Differential Evolution: Foundations, Perspectives, and Applications

By

Swagatam Das\textsuperscript{1} and P. N. Suganathan\textsuperscript{2}

\textsuperscript{1}Dept of Electronics and Telecommunication Engg., Jadavpur University
Kolkata 700032, India

\textsuperscript{2}School of Electrical and Electronic Engineering, Nanyang Technological University,
Singapore 639798, Singapore

SSCI 2011
Topics to be Covered

Part – I (By Dr. S. Das)
- Metaheuristics, multi-agent search, and DE
- Steps of the basic DE family of algorithms – a first look
- Control parameters of DE
- Some Significant Single-objective DE-variants for real-parameter optimization
- A stochastic mathematical model of DE-population and convergence analysis

Part – II (By Dr. P. N. Suganthan)
- Ensemble Strategies in DE (EPSDE)
- Constraint Handling in DE and ECHT-DE
- DE for Multi-objective Optimization
- DE for Large Scale Optimization
- DE for Multi-modal Optimization
Meta-heuristics

A metaheuristic is a heuristic method for solving a very general class of computational problems by combining user-given black-box procedures — usually heuristics themselves — in the hope of obtaining a more efficient or more robust procedure. The name combines the Greek prefix "meta" ("beyond", here in the sense of "higher level") and "heuristic" (from ευρίσκειν, heuriskein, "to find").
How a single agent can find global optima by following gradient descent?

\[ \vec{X}(n+1) = \vec{X}(n) - \mu \nabla f(\vec{X}) \]

Direction Of negative gradient
But What about these multi-modal, noisy and even discontinuous functions?

Gradient based methods get trapped in a local minima or the Function itself may be non differentiable.
Multi-Agent Optimization in Continuous Space

Randomly Initialized Agents
After Convergence

Most Agents are near Global Optima
Differential Evolution

• A stochastic population-based algorithm for continuous function optimization (Storn and Price, 1995)

• Finished 3rd at the First International Contest on Evolutionary Computation, Nagoya, 1996 (icsi.berkley.edu/~storn)

• Outperformed GA and PSO on a 34-function test suite (Vesterstrom & Thomsen, 2004)

• Continually exhibited remarkable performance in competitions on different kinds of optimization problems like dynamic, multi-objective, constrained, and multi-modal problems held under IEEE congress on Evolutionary Computation (CEC) conference series.
DE is an Evolutionary Algorithm

This Class also includes GA, Evolutionary Programming and Evolutionary Strategies

Basic steps of an Evolutionary Algorithm
Solutions are represented as vectors of size $D$ with each value taken from some domain.

May wish to constrain the values taken in each domain above and below.
Maintain Population - \( NP \)

We will maintain a population of size \( NP \)

\[
\begin{array}{c|c|c|c|c}
\hline
\tilde{X}_1 & x_{1,1} & x_{2,1} & \cdots & x_{D-1,1} & x_{D,1} \\
\hline
\tilde{X}_2 & x_{1,2} & x_{2,2} & \cdots & x_{D-1,2} & x_{D,2} \\
\hline
\tilde{X}_{NP} & x_{1,NP} & x_{2,NP} & \cdots & x_{D-1,NP} & x_{D,NP} \\
\hline
\end{array}
\]
Different values are instantiated for each $i$ and $j$.

$$x_{j,i,0} = x_{j,min} + rand_{i,j}[0,1] \cdot (x_{j,max} - x_{j,min})$$

Different $rand_{i,j}[0,1]$ values are instantiated for each $i$ and $j$. 
For each vector select three other parameter vectors randomly.

Add the weighted difference of two of the parameter vectors to the third to form a donor vector (most commonly seen form of DE-mutation):

\[ \vec{V}_{i,G} = \vec{X}_{r_1} + F \cdot (\vec{X}_{r_2} - \vec{X}_{r_3,G}). \]

The scaling factor \( F \) is a constant from \((0, 2)\)
Example of formation of donor vector over two-dimensional constant cost contours

Constant cost contours of Sphere function
Components of the donor vector enter into the trial offspring vector in the following way:

Let $j_{\text{rand}}$ be a randomly chosen integer between 1,...,$D$.

\[
    u_{j,i,G} = \begin{cases} 
        v_{j,i,G}, & \text{if } (\text{rand}_{i,j}[0,1] \leq Cr \text{ or } j = j_{\text{rand}}) \\ 
        x_{j,i,G}, & \text{otherwise,} 
    \end{cases}
\]
An Illustration of Binomial Crossover in 2-D Parametric Space:

