Variable Classification-based Co-estimation of Distribution Algorithm Assisted by a Cooperative Particle Swarm Optimizer

Wen Liu and Xue-feng Yan

ABSTRACT

Since estimation of distribution algorithms (EDAs) were proposed, many methods have been made to improve EDAs' performance. In this paper, a variable classification-based cooperative estimation of distribution algorithm (VC-CEDA) is proposed to balance the global and local searching ability of estimation of distribution algorithms. This study proposes a variable classification method to improve the efficiency and accuracy of probability model development. And a cooperative particle swarm optimizer that can improve the local searching ability is proposed. The VC-CEDA combines the advantages of traditional algorithm and cooperative operations. VC-CEDA shows significantly better performance on 10 test functions. In this study, the experimental results show that the VC-CEDA performs better than traditional EDAs.

INTRODUCTION

Evolutionary and swarm intelligence algorithms are optimization techniques that are inspired by biological evolution [1]. These algorithms are emblematical examples of leading-edge computing technologies and have thus been widely used in engineering science. They are superior to traditional optimization methods when used to solve global optimization and complex problems.

A popular evolutionary algorithm is the estimation of distribution algorithm (EDA). This type of algorithm describes the distribution and population trends in the evolution of the overall solution space by updating the probability model and using learning and sampling methods; EDAs involve probabilistic models instead of the crossover and mutation operations of traditional genetic algorithms [2]. This population evolutionary strategy avoids the randomness and blindness of traditional genetic algorithms. Hence, it can effectively improve the efficiency of evolutionary search. In addition, EDAs suffer from poor local optimization capacity and fail to
solve complex and high-dimensional multi-modal optimization problems, in which they tend to fall into the local optima and cause premature convergence. Premature convergence with EDAs can be caused by the rapid decline in population diversity [3]. Thus, reconstructing population diversity may solve the problem of premature convergence. In particular, the multi-group parallel evolution method is a useful way to maintain population diversity and prevent premature convergence. The specific method is described as follows [4]: the initial population in the search space is divided into multiple sub-groups (which correspond to multiple probability distribution models), each of which evolves independently. Information is exchanged (i.e. migration operation) among different sub-groups at a certain evolutionary generation to generate new population and evolve the next generation on the basis of this cycle until the algorithm termination condition is satisfied. A migration operation is proven to be effective for EDAs and is known to yield good results [5, 6, 7, 8, 9].

The present study proposes a new algorithm involving a variable classification (VC)-based cooperative EDA (VC-CEDA). The proposed algorithm uses a cooperative particle swarm optimizer (CPSO) to improve its local search ability. Population diversity is maintained with a double-layer grouping structure. The first layer is divided into three categories (i.e. uncorrelated, weakly dependent, and strongly dependent variables) with a VC method. The second layer is divided by grouping particles randomly. The random grouping method can ensure the presence of the same particles at different generations in different sub-groups. The effect of this method is similar to a migration operation. Modeling methods are applied to three different categories to build probabilistic models. The global model consists of information on three models. The new EDA population is generated with the three probabilistic models, the formation of which depends on the particles of the particle swarm optimizer (PSO). The VC-CEDA combines the advantages of EDAs and cooperative operations.

The rest of the paper is organized as follows. Section 2 describes the framework of the VC-CEDA in detail. Section 3 reports the performance analysis and experimental results. Section 4 concludes the paper with a summary.

NEW ALGORITHM FRAMEWORK

Variable Classification

The relationship among the variables of a multivariate Gaussian model is expressed by their variance. And $\text{cov}(x_i, x_j)$ is the covariance between variables $x_i$ and $x_j$, $\text{corr}(x_i, x_j)$ is the linear correlation coefficient between variables $x_i$ and $x_j$, $\sigma_i$ and $\sigma_j$ are the standard deviations of $x_i$ and $x_j$, respectively. Assume that there are $n$ variables:

\[
\text{cov}(x_i, x_j) = E((x_i - u_i)(x_j - u_j))
\]

(1)

\[
\text{corr}(x_i, x_j) = \frac{\text{cov}(x_i, x_j)}{\sigma_i \sigma_j}
\]

(2)

If correlation coefficients are close to zero, which indicates a weak dependence among variables [10], then they should be divided into uncorrelated variables. In this study, we regard variables as strongly dependent variables if the correlation coefficients are near one, which indicates a strong dependence among variables.
Therefore, the variables in this work are divided into three categories depending on their correlation matrices: uncorrelated, weakly dependent, and strongly dependent variables. This strategy is called VC method. The sets of the three categories of variables are expressed as $U$, $W$, and $S$, as defined below:

$$
U = \{ x_i | \text{corr}(x_i, x_j) \leq \theta_1, \forall j = 1, \ldots, n, j \neq i \} \\
W = \{ x_i | x_i \not\in U, x_i \neq s, i = 1, \ldots, n \} \\
S = \{ x_i | \text{corr}(x_i, x_j) \geq \theta_2, \forall j = 1, \ldots, n, j \neq i \}
$$

(3)

**Procedure of New Algorithm**

The procedure of the VC-CEDA can be described as follows:

**Step 1: Initialization.** The sizes $N_1$ and $N_2$ of the EDA population and particles, respectively, are determined. The maximum value of generations $G_{\text{max}}$ is defined. The original EDA population and the initial particles are randomly generated. The initial generation is $G = 0$. The specific method is as follows: the randomly generated initial values are between $[0,1]$. $x_{\text{low}}$ is the lower limit of the solution space, $x_{\text{high}}$ is the upper limit of the solution space.

$$
x_{j,i} = x_{\text{low}} + (x_{\text{high}} - x_{\text{low}}) \times \text{rand}(1)
$$

(4)

$$
p_{q,i} = x_{\text{low}} + (x_{\text{high}} - x_{\text{low}}) \times \text{rand}(1)
$$

(5)

Where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, N_1$, $q = 1, 2, \ldots, N_2$, and $n$ is the number of variables.

**Step 2: Evaluation and selection.** The fitness values are evaluated for the EDA population $X(G)$. The population is sorted according to a certain order, and a truncation selection method is used to choose the superior group $\text{Sup}(G)$.

**Step 3: Variable classification.** A covariance matrix is calculated on the basis of superior group $\text{Sup}(G)$. The variables are classified depending on the covariance matrix. The superior group $\text{Sup}(G)$ is then divided into three categories with the VC method. On the basis of the classification results of the superior group $\text{Sup}(G)$, the particles $P(G)$ are correspondingly divided into three categories: $P_U(G)$, $P_W(G)$, and $P_S(G)$. The VC process is shown in Fig. 1.

**Step 4: Variable classification-based particle swarm optimizer.** For particles $P(G)$, all of them have fitness values determined by an objective function. Each particle has a velocity that indicates the direction and distance of its evolution. The particles follow the current optimal particles and the best information on all the particles to identify the optimal solution in the solution space. The velocity $v_{ij}(G)$ and position $x_{ij}(G)$ of the $i$-th particle are updated by the following equations.

$$
v_{y}(G+1) = w \cdot v_{y}(G) + c_1 \cdot r_1 \cdot (p_{y_{\text{best}}}(G) - x_{y}(G)) + c_2 \cdot r_2 \cdot (g_{y_{\text{best}}}(G) - x_{y}(G))
$$

(6)

$$
x_{y}(G+1) = x_{y}(G) + v_{y}(G+1)
$$

(7)

In the VC-PSO, the particles for $P_U(G)$, $P_W(G)$, and $P_S(G)$ are randomly divided into multiple components of the $s$-dimensional vector. The velocity and
position of these particles in the same components are updated synchronously with formulas (6) and (7), respectively. A random re-grouping operation forms the double-layer structure. The framework of the double-layer grouping structure in the VC-PSO is shown in Fig. 2.

