A Cultivated Differential Evolution Algorithm using modified Mutation and Selection Strategy

Pooja *1 Praveena Chaturvedi1 Pravesh Kumar2
1. Department of Computer Science, Gurukula Kangri Vishwavidyalaya, Haridwar, India
2. Department of Mathematics, AMITY University, Gurgaon, India
* E-mail: mcapooja.singh2007@gmail.com

Abstract
In the present study a modified new variant of Differential Evolution (DE) is proposed, named Cultivated Differential Evolution (CuDE). This algorithm is different from basic DE in two ways. Firstly, the selection of the base vector for mutation operation is not totally randomized while in basic DE uniformly generated random numbers serve this task. Secondly, information preservation concept is used to generate population for the next generation. The performance of the proposed algorithm is validated on a bed of eight benchmark problems taken from literature and compared against the basic DE and some other variants of DE. The numerical results show that the proposed algorithm helps in formulating a better trade-off between convergence rate and efficiency.

Keywords: Differential Evolution, Mutation, Selection, Information Preservation, Population Segmentation

1. Introduction
Differential Evolution (DE) is a variant of Evolutionary Algorithm (EA) which was proposed by Storn and Price in 1997 (Storn & Price 1997). DE is used for solving global optimization problems over continuous spaces. It is a simple and efficient search engine which can handle nonlinear, non-differentiable and multimodal objective functions and a wide range of real life problems such as engineering design (Plagianakos et al. 2008), chemical engineering (Wang et al. 2007) and pattern recognition (Ilonen et al. 2003) and so on (Ali et al. 2013).

For enhancing the performance of DE, Several variants of the same are available in the literature. Some of which are: Modified DE (MDE) (Babu et al. 2006), DE with global and local neighborhood (DEGL) (Das et al. 2009), Cauchy mutation DE (CDE) (Ali & Pant 2010), Opposition based DE(ODE) (Rahnamayan et al. 2008), DE with Trigonometric Mutation (TDE) (Fan & Lampinen 2003), Self adaptive DE (SaDE) (Qin et al. 2009), Learning enhance DE (LeDE) (Cai et al. 2011), Fuzzy adaptive DE (FADE) (Liu & Lampinen 2005), DE with self-adaptive control parameter (JDE) (Brest et al. 2006), DE with random localization (DERL) (Kaelo & Ali 2006), Mixed mutation strategy based DE (Pant et al. 2009), DE with simplex crossover local search (DEahcSPX) (Noman & Iba 2008), adaptive DE with optional external archive (JADE) (Zhang & Sanderson 2009), and so on.

In the present study a different selection strategy named Reserve Selection is used for deciding the area from which the vectors for the mutation are to be chosen. This selection mechanism is based on the technique called ‘population segmentation’ a process of splitting into a Non-reserved Area and a Reserved area (Chen et al. 2007). The reserved area consists of best fit individuals (elite candidates) and non-reserved area maintains the rest of the population. The main operator of DE is mutation, which takes the solution vectors towards a global optimum. In the proposed algorithm, base vector is selected from the reserve area and the difference vectors are selected from the non-reserved area as to focus on the exploitation of the search space which then takes part in the mutation operation. Next the selection of the population for the next generation is formulated by using the information preservation concept taken from literature (Kumar et al. 2011).

The rest of the paper is organized as follows: Section 2 provides a compact overview of DE. Section 3 presents the proposed CuDE algorithm with flow chart. Benchmark problems and experimental settings are given in Section 4. Results and comparisons are reported in Section 5 and finally the conclusion derived from the present study is drawn in Section 6.

2. Basic Differential Evolution Algorithm
Differential Evolution (DE) is proposed by Storn and Price (Storn & Price 1997) is simple, fast and robust evolutionary algorithm. A brief introduction of the basic DE is given as follows:

DE starts with a population of NP solutions: \( X_{i,g}, i = 1, ..., NP \), where the index \( i \) denotes the \( i^{th} \) candidate
solution of the population and G denotes the generation to which the population belongs. The three main operators of DE are mutation, crossover and selection.