Three possible trial vectors:

i) \( \vec{U}_{i,G} = \vec{V}_{i,G} \) such that both the components of \( \vec{U}_{i,G} \) are inherited from \( \vec{V}_{i,G} \).

ii) \( \vec{U}_{i,G}' \), in which the first component \( (j = 1) \) comes from \( \vec{V}_{i,G} \) and the second one \( (j = 2) \) from \( \vec{X}_{i,G} \).

iii) \( \vec{U}_{i,G}'' \), in which the first component \( (j = 1) \) comes from \( \vec{X}_{i,G} \) and the second one \( (j = 2) \) from \( \vec{V}_{i,G} \).
Exponential (two-point modulo) Crossover:

First choose integers \( n \) (as starting point) and \( L \) (number of components the donor actually contributes to the offspring) from the interval \([1, D]\)

\[
\begin{align*}
    u_{j,i,G} = \begin{cases} 
        v_{j,i,G}, & \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \ldots, \langle n+L-1 \rangle_D \\
        x_{j,i,G}, & \text{for all other } j \in [1, D],
    \end{cases}
\end{align*}
\]

where the angular brackets \( \langle \cdot \rangle_D \) denote a modulo function with modulus \( D \).

Pseudo-code for choosing \( L \):

\[
\begin{align*}
    & L = 0; \\
    & \text{DO} \\
    & \quad \{ \\
    & \quad \quad L = L+1; \\
    & \quad \} \text{ WHILE } ((\text{rand}[0,1] \leq Cr) \text{ AND } (L < D)) \; ;
\end{align*}
\]
Example: Let us consider the following pair of donor and target vectors

\[
\tilde{X}_{i,G} = \begin{bmatrix}
3.82 \\
4.78 \\
-9.34 \\
5.36 \\
-3.77
\end{bmatrix}
\]  
\[
\tilde{V}_{i,G} = \begin{bmatrix}
8.12 \\
10 \\
-10 \\
-3.22 \\
-1.12
\end{bmatrix}
\]

Suppose \( n = 3 \) and \( L = 3 \) for this specific example. Then the exponential crossover process can be shown as:
“Survival of the fittest” principle in selection: The trial offspring vector is compared with the target vector and that on with a better fitness is admitted to the next generation.

\[
\tilde{X}_{i,G+1} = \tilde{U}_{i,G}, \text{ if } f(\tilde{U}_{i,G}) \leq f(\tilde{X}_{i,G})
\]

\[
= \tilde{X}_{i,G}, \text{ if } f(\tilde{U}_{i,G}) > f(\tilde{X}_{i,G})
\]
An Example of Optimization by DE

Consider the following two-dimensional function

$$f (x, y) = x^2 + y^2$$  \(\text{The minima is at } (0, 0)\)

Let’s start with a population of 5 candidate solutions randomly initiated in the range \((-10, 10)\)

$$X_{1,0} = [2, -1] \quad X_{2,0} = [6, 1] \quad X_{3,0} = [-3, 5] \quad X_{4,0} = [-2, 6]$$

$$X_{5,0} = [6, -7]$$

For the first vector \(X_1\), randomly select three other vectors say \(X_2, X_4\) and \(X_5\)

Now form the donor vector as, \(V_{1,0} = X_{2,0} + F \cdot (X_{4,0} - X_{5,0})\)

$$V_{1,0} = \begin{bmatrix} 6 \\ 1 \end{bmatrix} + 0.8 \times \left\{ \begin{bmatrix} -2 \\ 6 \end{bmatrix} - \begin{bmatrix} 6 \\ -7 \end{bmatrix} \right\} = \begin{bmatrix} -0.4 \\ 10.4 \end{bmatrix}$$
Now we form the trial offspring vector by exchanging components of \( V_{1,0} \) with the target vector \( X_{1,0} \)

Let \( rand[0, 1) = 0.6 \). If we set \( Cr = 0.9 \), since \( 0.6 < 0.9 \), \( u_{1,1,0} = V_{1,1,0} = -0.4 \)

Again next time let \( rand[0, 1) = 0.95 > Cr \)
Hence \( u_{1,2,0} = x_{1,2,0} = -1 \)

So, finally the offspring is \( U_{1,0} = \begin{bmatrix} -0.4 \\ -1 \end{bmatrix} \)

Fitness of parent: \( f(2, -1) = 2^2 + (-1)^2 = 5 \)
Fitness of offspring: \( f(-0.4, -1) = (-0.4)^2 + (-1)^2 = 1.16 \)

