namely EDA2 [2], EDA<sub>VERS</sub> [16], EDA/LS [3], EDA/LS-MS [18], BUMDA [19] and TRA-EDA (which is actually the traditional multivariate Gaussian EDA) [5]

In the experiments, we set the dimension size of the CEC 2014 benchmark functions as 100, and set the maximum number of fitness evaluations ( $FES_{max}$ ) as 10000\**D* (where *D* is the dimension

size) for all algorithms. For fair comparisons, except for the population size *NP*, the parameters of each compared algorithm are set as recommended in the associated papers. With respect to the population size, we tune its settings for all algorithms on the 100-*D* CEC 2014 benchmark set. Specifically, after the preliminary experiments, the population sizes of LLEDA, EDA2, EDA<sub>VERS</sub>, EDA/LS, EDA/LS-MS, BUMDA, and TRA-EDA are 2800, 300, 700, 150, 2000, 1500, and 6000, respectively. In addition, for LLEDA, the number of layers *NL* is set as 4 and the learning step *F* is set as 1.8.

In addition, to make fair and comprehensive comparisons, we execute each algorithm independently 30 times on each benchmark function, and then report the mean value and the standard deviation value over the 30 runs to evaluate the optimization performance of each algorithm. Besides, we also perform the Wilcoxon rank-sum test at the significance level of  $\alpha = 0.05$  to compare LLEDA with each compared EDA variant on each benchmark problem. Meanwhile, the Friedman test at the significance level of  $\alpha = 0.05$

is also carried out to obtain the overall optimization performance of each algorithm on the whole 100-*D* CEC 2014 benchmark set.

Table 1 shows the comparison results between LLEDA and the compared EDA variants. The symbols “+”, “-”, and “=” above the mean values mean that LLEDA is significantly better than, significantly worse than and equivalent to the associated compared EDAs on the corresponding functions, which is determined by the Wilcoxon rank-sum test results. “w/t/l” in the second to last row counts the numbers of “+”, “=”, and “-”, respectively. From this table, we obtain the following findings:

- 1) From the Friedman test results, it is observed that LLEDA achieves the lowest rank among all algorithms, and its rank value is much lower than the other algorithms except for EDA2. This proves that LLEDA obtains the best overall performance on the 100-*D* CEC 2014 benchmark set and it is significantly superior to all compared EDAs except for EDA2.
- 2) From the Wilcoxon rank sum test results, it is found that LLEDA is significantly superior to EDA/LS, EDA/LS-MS, and TRA-EDA on at least 24 functions, and shows inferiority to them on at most 5 functions. Compared with EDA<sub>VERS</sub> and BUMDA, LLEDA significantly outperforms them on 19 and 15 functions, respectively. In competition with EDA2, LLEDA achieve highly competitive performance on the 100-*D* CEC 2014 benchmark set.

In summary, the above experiments demonstrate the great superiority of LLEDA to the compared state-of-the-art EDA variants on the 100-*D* CEC 2014 benchmark set. Such superiority of LLEDA mainly benefits from the proposed layered learning method and the cross-generation individual selection strategy. With the layered learning strategy, LLEDA maintains multiple high-quality probability distribution models. In this way, the evolutionary information of the population is fully made use of to sampled diversified individuals. With the help of the cross-generation individual selection scheme, LLEDA selects high-quality but diversified individuals for the parent population. By means of the cohesive cooperation between the two mechanisms, LLEDA expectedly compromises search intensification and diversification well to find high-quality solutions.

## 4 CONCLUSION

In this paper, we have proposed a layered learning estimation of distribution algorithm (LLEDA) to improve the optimization performance of EDA in solving complicated optimization problems. By means of dividing the population into multiple layers and separately estimating the probability distribution model of each layer, LLEDA maintains multiple distribution models to fully utilize the evolutionary information of the whole population. To improve the quality of the estimated distribution models, LLEDA lets the mean vector of each lower layer learn from the one of a randomly selected higher layer. In this way, it is expected that multiple high-quality distribution models can be maintained to sample high-quality and diversified offspring to seek promising solutions. In addition, a cross-generation individual selection strategy and a local search mechanism have further introduced into LLEDA to enhance the search diversity and promote the local exploitation, so that high-quality solutions can be found.

Comparative experiments have been conducted on the 100-*D*

CEC 2014 benchmark set by comparing LLEDA with several state-of-the-art EDA variants. Experimental results have demonstrated the great superiority of LLEDA to the compared EDA variants in solving optimization problems.

## ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62006124 and U20B2061, in part by the Natural Science Foundation of Jiangsu Province under Project BK20200811, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 20KJB520006, and in part by in part by the National Research Foundation of Korea (NRF-2021H1D3A2A01082705).

## REFERENCES

- [1] Yang, Q., Li, Y., Gao, X.-D., Ma, Y.-Y., Lu, Z.-Y., Jeon, S.-W. and Zhang, J. An Adaptive Covariance Scaling Estimation of Distribution Algorithm. *Mathematics*, 9, 24 (2021).
- [2] Liang, Y., Ren, Z., Yao, X., Feng, Z., Chen, A. and Guo, W. Enhancing Gaussian Estimation of Distribution Algorithm by Exploiting Evolution Direction With Archive. *IEEE Transactions on Cybernetics*, 50, 1 (2020), 140-152.
- [3] Zhou, A., Sun, J. and Zhang, Q. An Estimation of Distribution Algorithm With Cheap and Expensive Local Search Methods. *IEEE Transactions on Evolutionary Computation*, 19, 6 (2015), 807-822.
- [4] Yang, Q., Chen, W.-N., Li, Y., Chen, C. P., Xu, X.-M. and Zhang, J. Multimodal Estimation of Distribution Algorithms. *IEEE transactions on cybernetics*, 47, 3 (2016), 636-650.
- [5] Larrañaga, P. and Lozano, J. A. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. *Genetic Algorithms and Evolutionary Computation*, 2 (2001).
- [6] Yang, Q., Chen, W., Deng, J. D., Li, Y., Gu, T. and Zhang, J. A Level-Based Learning Swarm Optimizer for Large-Scale Optimization. *IEEE Transactions on Evolutionary Computation*, 22, 4 (2018), 578-594.
- [7] Yang, Q., Chen, W. N., Gu, T., Jin, H., Mao, W. and Zhang, J. An Adaptive Stochastic Dominant Learning Swarm Optimizer for High-Dimensional Optimization. *IEEE Transactions on Cybernetics*, 50, 7 (2020), 3393-3408.
- [8] Yang, Q., Chen, W. N., Gu, T., Zhang, H., Yuan, H., Kwong, S. and Zhang, J. A Distributed Swarm Optimizer With Adaptive Communication for Large-Scale Optimization. *IEEE Transactions on Cybernetics*, 50, 7 (2020), 3393-3408.
- [9] Ceberio, J., Irurizki, E., Mendiburu, A. and Lozano, J. A. A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence*, 1, 1 (2012), 103-117.
- [10] Krejca, M. S. *Theoretical Analyses of Univariate Estimation-of-distribution Algorithms*. Universität Potsdam, 2019.
- [11] Yang, Q., Chen, W.-N. and Zhang, J. Probabilistic Multimodal Optimization. *Metaheuristics for Finding Multiple Solutions* (2021), 191-228.
- [12] Gahl, J., Bosman, P. A. and Rothlauf, F. The Correlation-Triggered Adaptive Variance Scaling IDEA. *Proceedings of Annual Conference on Genetic and Evolutionary Computation* (2006), 397-404.
- [13] Bosman, P. A., Gahl, J. and Rothlauf, F. SDR: A Better Trigger for Adaptive Variance Scaling in Normal EDAs. *Proceedings of Annual Conference on Genetic and Evolutionary Computation* (2007), 492-499.
- [14] Cai, Y., Sun, X., Xu, H. and Jia, P. Cross Entropy and Adaptive Variance Scaling in Continuous EDA. *Proceedings of Annual Conference on Genetic and Evolutionary Computation* (2007), 609-616.
- [15] Bosman, P. A., Gahl, J. and Thierens, D. Enhancing the Performance of Maximum-likelihood Gaussian EDAs Using Anticipated Mean Shift. *International Conference on Parallel Problem Solving from Nature* (2008), 133-143.
- [16] Ren, Z., He, C., Zhong, D., Huang, S. and Liang, Y. Enhance Continuous Estimation of Distribution Algorithm by Variance Enlargement and Reflecting Sampling. *IEEE Congress on Evolutionary Computation* (2016), 3441-3447.
- [17] Liang, J. J., Qu, B. Y. and Suganthan, P. N. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-parameter Numerical Optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore* (2013).
- [18] Fang, H., Zhou, A. and Zhang, G. An Estimation of Distribution Algorithm Guided by Mean Shift. *IEEE Congress on Evolutionary Computation* (2016), 3268-3275.
- [19] Valdez, S. I., Hernández, A. and Botello, S. A Boltzmann Based Estimation of Distribution Algorithm. *Information Sciences*, 236 (2013), 126-137.