- **Mutation**: The mutation operation of DE applies the vector difference between the existing population members in determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three solutions \( \{X_{r_1}, X_{r_2}, X_{r_3}\} \) in the population set of (say) \( NP \) elements. The \( i^{th} \) perturbed individual, \( V_{i,G} \), is generated based on the three chosen solutions, as follows:

\[
V_{i,G} = X_{r_{i,G}} + F*(X_{r_{i,G}} - X_{r_{j,G}})
\]

Where, \( i = 1, ..., NP \), \( r_1, r_2, r_3 \in \{1, ..., NP\} \) are randomly selected such that \( r_1 \neq r_2 \neq r_3 \neq i \), and \( F \) is the control parameter such that \( F \in [0,1] \).

- **Crossover**: Once the perturbed individual \( V_{i,G} = (v_{1,i,G}, v_{2,i,G}, ..., v_{n,i,G}) \) is generated, it is subjected to a crossover operation with target individual \( X_{i,G} = (x_{1,i,G}, x_{2,i,G}, ..., x_{n,i,G}) \), that finally generates the trial solution, \( U_{i,G} = (u_{1,i,G}, u_{2,i,G}, ..., u_{n,i,G}) \), as follows:

\[
U_{j,i,G} = \begin{cases} 
  v_{j,i,G} & \text{if } \text{rand} \leq C_r \text{ or } j = jj \\
  x_{j,i,G} & \text{otherwise}
\end{cases}
\]

Where, \( j = 1, ..., n \), \( jj \in \{1, ..., n\} \) is a random parameter’s index, chosen once for each \( i \). The crossover rate \( C_r \in [0,1] \) is set by the user.

- **Selection**: The selection scheme of DE also differs from that of other EAs. The population for the next generation is selected from the solution in current population and its corresponding trial solution according to the following rule:

\[
X_{i,G+1} = \begin{cases} 
  U_{i,G} & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\
  X_{i,G} & \text{otherwise}
\end{cases}
\]

Thus, each solution of the temporary (trial) population is compared with its counterpart in the current population. The one with the lower objective function value will survive from the tournament selection to the population of the next generation. As a result, all the solutions for the next generation are as good as or better than their counterparts in the current generation. In DE, trial solution is not compared against all the solutions in the current generation, but only against one solution, its counterpart, in the current generation.

### 3. Proposed Algorithm

In this section, the modified operators of the basic algorithm are described. The proposed algorithm uses different mutation and selection operation from that of basic DE as for maximum exploration of the search space.

#### 3.1 Mutation

Three different vectors are chosen from the whole population for performing the basic mutation operation, one of which is to be perturbed is called base vector (donor vector) and the rest two are known as difference vectors. The convergence speed of basic DE highly depends on the selection of the base vector (Kaelo & Ali 2006) and if the base vector is fitter than the difference vectors, the convergence speed of the DE will be better. The nature of
the base vector has a direct impact on the newly generated mutant vector. The Proposed Strategy uses the same concept.

A new mutation scheme is proposed in the present study, in which sorting of the population is performed by taking the fitness of the individuals into consideration. Then the population is divided into two areas: one where the elite individuals are kept and the other one where the rest of the population resides. The proposed CuDE algorithm uses a Reserve Area (RA) maintaining the elite individuals which serves the base vector having size \( NP \times m \%), where \( m \) is an user defined integer and the remaining population Non-Reserved Area (NRA) having size \( NP - NP \times m \%), serves other two difference vectors which are uniformly randomly generated vectors from that population.

3.2 Selection

The CuDE algorithm uses information preservation concept of IPDE (Kumar et al. 2011) in which population \( NP \) of target vectors and population \( NP \) of trial vectors are combined. Now the size of the whole population becomes \( 2 \times NP \). Then sort the whole population and take the best fit \( NP \) individuals for the next generation. This strategy escapes us from loss of potential information.

3.3 Flowchart of the proposed CuDE algorithm

The flow chart of the proposed CuDE algorithm, explaining its working, is shown in Figure 1.

4. Benchmark Problems and Experimental Set-up

We have validated the proposed algorithm on 8 traditional benchmark problems (Zhang & Sanderson 2009). These test problems are given below in Table 1 with the parameters, population \( NP \), Dimension \( D \), \( F \), \( C \), \( m \), Value to reach \( VTR \), maximum Number of Function Evaluation NFE.