Hence the parent is replaced by offspring at \( G = 1 \)
<table>
<thead>
<tr>
<th>Population at $G = 0$</th>
<th>Fitness at $G = 0$</th>
<th>Donor vector at $G = 0$</th>
<th>Offspring Vector at $G = 0$</th>
<th>Fitness of offspring at $G = 1$</th>
<th>Evolved population at $G = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{1,0} = [2, -1]$</td>
<td>5</td>
<td>$V_{1,0} = [-0.4, 10.4]$</td>
<td>$U_{1,0} = [-0.4, -1]$</td>
<td>1.16</td>
<td>$X_{1,1} = [-0.4, -1]$</td>
</tr>
<tr>
<td>$X_{2,0} = [6, 1]$</td>
<td>37</td>
<td>$V_{2,0} = [1.2, -0.2]$</td>
<td>$U_{2,0} = [1.2, 1]$</td>
<td>2.44</td>
<td>$X_{2,1} = [1.2, 1]$</td>
</tr>
<tr>
<td>$X_{3,0} = [-3, 5]$</td>
<td>34</td>
<td>$V_{3,0} = [-4.4, -0.2]$</td>
<td>$U_{3,0} = [-4.4, -0.2]$</td>
<td>19.4</td>
<td>$X_{3,1} = [-4.4, -0.2]$</td>
</tr>
<tr>
<td>$X_{4,0} = [-2, 6]$</td>
<td>40</td>
<td>$V_{4,0} = [9.2, -4.2]$</td>
<td>$U_{4,0} = [9.2, 6]$</td>
<td>120.64</td>
<td>$X_{4,1} = [-2, 6]$</td>
</tr>
<tr>
<td>$X_{5,0} = [6, 7]$</td>
<td>85</td>
<td>$V_{5,0} = [5.2, 0.2]$</td>
<td>$U_{5,0} = [6, 0.2]$</td>
<td>36.04</td>
<td>$X_{5,1} = [6, 0.2]$</td>
</tr>
</tbody>
</table>
Locus of the fittest solution: DE working on 2D Sphere Function

\[ x(1) = 5.326628e-004 \quad x(2) = -7.506413e-004 \]

Number of function evaluations: 330
Locus of the fittest solution: DE working on 2D Rosenbrock Function

Comment Window

\[
x(1) = 9.990996e-001 \quad x(2) = 9.981907e-001
\]

Number of function evaluations: 825
Five most frequently used DE mutation schemes

“DE/rand/1”: \( \vec{V}_i(t) = \vec{X}_{r_1}(t) + F \cdot (\vec{X}_{r_2}(t) - \vec{X}_{r_3}(t)) \).

“DE/best/1”: \( \vec{V}_i(t) = \vec{X}_{best}(t) + F \cdot (\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t)) \).

“DE/target-to-best/1”: \( \vec{V}_i(t) = \vec{X}_i(t) + F \cdot (\vec{X}_{best}(t) - \vec{X}_i(t)) + F \cdot (\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t)) \).

“DE/best/2”: \( \vec{V}_i(t) = \vec{X}_{best}(t) + F \cdot (\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t)) + F \cdot (\vec{X}_{r_3}(t) - \vec{X}_{r_4}(t)) \).

“DE/rand/2”: \( \vec{V}_i(t) = \vec{X}_{r_1}(t) + F_1 \cdot (\vec{X}_{r_2}(t) - \vec{X}_{r_3}(t)) + F_2 \cdot (\vec{X}_{r_4}(t) - \vec{X}_{r_5}(t)) \).

The general convention used for naming the various mutation strategies is DE/x/y/z, where DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used (exp: exponential; bin: binomial)
Basic Control Parameters of DE:

The Scale Factor $F$:

1) DE is much more sensitive to the choice of $F$ than it is to the choice of $Cr$.

2) The upper limit of the scale factor $F$ is empirically taken as 1. Although it does not necessarily mean that a solution is not possible with $F > 1$, however, until date, no benchmark function that was successfully optimized with DE required $F > 1$.

3) Zaharie derived a lower limit of $F$ and the study revealed that if $F$ is sufficiently small, the population can converge even in the absence of selection pressure.

Zaharie’s Formula for evolution of population-variance in absence of selection:

$$Var(P_{x,G}) = \left(2.F^2 \cdot p_{Cr} - \frac{2.p_{Cr}}{NP} + \frac{p_{Cr}^2}{NP} + 1\right)^G \cdot Var(P_{x,0})$$

Critical Value of $F$:

$$F_{crit} = \sqrt{\frac{1 - \frac{p_{Cr}}{2}}{NP}}$$

The population variance decreases when $F < F_{crit}$ and increases if $F > F_{crit}$. 
Selection and Tuning of $F$ in DE: Some Early Approaches

1) Typically $0.4 < F < 0.95$ with $F = 0.9$ can serve as a good first choice

2) Randomizing $F$ may yield good results over a variety of functions: Price et al. coined the following terms in this context:

Dither: scales the length of vector differentials because the same factor, $F$, is applied to all components of a difference vector. (in 2005, Das et al. demonstrated the use of Dither in improving DE’s performance. They varied $F$ uniformly randomly between 0.5 and 1 for each vector.

Jitter: Generates a new value of $F$ for every parameter in every vector is called jitter.

3) Das et al. also proposed a DE with time-varying scale factor where the value of $F$ is linearly decreased from 1 to 0.5 with a view of promoting exploration of diverse regions of the search volume during earlier stages of search while favoring exploitation during the final stages.

4) A fitness-based adaptation of $F$ was proposed by Ali et al. as:

$$F = \begin{cases} 
\max \left\{ l_{\min}, 1 - \frac{|f_{\max}|}{|f_{\min}|} \right\} & \text{if } \left| \frac{f_{\max}}{f_{\min}} \right| < 1 \\
\max \left\{ l_{\min}, 1 - \frac{f_{\min}}{f_{\max}} \right\} & \text{otherwise},
\end{cases}$$

$l_{\min} = 0.4$ is the lower bound of $F$.


The Crossover Rate \( Cr \):

1) The parameter \( Cr \) controls how many parameters in expectation, are changed in a population member.
2) Low value of \( Cr \), a small number of parameters are changed in each generation and the stepwise movement tends to be orthogonal to the current coordinate axes.
3) High values of \( Cr \) (near 1) cause most of the directions of the mutant vector to be inherited prohibiting the generation of axis orthogonal steps.

Empirical distribution of trial vectors for three different values of \( Cr \) has been shown. The plots were obtained by running DE on a single starting population of 10 vectors for 200 generations with selection disabled.

For schemes like DE/rand/1/bin the performance is rotationally invariant only when \( Cr = 1 \). At that setting, crossover is a vector-level operation that makes the trial vector a pure mutant i.e.

\[
\vec{U}_{i,G} = \vec{X}_{r_i,G} + F \cdot (\vec{X}_{r_2,G} - \vec{X}_{r_3,G}).
\]
The Crossover Rate $Cr$ (Contd.):

A low $Cr$ value (e.g. 0 or 0.1) results in a search that changes each direction (or a small subset of directions) separately.

This is an effective strategy for functions that are separable or decomposable i.e. $f(\vec{X}) = \sum_{i=1}^{D} f_i(x_i)$

**A Fitness-based adaptation scheme for $Cr$:** Recently Ghosh *et al.* suggested the following scheme for fitness-based adaptation of $Cr$. The basic idea is: if the fitness of the donor vector gets worse, value of $Cr$ should be higher and vice-versa.

Define: $\Delta f_{\text{donor}_i} = \left| f(\vec{V}_i) - f(\vec{X}_{\text{best}}) \right|$

If $f(\vec{V}_i) \leq f(\vec{X}_{\text{best}})$, $Cr_i = Cr_{\text{const}}$, Else $Cr_i = Cr_{\text{min}} + \frac{(Cr_{\text{max}} - Cr_{\text{min}})}{1 + \Delta f_{\text{donor}_i}}$

Parametric setup $Cr_{\text{max}} = 0.8$, $Cr_{\text{min}} = 0.1$

yielded fairly robust performance over a wide variety of benchmarks.

The population size $NP$

1) The influence of $NP$ on the performance of DE is yet to be extensively studied and fully understood.

2) Storn and Price have indicated that a reasonable value for $NP$ could be chosen between $5D$ and $10D$ ($D$ being the dimensionality of the problem).

3) Brest and Maučec presented a method for gradually reducing population size of DE. The method improves the efficiency and robustness of the algorithm and can be applied to any variant of a DE algorithm.

4) Mallipeddi and Suganthan proposed a DE algorithm with an ensemble of parallel populations, where the number of Function Evaluations (FEs) allocated to each population is self-adapted by learning from their previous experiences in generating superior solutions. Consequently, a more suitable population size along with its parameter settings can be determined adaptively to match different search / evolution phases.


A Few Significant and Improved Variants of DE for Continuous Single Objective Optimization
**DE with Arithmetic Crossover**

1) In *continuous* or *arithmetic* recombination, the individual components of the trial vector are expressed as a linear combination of the components from mutant/donor vector and the target vector.

   General form: 
   $$ \vec{W}_{i,G} = \vec{X}_{r_1,G} + k_i.(\vec{X}_{r_1,G} - \vec{X}_{r_2,G}) $$

2) 'DE/current-to-rand/1' replaces the binomial crossover operator with the rotationally invariant arithmetic line recombination operator to generate the trial vector by a linear arithmetic recombination of target and donor vectors:

   $$ \vec{U}_{i,G} = \vec{X}_{i,G} + k_i.(\vec{V}_{i,G} - \vec{X}_{i,G}) $$

   which further simplifies to:
   $$ \vec{U}_{i,G} = \vec{X}_{i,G} + k_i.(\vec{X}_{r_1,G} - \vec{X}_{i,G}) + F'.(\vec{X}_{r_2,G} - \vec{X}_{r_3,G}) $$

Change of the trial vectors generated through the discrete and random intermediate recombination due to rotation of the coordinate system.

$\vec{U}^{R\perp}_{i,G}$ and $\vec{U}^{R\parallel}_{i,G}$ indicate the new trial vectors due to discrete recombination in rotated coordinate system.
The ‘jDE’ Algorithm (Brest et al., 2006)

• Control parameters $F$ and $Cr$ into the individual and adjusted them by introducing two new parameters $\tau_1$ and $\tau_2$

• The new control parameters for the next generation are computed as follows:

\[
F_{i,G+1} = \begin{cases} 
F_l + \text{rand}_1 \times F_u & \text{if } \text{rand}_2 < \tau_1 \\
F_{i,G} & \text{else.}
\end{cases}
\]

\[
Cr_{i,G+1} = \begin{cases} 
\text{rand}_3 & \text{if } \text{rand}_4 < \tau_2 \\
Cr_{i,G} & \text{else,}
\end{cases}
\]

\[
\tau_1 = \tau_2 = 0.1 \quad F_l = 0.1,
\]

The new $F$ takes a value from $[0.1, 0.9]$ while the new $Cr$ takes a value from $[0, 1]$.

Self-Adaptive DE (SaDE) (Qin et al., 2009)

- Includes both control parameter adaptation and strategy adaptation

Strategy Adaptation:

Four effective trial vector generation strategies: DE/rand/1/bin, DE/rand-to-best/2/bin, DE/rand/2/bin and DE/current-to-rand/1 are chosen to constitute a strategy candidate pool.

For each target vector in the current population, one trial vector generation strategy is selected from the candidate pool according to the probability learned from its success rate in generating improved solutions (that can survive to the next generation) within a certain number of previous generations, called the Learning Period (LP).
Control Parameter Adaptation:

1) \textit{NP} is left as a user defined parameter.

2) A set of \( F \) values are randomly sampled from normal distribution \( N(0.5, 0.3) \) and applied to each target vector in the current population.

3) \( Cr \) obeys a normal distribution with mean value \( Cr_m \) and standard deviation \( Std = 0.1 \), denoted by \( N(Cr_m, Std) \) where \( Cr_m \) is initialized as 0.5.

4) SaDE gradually adjusts the range of \( Cr \) values for a given problem according to previous \( Cr \) values that have generated trial vectors successfully entering the next generation.

Opposition-based DE (Rahnamayan et al., 2008)

- Three stage modification to original DE framework based on the concept of *Opposite Numbers*:

  Let x be a real number defined in the closed interval [a, b]. Then the opposite number of x may be defined as:

  \[ x = a + b - x \]

**ODE Steps:**

1) **Opposition based Population Initialization**: Fittest NP individuals are chosen as the starting population from a combination of NP randomly generated population members and their opposite members.

2) **Opposition Based Generation Jumping**: In this stage, after each iteration, instead of generating new population by evolutionary process, the opposite population is calculated with a predetermined probability \( Jr () \) and the \( NP \) fittest individuals may be selected from the current population and the corresponding opposite population.
3) **Opposition Based Best Individual Jumping:** In this phase, at first a difference-offspring of the best individual in the current population is created as:

\[
\vec{X}_{\text{new\_best},G} = \vec{X}_{\text{best},G} + F' \cdot (\vec{X}_{r_1,G} - \vec{X}_{r_2,G})
\]

where \( r_1 \) and \( r_2 \) are mutually different random integer indices selected from \{1, 2, ..., \( NP \)\} and \( F' \) is a real constant. Next the opposite of offspring is generated as \( \vec{X}_{\text{opp\_newbest}G} \). Finally the current best member is replaced by the fittest member of the set \( \{\vec{X}_{\text{best},G}, \vec{X}_{\text{new\_best},G}, \vec{X}_{\text{opp\_newbest},G}\} \)
JADE (Zhang and Sanderson, 2009)

1) Uses DE/current-to-pbest strategy as a less greedy generalization of the DE/current-to-best/ strategy. Instead of only adopting the best individual in the DE/current-to-best/1 strategy, the current-to-pbest/1 strategy utilizes the information of other good solutions.

Denoting $\vec{X}_{best,G}^p$ as a randomly chosen vector from the top 100$p$% individuals of the current population,

**DE/current-to-pbest/1 without external archive**: $\vec{V}_{i,G} = \vec{X}_{i,G} + F_i \cdot (\vec{X}_{best,G}^p - \vec{X}_{i,G}) + F_i \cdot (\vec{X}_{r_i,G}^1 - \vec{X}_{r_2,G}^1)$

2) JADE can optionally make use of an external archive (A), which stores the recently explored inferior solutions. In case of DE/current-to-pbest/1 with archive, $\vec{X}_{i,G}^p$, $\vec{X}_{best,G}^p$, and $\vec{X}_{r_i,G}^1$ are selected from the current population $P$, but $\vec{X}_{r_2,G}^1$ is selected from $P \cup A$
JADE (Contd..)

3) JADE adapts the control parameters of DE in the following manner:

A) \( Cr \) for each individual and at each generation is randomly generated from a normal distribution \( N(\mu_{Cr}, 0.1) \) and then truncated to \([0, 1]\).

The mean of normal distribution is updated as: \[ \mu_{Cr} = (1 - c) \mu_{Cr} + c.mean_A(S_{Cr}) \]

where \( S_{Cr} \) be the set of all successful crossover probabilities \( Cr \)s at generation \( G \)

B) Similarly for each individual and at each generation \( F_i \) is randomly generated from a Cauchy distribution \( C(\mu_F, 0.1) \) with location parameter \( \mu_F \) and scale parameter 0.1.

\( F_i \) is truncated if \( F_i > 1 \) or regenerated if \( F_i \leq 0 \)

The location parameter of the Cauchy distribution is updated as: \[ \mu_F = (1 - c) \mu_F + c.mean_L(S_F) \]

where \( S_F \) is the set of all successful scale factors at generation \( G \) and \( mean_L \) is the Lehmer mean:

\[ mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \]

JADE usually performs best with \( 1/c \) chosen from \([5, 20]\) and \( p \) from \([5\%, 20\%]\)
Local Mutation Model:

$$\vec{L}_i(t) = \vec{X}_i(t) + \alpha \cdot (\vec{X}_{n\_best_i}(t) - \vec{X}_i(t)) + \beta \cdot (\vec{X}_p(t) - \vec{X}_q(t))$$

Global Mutation Model:

$$\vec{g}_i(t) = \vec{X}_i(t) + \alpha \cdot (\vec{X}_{g\_best}(t) - \vec{X}_i(t)) + \beta \cdot (\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t))$$

Combined Model for Donor Vector generation:

$$\vec{V}_i(t) = w.\vec{g}_i(t) + (1 - w).\vec{L}_i(t)$$

The weight factor $w$ may be adjusted during the run or self-adapted through the evolutinal learning process.
On Timing Complexity of DEGL

Running of $G_{\text{max}}$ no. of generations, complexity of DE/rand/1/bin :

$$O(NP \cdot D \cdot G_{\text{max}})$$

Runtime complexity of DE/target-to-best/1/bin :

$$O(\max(NP \cdot G_{\text{max}}, NP \cdot D \cdot G_{\text{max}})) = O(NP \cdot D \cdot G_{\text{max}})$$

Worst case timing complexity of DEGL: 

$$O(\max(NP \cdot k \cdot G_{\text{max}}, NP \cdot D \cdot G_{\text{max}}))$$

where $k$ is neighborhood radius.

Asymptotic order of complexity remains 

$$O(NP \cdot D \cdot G_{\text{max}})$$

if $k < D$, which is usually the case for high-dimensional functions.

DEGL does not impose any serious burden on the runtime complexity of the existing DE variants.
A Special Issue on DE just appeared in:
Feb. 2011

Guest Editors: Swagatam Das, Jadavpur University, India
P. N. Suganthan, Nanyang Technological University, Singapore.
Carlos A. Coello Coello, Col. San Pedro Zacatenco, México.

This tutorial is partly based on the survey article published in this special issue:

S. Das and P. N. Suganthan, “Differential evolution – a survey of the state-of-the-art”,
Introduction to a few DE-variants from the special issue of IEEE TEVC:

   - An interesting Estimation of Distribution Algorithm (EDA) which, analogously to other compact Evolutionary Algorithms (cEAs), does not store and process the entire population and all of its individuals, but adopts instead a static representation of the population to perform the optimization process.
   - However, unlike other cEAs, cDE encodes the actual survivor selection mechanism of the original DE algorithm, as well as its original search logic (i.e., its crossover and mutation operators).
   - cDE uses a fairly limited amount of memory, which makes it appropriate for hardware implementations.

-adopts three different trial vector generation strategies (rand/1/bin, rand/2/bin, and current-to-rand/1) and three mechanisms to control DE’s parameter settings. The parametric choices were:

1) $F = 1.0$, $Cr = 0.1$ - meant for dealing with separable problems.
2) $F = 1.0$, $Cr = 0.9$ – meant for maintaining population diversity.
3) $F = 0.8$, $Cr = 0.2$ – meant for exploiting the search space and achieving better convergence characteristics.

-Such strategies offer different advantages and, somehow, complement one another.

-is based on framework that incorporates information of neighboring individuals to guide the search towards the global optimum in a more efficient manner.

-The main idea is to adopt a stochastic selection mechanism in which the probability of selecting an individual to become a parent is inversely proportional to its distance from the individual undergoing mutation.

-This will favor the search in the vicinity of the mutated individual, which should promote a proper exploitation of such a neighborhood, without sacrificing the exploration capabilities of the mutation operator.

-The authors incorporate the proposed framework to several DE variants, finding that in most cases its use significantly improves the performance of the algorithm (when there is no improvement, there is no significant degradation in performance either).
Stochastic Analysis of the DE

- DE has been modeled as a stochastic process with a time-varying PDF.

- The mutation, crossover and selection steps have been analyzed to produce a recurrence relation relating the PDF at time $n+1$ with the present PDF for time $n$.

- Investigation of the relation shows the existence of a Lyapunov functional (dependent on the PDF) which proves asymptotic stability of the DE system.

Assumptions on the Objective Function:

The objective function $f(\bar{X})$:

1) is continuous and belongs to class of $C^2$ (please note that a function $f$ is said to be of class $C^k$ if the derivatives $f^1, f^2, \ldots, f^k$ exist and are continuous).

2) possesses a unique global optimum in the range of search. Note that the function can still remain multimodal as it can have multiple local optima.

The high-dimensional integral of a scalar function like $g(\bar{x})$ over several components of a vector i.e.

$$\int \int \ldots \int g(x_1, x_2, \ldots x_D).dx_1.dx_2 \ldots dx_D$$

is expressed through a single integration over a vector space like

$$\int_{\mathbb{R}^D} g(\bar{x}).d\bar{x}$$
Towards a recursive relation of the population PDFs

In binomial crossover, we assume a component is inherited from donor vector with probability $P_C$ and from the corresponding target vector with a probability $1 - P_C$.

For the $m$-th crossover combination, we define a $D$-dimensional string $\theta_m$ such that a ‘1’ in $\theta_{m,d}$ (the $d$-th bit position of $\theta_m$) indicates that the $d$-th dimension is taken from the donor, and a ‘0’ indicates that the $d$-th dimension is taken from the parent. For example, $m = 2$ and $D = 3$ is represented by ‘001’.

We define the following sets: $\alpha_m = \{a : 1 \leq a \leq D$ and $\theta_{m,a} = 1\}$ and $\beta_m = \{b : 1 \leq b \leq D$ and $\theta_{m,b} = 0\}$.

The probability of the $m$-th combination appearing in the crossover: $P_m = \binom{D}{\nu} C_\nu P_C^\nu (1 - P_C)^{D-\nu}$

$\nu = \text{Cardinality of set } \alpha_m$
Next we prove the following:

1) At time $n$ the PDF of the donor vector corresponding to the $i$-th target vector is given by:

$$p_{V_{i,n}}(V_{i,n}) = \frac{1}{F^{2N_p-1}} P_3 \sum_{a \neq b \neq c \neq i} \sum \sum p_{X_{a,n}}(V_{i,n}) \ast p_{X_{b,n}}\left(\frac{V_{i,n}}{F}\right) \ast p_{X_{c,n}}\left(\frac{\bar{V}_{i,n}}{F}\right)$$