![Framework of the double-layer grouping.](image)

Step 5: Modelling. The UMDAc model for particles $P(G)$ is applied for three different variables. The probability model of the entire search space combines three models of three types of variables. The mean is a combination of three vectors. The covariance matrix includes three covariance matrices in a diagonal position, and the remaining elements of the covariance matrix are zero.

The UMDAc is an EDA for cases involving independent variables. Its joint density function follows a normal distribution. Given a set of samples, we can use the maximum likelihood estimation method to estimate the distribution of the mean $\mu_i$ and variance $\sigma_i^2$.

$$\hat{\mu}_i = \frac{1}{N} \sum_{k=1}^{N} x_{ik}$$

$$\sigma_i^2 = \frac{1}{N} \sum_{k=1}^{N} (x_{ik} - \hat{\mu}_i)^2$$

Step 6: EDA population sampling. New EDA individuals are generated for the next generation according to the global model.

Step 7: Termination. If $G < G_{\text{max}}$, go to step 2.

**EXPERIMENTS AND RESULTS**

Ten test functions were adopted to test the performance of the VC-CEDA. These test functions were minimization problems, as summarized in Table 1. The VC-CEDA was compared with a traditional EDA using the UMDAc model. The mean, variance, and minimum and maximum values of the algorithms, all of which are searchable, are identified.

<table>
<thead>
<tr>
<th>Table 1. Test functions.</th>
<th></th>
</tr>
</thead>
</table>
Expression | Bounds
--- | ---
\( f_1(x) = \sum_{i=1}^{n} x_i^2 \) | [-5.12, 5.12]
\( f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \) | [-10, 10]
\( f_3(x) = (\sum_{i=1}^{n} (i+1) x_i^2) + \text{rand}(0,1) \) | [-1.28, 1.28]
\( f_4(x) = \sum_{i=1}^{n} \left( x_i - 10 \cos(2\pi x_i) + 10 \right) \) | [-5.12, 5.12]
\( f_5(x) = \left( \frac{1}{4000} \sum_{i=1}^{n} x_i^2 \right) - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i+1}} \right) + 1 \) | [-600, 600]
\( f_6(x) = \max \{|x_i|, 0 \leq i < n\} \) | [-100, 100]
\( f_7(x) = \sum_{i=1}^{n} x_i \sin \left( \sqrt{|x_i|} \right) \) | [-500, 500]
\( f_8(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos \left( 2\pi x_i \right) \right) + 20 + e \) | [-32, 32]
\( f_9(x) = \frac{1}{n} \sum_{i=1}^{n} \left( x_i - 16x_i^3 + 5x_i \right) \) | [-5.5]
\( f_{10}(x) = \sum_{i=1}^{n} \left( x_i - x_{\text{ref}}^i \right)^2 + (x_i - 1)^2 \) | [-10, 10]

### Analysis of Classification Parameters \( \theta_1, \theta_2 \)

The variables were classified into a group (U, W, and S) using the classification parameters \( \theta_1, \theta_2 \) and the correlation matrix, which was calculated at the current generation. A correlation matrix reflects the relationship among variables in the search space. The values of \( \theta_1, \theta_2 \) reflect the dependency level of a probability model. If \( \theta_1 \) is large, many variables are classified in group U. If \( \theta_2 \) is small, many variables are classified in group S. Conversely, a small \( \theta_1 \) can cause group U to be easily empty, and a large \( \theta_2 \) can cause group S to be easily empty.

#### Table 2. Test on classification parameters for 100-D functions.

<table>
<thead>
<tr>
<th>( \theta_1 )</th>
<th>0.3</th>
<th>0.35</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_2 )</td>
<td>0.4</td>
<td>0.45</td>
<td>0.5</td>
</tr>
<tr>
<td>U</td>
<td>21</td>
<td>19</td>
<td>23</td>
</tr>
<tr>
<td>W</td>
<td>60</td>
<td>75</td>
<td>73</td>
</tr>
<tr>
<td>S</td>
<td>19</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

#### Table 3. Test on classification parameters for 500-D functions.

<table>
<thead>
<tr>
<th>( \theta_1 )</th>
<th>0.3</th>
<th>0.35</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_2 )</td>
<td>0.4</td>
<td>0.45</td>
<td>0.5</td>
</tr>
<tr>
<td>U</td>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>W</td>
<td>235</td>
<td>405</td>
<td>476</td>
</tr>
<tr>
<td>S</td>
<td>261</td>
<td>88</td>
<td>18</td>
</tr>
</tbody>
</table>

Considering these factors, we chose eight different combinations of the two parameters to balance the number of variables among the three groups; that is, \( \theta_1 = \{0.3, 0.35, 0.4\} \) and \( \theta_2 = \{0.4, 0.45, 0.5\} \). Table 2 shows that for the 100-D functions, the distribution of the variables in which group is relatively equilibrium as long as \( \theta_1 = 0.3 \) and \( \theta_2 = 0.4 \). For the 500-D functions, the best classification parameters are \( \theta_1 = 0.4 \) and \( \theta_2 = 0.45 \). On the basis of these results, we can conclude that classification parameters for different dimensional problems have an important influence on VC results.
Comparison of the VC-CEDA with a Traditional EDA

Place the figure as close as possible after the point where it is first referenced in the text. If there is a large number of figures and tables it might be necessary to place some before.

The VC-CEDA was compared with a traditional EDA in terms of 100-D problems. The parameters were set as follows. For the EDA, $G_{\text{max}} = 650$, $N = 200$, and the evaluation time is $650 \times 200 = 130,000$. For the VC-CEDA, $G_{\text{max}} = 500$, $N_1 = 200$, $N_2 = 0.3 \times N_1$, and the evaluation time is $500 \times (200 + 60) = 130,000$. The results were averaged over 30 independent runs.

Table 4 shows the comparison results. The EDA demonstrated a better performance than the VC-CEDA for F1, F5, F7, and F8 because the EDA involves only one model, whereas the VC-CEDA comprises three sub-models. Hence, the VC-CEDA cannot describe the relationship among the three models in detail. Nevertheless, step 4 of the VC-CEDA improved its performance for some functions. As shown in Table 4, the VC-CEDA achieved a better performance than the EDA for F2, F3, F4, F6, F9 and F10 because the traditional EDA uses a normal distribution for complex optimization problems and the Gauss distribution model cannot effectively describe the distribution of the solution space. The performance of the VC-CEDA also exhibited improvement because of the two-layer grouping structure and cooperation operation in Step 4. In sum, the performance of the VC-CEDA for 100-D problems is better than that of EDAs.

The VC-CEDA was also applied to 500-D functions and was then compared with the EDA. The evaluation times of both algorithms were set to 130,000. The results were averaged over 30 independent runs, as shown in Table 5. The VC-CEDA performed better than the EDA on F1, F2, F3, F4, F5, F6, F8, F9, and F10. Moreover, the EDA performed better than the VC-CEDA on F7. The traditional EDA is better than VC-CEDA for 500-D for F7 has same justification from better performance of EDA than VC-CEDA in 100-D tests. In sum, the average performance of the VC-CEDA for 500-D functions is better than that of EDAs. As indicated in the differences between Tables 4 and 5, the VC-CEDA is more effective than EDAs when used to deal with high-dimensional problem.

### Table 4. Compare VC-CEDA with EDA on 100-D functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$F_4$</th>
<th>$F_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDA</td>
<td>Mean</td>
<td>8.6430e-12</td>
<td>5.5615e-20</td>
<td>0.0366</td>
<td>26.7347</td>
</tr>
<tr>
<td></td>
<td>Cov</td>
<td>2.2343e-21</td>
<td>1.3448e-41</td>
<td>1.1622e-04</td>
<td>26.7347</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>3.7775e-43</td>
<td>3.3454e-20</td>
<td>0.0213</td>
<td>19.8995</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>1.4598e-08</td>
<td>8.7455e-20</td>
<td>0.0595</td>
<td>37.8336</td>
</tr>
<tr>
<td>VC-CEDA</td>
<td>Mean</td>
<td>2.8351e-09</td>
<td>3.0402e-32</td>
<td>8.5271e-04</td>
<td>3.1662e-06</td>
</tr>
<tr>
<td></td>
<td>Cov</td>
<td>1.9835e-16</td>
<td>3.9004e-65</td>
<td>8.4006e-08</td>
<td>2.8137e-10</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>2.5436e-34</td>
<td>1.9003e-32</td>
<td>4.2466e-04</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>6.8220e-08</td>
<td>4.0685e-32</td>
<td>0.0015</td>
<td>9.1931e-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>$F_6$</th>
<th>$F_7$</th>
<th>$F_8$</th>
<th>$F_9$</th>
<th>$F_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDA</td>
<td>Mean</td>
<td>23.0853</td>
<td>-</td>
<td>22.184e+04</td>
<td>7.0121e-08</td>
</tr>
<tr>
<td></td>
<td>Cov</td>
<td>23.8711</td>
<td>1.2341e+03</td>
<td>2.2018e-04</td>
<td>0.8372</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>15.2518</td>
<td>-</td>
<td>2.4472e+04</td>
<td>7.9936e-15</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>35.0303</td>
<td>-</td>
<td>1.8705e+04</td>
<td>0.0019</td>
</tr>
<tr>
<td>VC-CEDA</td>
<td>Mean</td>
<td>0.8129</td>
<td>-</td>
<td>6.7184e+03</td>
<td>9.3936e-05</td>
</tr>
<tr>
<td></td>
<td>Cov</td>
<td>0.0172</td>
<td>2.1514e-05</td>
<td>1.0833e-07</td>
<td>9.4538e-09</td>
</tr>
<tr>
<td></td>
<td>EDA</td>
<td>VC-CEDA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>0.0278</td>
<td>3.4580e-04</td>
<td>0.0121</td>
<td>2.6414</td>
</tr>
<tr>
<td></td>
<td>Cov</td>
<td>0.0401</td>
<td>0.0014</td>
<td>0.0012</td>
<td>9.6705e-08</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>0.0046</td>
<td>0.4693</td>
<td>0.0012</td>
<td>4.0000e-05</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.8693</td>
<td>3.4092</td>
<td>3.4092</td>
<td>0.8693</td>
</tr>
</tbody>
</table>

**Table 5.** Compare VC-CEDA with EDA on 500-D functions.

**CONCLUSIONS**

In this paper, the proposed VC-CEDA represents a new hybrid mechanism for EDAs and cooperative optimization and features several advantages over traditional EDAs. In the VC-CEDA, a cooperative operation is applied to obtain a specific evolutionary direction for the whole evolution process. This operation can improve evolutionary efficiency, and individuals can obtain enhanced solutions. The experimental results for 10 test functions obtained with the VC-CEDA were compared with those obtained with a traditional EDA. The VC-CEDA provides three sub-models which can put related variables in the same sub-model to describe the search space, and a VC-based cooperative particle swarm optimizer that improves the local searching ability. Thus, the VC-CEDA has better average performance. For future work, how to reduce the computational cost needs further study. Besides, making control parameters dynamically adjustable and letting each parameter have its own self-adaptive strategy to control itself in the proposed algorithm may be beneficial.

**ACKNOWLEDGEMENTS**

The authors gratefully acknowledge the support from the following foundations: 973 project of China (2013CB733605), National Natural Science Foundation of China (21176073) and the Fundamental Research Funds for the Central Universities.
REFERENCES