The algorithm is compiled in Dev C++ and is executed on Intel Core i3 PC with 4 GB RAM, 50 times for each test problem. The inbuilt \( Rand() \) function of C++ is used to generate the uniformly distributed random numbers. In every case, a run is terminated when NFE reaches the threshold value of maximum NFE.

Performance Criteria: To evaluate the performance of the algorithms, the performance criteria are selected from the literature (Rahnamayan et al. 2008, Tang et al. 2008). These criteria are:

- \( NFEs \): The NFE is recorded when the VTR is reached before to reach maximum NFE i.e. we set the termination criteria as \( |f_{opt} - f_{global}| \leq VTR \) and record average NFE of a successful run over 50 runs.
- \( Error \): The average error \( |f_{opt} - f_{global}| \) is recorded by using predefined maximum NFEs, in each run. Also the average and standard deviation of the fitness values are calculated.
- \( Convergence Graph \): The convergence graphs show the mean fitness performance of the total runs, in the respective experiments.
- \( Acceleration rate (AR) in \% \): The acceleration rate is used to compare the convergence speeds between CuDE and other algorithms (Rahnamayan et al. 2008, Kumar et al. 2011). It is defined as follows:

\[
AR = \frac{\sum NFE_I - \sum NFE_{II}}{\sum NFE_I} \% 
\]

Where \( I \) and \( II \) are two different algorithms.

5. Numerical Results and Comparisons

In Table 2, Comparison of CuDE is driven against basic DE, IPDE and DERL algorithms. As discussed above, the prime requirement for the proposed CuDE algorithm is the selection of RA, i.e. the value of \( m \) should be chosen neither very small nor very large. Now for testing purpose we have taken two cases for which the value of \( m \) is 20% and 30% respectively of the whole population. For both the cases CuDE is giving better results than basic DE, IPDE and DERL. But it is clear from the Table 2 that for \( m=20\% \), it is producing much optimized results.

The comparison is shown in terms of average NFEs, acceleration rate, mean fitness and standard deviation in Table 2, 3 and 4 respectively.
We can see from Table 2, with CuDE (m=20) each benchmark problem takes lesser NFEs to reach the VTR than basic DE, IPDE and DERL and total NFEs for each algorithm is shown in Table 2 with which the acceleration rate of CuDE (m=20) and CuDE (m=30) with respect to basic DE, IPDE and DERL are calculated and shown in Table 3.

In Table 4, Comparison in terms of average error and standard deviation of 50 runs is derived and from the table it is quite clear that CuDE (m=20) produces much better results than the other algorithms taken for comparison.

In Figure 2, convergence graphs between average error and NFE are shown for function $F_1$, $F_2$, $F_5$, $F_6$, $F_7$ respectively.

6. Conclusion

The objective of the present study is to use a modified selection strategy for choosing the individuals for mutation operation and to use the information preservation concept from IPDE for evaluating the new population for the next generation. In this study, the search space is divided into two areas (maximum exploration of the search space): first having the best fit individuals from which the donar vector is selected, second having low fit individuals from which the difference vectors are selected. The proposed CuDE algorithm is validated on a set of 8 standard benchmark problems and for m=20, CuDE produces best results. Numerical results derive that the proposed algorithm performs better than basic DE and some other variants of it.

References


Figure 1. Flowchart of CuDE algorithm

Table 1. Test problems with their Parameter Settings
Table 2. Experimental Results and Comparison of CuDE (m=20, 30) in terms of Average NFE of 50 runs for Standard Test Problems

