Experimental Study on Recent Advances in Differential Evolution Algorithm

G. Jeyakumar, Amrita School of Engineering, India
C. Shanmugavelayutham, Amrita School of Engineering, India

ABSTRACT

The Differential Evolution (DE) is a well known Evolutionary Algorithm (EA), and is popular for its simplicity. Several novelties have been proposed in research to enhance the performance of DE. This paper focuses on demonstrating the performance enhancement of DE by implementing some of the recent ideas in DE’s research viz. Dynamic Differential Evolution (dDE), Multiple Trial Vector Differential Evolution (mtvDE), Mixed Variant Differential Evolution (mvDE), Best Trial Vector Differential Evolution (btvDE), Distributed Differential Evolution (diDE) and their combinations. The authors have chosen fourteen variants of DE and six benchmark functions with different modality viz. Unimodal Separable, Unimodal Nonseparable, Multimodal Separable, and Multimodal Nonseparable. On analyzing distributed DE and mixed variant DE, a novel mixed-variant distributed DE is proposed whereby the subpopulations (islands) employ different DE variants to cooperatively solve the given problem. The competitive performance of mixed-variant distributed DE on the chosen problem is also demonstrated. The variants are well compared by their mean objective function values and probability of convergence.

Keywords: Base Vector Selection, Differential Evolution, Distributed Differential Evolution, Dynamic Differential Evolution, Mixed Perturbation, Multiple Trail Vectors

INTRODUCTION

Differential Evolution (DE), proposed by Storn and Price (1995, 1999), is a simple yet powerful evolutionary algorithm (EA) for global optimization in the continuous search domain (Price, 1999). DE has shown superior performance in both widely used benchmark functions and real-world problems (Price et al., 2005; Vesterstrom & Thomsen, 2004). Like other EAs, DE is a population-based stochastic global optimizer employing mutation, recombination and selection operators and is capable of solving reliably nonlinear and multimodal problems. However, it has some unique characteristics that make it different from other members of the EA family. DE uses a differential mutation operation based on the distribution of parent solutions in the current population, coupled with recombination with a predetermined parent to generate a trial vector (offspring) followed by a one-to-one greedy selection scheme between the trial vector

DOI: 10.4018/jaec.2011040103
and the parent. The algorithmic description of a classical DE is depicted in Figure 1.

Depending on the way the parent solutions are perturbed to generate a trial vector, there exist many trial vector generation strategies and consequently many DE variants. With seven commonly used differential mutation strategies (Montes et al., 2006), as listed in Table 1, and two crossover schemes (binomial and exponential), we get fourteen possible variants of DE viz. rand/1/bin, rand/1/exp, best/1/bin, best/1/exp, rand/2/bin, rand/2/exp, best/2/bin, best/2/exp.

Table 1. Differential mutation strategies

<table>
<thead>
<tr>
<th>Nomenclature</th>
<th>Variant</th>
</tr>
</thead>
<tbody>
<tr>
<td>rand/1</td>
<td>$V_{i,e} = \frac{X_{i,e}}{r_{2,e}^{r}} + F\left(\frac{X_{i,e}}{r_{2,e}^{r}} - \frac{X_{1,e}}{r_{2,e}^{r}}\right)$</td>
</tr>
<tr>
<td>best/1</td>
<td>$V_{i,e} = \frac{X_{\text{best},e}}{r_{2,e}^{r}} + F\left(\frac{X_{i,e}}{r_{2,e}^{r}} - \frac{X_{\text{best},e}}{r_{2,e}^{r}}\right)$</td>
</tr>
<tr>
<td>rand/2</td>
<td>$V_{i,e} = \frac{X_{i,e}}{r_{2,e}^{r}} + F\left(\frac{X_{i,e}}{r_{2,e}^{r}} - \frac{X_{1,e}}{r_{2,e}^{r}} + \frac{X_{1,e}}{r_{4,e}^{r}} - \frac{X_{i,e}}{r_{4,e}^{r}}\right)$</td>
</tr>
<tr>
<td>best/2</td>
<td>$V_{i,e} = \frac{X_{\text{best},e}}{r_{2,e}^{r}} + F\left(\frac{X_{i,e}}{r_{2,e}^{r}} - \frac{X_{\text{best},e}}{r_{2,e}^{r}} + \frac{X_{1,e}}{r_{4,e}^{r}} - \frac{X_{i,e}}{r_{4,e}^{r}}\right)$</td>
</tr>
<tr>
<td>current-to-rand/1</td>
<td>$V_{i,e} = \frac{X_{i,e}}{r_{2,e}^{r}} + K\left(\frac{X_{1,e}}{r_{2,e}^{r}} - \frac{X_{i,e}}{r_{2,e}^{r}}\right) + F\left(\frac{X_{1,e}}{r_{2,e}^{r}} - \frac{X_{1,e}}{r_{4,e}^{r}}\right)$</td>
</tr>
<tr>
<td>current-to-best/1</td>
<td>$V_{i,e} = \frac{X_{i,e}}{r_{2,e}^{r}} + K\left(\frac{X_{\text{best},e}}{r_{2,e}^{r}} - \frac{X_{i,e}}{r_{2,e}^{r}}\right) + F\left(\frac{X_{1,e}}{r_{2,e}^{r}} - \frac{X_{1,e}}{r_{4,e}^{r}}\right)$</td>
</tr>
<tr>
<td>rand-to-best/1</td>
<td>$V_{i,e} = \frac{X_{i,e}}{r_{2,e}^{r}} + K\left(\frac{X_{\text{best},e}}{r_{2,e}^{r}} - \frac{X_{i,e}}{r_{2,e}^{r}}\right) + F\left(\frac{X_{1,e}}{r_{2,e}^{r}} - \frac{X_{1,e}}{r_{4,e}^{r}}\right)$</td>
</tr>
</tbody>
</table>

Figure 1. General structure of DE algorithm

```
Population Initialization X(0) ← {x_i(0),...,x_N(0)}
g ← 0
Compute {f(x_i(g)),...,f(x_N(g))}
while the stopping condition is false do
    for i = 1 to NP do
        y_i ← generateMutant(X(g))
        z_i ← crossover(x_i(g), y_i)
        if f(z_i) < f(x_i(g)) then
            x_i(g+1) ← z_i
        else
            x_i(g+1) ← x_i(g)
        end if
    end for
    g ← g+1
    Compute {f(x_1(g)),...,f(x_N(g))}
end while
```
Link Prediction Evaluation Using Palette Weisfeiler-Lehman Graph Labelling Algorithm
[www.igi-global.com/article/link-prediction-evaluation-using-palette-weisfeiler-lehman-graph-labelling-algorithm/233680?camid=4v1a](www.igi-global.com/article/link-prediction-evaluation-using-palette-weisfeiler-lehman-graph-labelling-algorithm/233680?camid=4v1a)

Prognostics and Health Management of Industrial Equipment
[www.igi-global.com/chapter/prognostics-health-management-industrial-equipment/69686?camid=4v1a](www.igi-global.com/chapter/prognostics-health-management-industrial-equipment/69686?camid=4v1a)