2) If the random variable $\vec{X}_n$ represents any population-member at time $n$, and the random variable $\vec{X}_{n+1}$ represents the same member at time $n+1$, then the PDF of $\vec{X}_{n+1}$ is given by:

$$p_{\vec{X}_{n+1}}(\vec{x}_{n+1}) = \sum_m P_m \cdot p_{\vec{X}_n}(\vec{x}_{n+1}) \int_{f(x_{n+1}) < f(\vec{\lambda}, \vec{x}_{n+1}, \beta_m)} p_{\vec{X}_n}(\vec{\lambda}, \vec{\mu}). \, d\vec{\lambda} \, . d\vec{\mu}$$

$$+ \sum_m P_m \cdot \int_{f(x_{n+1}) \geq f(\vec{\gamma}, x_{n+1}, \beta_m)} p_{\vec{X}_n}(\vec{\gamma}, x_{n+1}, \beta_m). p_{\bar{V}_n}(\vec{x}_{n+1}, \alpha_m, \vec{\eta}). \, d\vec{\gamma} \, . d\vec{\eta}$$
Terms in the relation:

\[ p_{\vec{x}_n} \text{ and } p_{\vec{x}_{n+1}} \text{ are PDFs of the population vectors at times } n \text{ and } n+1 \text{ respectively. We have shown that the PDFs for all vectors in the population are identical, so we do not refer to any vector in particular.} \]

\[ p_{\vec{V}_n} \text{ is the PDF of the donor vectors at time } n. \text{ Here also we do not refer to any vector in particular.} \]

\[ \text{Our objective is to express } p_{\vec{x}_{n+1}} (\vec{x}_{n+1}) \text{ in terms of } p_{\vec{x}_{n+1}} \text{ and } p_{\vec{V}_n} \text{ for the point } \vec{x}_{n+1}. \]
What does it mean?

- The recurrence relation portrays the heuristics in the DE algorithm.
- Failure of the population PDF at time $n$ to find good solutions may contribute to selection of the donor PDF.
- Similarly, failure of the donor PDF to find good solutions may contribute to selection of the original population PDF.
The Lyapunov Functional

• Investigation of the recurrence relation shows that there exists a Lyapunov functional dependent on the population PDF

\[ V[p(\bar{x})] = \left( \int_{\mathbb{R}^D} f(\bar{x}) p(\bar{x}) \, d\bar{x} \right) - f(\bar{x}^*) \]

• The Lyapunov functional strictly decreases to zero with time, hence the dynamics is asymptotically convergent at the equilibrium PDF:

\[ p_E(\bar{x}) = \delta(\bar{x} - \bar{x}^*) \]
We prove the following things about the Lyapunov Functional:

1) The functional 
\[
V[p(\bar{x})] = \left( \int_{\mathbb{R}^d} f(\bar{x}) p(\bar{x}) \, d\bar{x} \right) - f(\bar{x}^*)
\]
is positive definite w.r.t the equilibrium PDF 
\[
p_E(\bar{x}) = \delta(\bar{x} - \bar{x}^*)
\]

2) The functional 
\[
V[p_{X_{n+1}}(\bar{x})] - V[p_{X_n}(\bar{x})]
\]
defined on the set of all PDFs 
\[p(\bar{x})\]
is negative semi-definite w.r.t the equilibrium PDF 
\[
p_E(\bar{x}) = \delta(\bar{x} - \bar{x}^*)
\]

3) For the functional dynamics arising from the transition equation, 
\[
V[p_{X_{n+1}}(\bar{x})] - V[p_{X_n}(\bar{x})]
\]
does not identically vanish along any trajectory taken by the PDF 
\[
p_{\bar{x}_n}(\bar{x}_n)
\]
for all 
\[n \geq 0\]

Dynamics given by the PDF transition equation is asymptotically stable at the equilibrium PDF 
\[
p_E(\bar{x}) = \delta(\bar{x} - \bar{x}^*)
\]
How correct is the model?

• To check the correctness of our model, we test the actual DE, with DE/1/rand/bin mutation and a large population size (1000), and apply it to some standard one-dimensional benchmarks.

• The estimated PDFs obtained through the experiment are found to be pretty close to the PDFs predicted through the recurrent stochastic model.

• The estimated Lyapunov functional also reduces to zero and is very close to the predicted Lyapunov functional.
The Sphere Function

PDFs predicted for different time instants through the recurrent stochastic model

PDFs estimated by running the DE algorithm
Griewank’s Function

PDFs predicted for different time instants through the stochastic model

PDFs estimated by running the DE algorithm
Shifted Rastrigin’s Function

PDFs predicted for different time instants through the stochastic model

PDFs estimated by running the DE algorithm
Conclusions

• The stochastic model accurately predicts the behavior of the DE algorithm for a large population size.

• The model is also successful in showing that the population vectors converge at the global optimum point, provided it exists uniquely.

• Further research can be undertaken to predict the algorithm’s behavior for a finite number of vectors.