<table>
<thead>
<tr>
<th>Fun.</th>
<th>DE</th>
<th>IPDE</th>
<th>DERL</th>
<th>m=20</th>
<th>m=30</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_1</td>
<td>116000</td>
<td>80030</td>
<td>67400</td>
<td>32930</td>
<td>42700</td>
</tr>
<tr>
<td>F_2</td>
<td>162410</td>
<td>123510</td>
<td>86410</td>
<td>51460</td>
<td>67825</td>
</tr>
<tr>
<td>F_3</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>F_4</td>
<td>437970</td>
<td>204000</td>
<td>262000</td>
<td>126400</td>
<td>153470</td>
</tr>
<tr>
<td>F_5</td>
<td>117030</td>
<td>117170</td>
<td>76480</td>
<td>42700</td>
<td>54175</td>
</tr>
<tr>
<td>F_6</td>
<td>412040</td>
<td>287180</td>
<td>211870</td>
<td>131390</td>
<td>156250</td>
</tr>
<tr>
<td>F_7</td>
<td>174030</td>
<td>131860</td>
<td>96300</td>
<td>53670</td>
<td>72550</td>
</tr>
<tr>
<td>F_8</td>
<td>106090</td>
<td>82970</td>
<td>59800</td>
<td>35320</td>
<td>44050</td>
</tr>
<tr>
<td>Σ</td>
<td>1525570</td>
<td>1026720</td>
<td>860260</td>
<td>473870</td>
<td>591020</td>
</tr>
</tbody>
</table>

Table 3. Acceleration Rate of CuDE with respect to DE, IPDE and DERL

<table>
<thead>
<tr>
<th>w.r.t.</th>
<th>m=20</th>
<th>m=30</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>68.94%</td>
<td>61.26%</td>
</tr>
<tr>
<td>IPDE</td>
<td>53.85%</td>
<td>42.44%</td>
</tr>
<tr>
<td>DERL</td>
<td>44.92%</td>
<td>31.30%</td>
</tr>
</tbody>
</table>

Table 4. Average error and Standard Deviation* for 50 runs when NFE is fixed

<table>
<thead>
<tr>
<th>Fun.</th>
<th>DE</th>
<th>IPDE</th>
<th>DERL</th>
<th>m=20</th>
<th>m=30</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_1</td>
<td>3.56e-14</td>
<td>7.09e-20</td>
<td>5.04e-30</td>
<td>7.26e-54</td>
<td>2.87e-40</td>
</tr>
<tr>
<td>F_2</td>
<td>1.83e-14</td>
<td>4.80e-20</td>
<td>7.75e-31</td>
<td>2.71e-54</td>
<td>1.82e-40</td>
</tr>
<tr>
<td>F_3</td>
<td>6.96e-08</td>
<td>5.64e-11</td>
<td>7.55e-15</td>
<td>3.99e-15</td>
<td>4.89e-15</td>
</tr>
<tr>
<td>F_4</td>
<td>3.03e-08</td>
<td>1.66e-11</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>1.54e-15</td>
</tr>
<tr>
<td>F_5</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>F_6</td>
<td>2.04e-10</td>
<td>2.13e-19</td>
<td>1.73e-29</td>
<td>5.44e-45</td>
<td>3.99e-30</td>
</tr>
<tr>
<td>F_7</td>
<td>3.39e-10</td>
<td>2.99e-19</td>
<td>3.00e-29</td>
<td>1.91e-45</td>
<td>1.42e-30</td>
</tr>
<tr>
<td>F_8</td>
<td>4.77e-03</td>
<td>3.81e-03</td>
<td>1.88e-03</td>
<td>1.62e-03</td>
<td>1.57e-03</td>
</tr>
<tr>
<td>F_10</td>
<td>2.37e-11</td>
<td>1.76e-17</td>
<td>1.44e-24</td>
<td>1.56e-38</td>
<td>5.99e-35</td>
</tr>
<tr>
<td>F_11</td>
<td>2.04e-11</td>
<td>1.74e-17</td>
<td>1.91e-24</td>
<td>1.56e-38</td>
<td>4.29e-35</td>
</tr>
<tr>
<td>F_12</td>
<td>3.65e-10</td>
<td>6.57e-14</td>
<td>2.50e-20</td>
<td>1.25e-36</td>
<td>2.59e-27</td>
</tr>
<tr>
<td>F_13</td>
<td>1.27e-10</td>
<td>4.88e-14</td>
<td>1.07e-20</td>
<td>5.88e-37</td>
<td>1.18e-27</td>
</tr>
<tr>
<td>F_14</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
</tr>
<tr>
<td>F_15</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
<td>0.0e+00</td>
</tr>
</tbody>
</table>
Figure 2. Convergence graph between Average Error and NFE for (a) Sphere, (b) Ackley, (c) Noise, (d) Schwefel 1.2, (e) Schwefel 2.22 respectively
The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage: 
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar