Differential Evolution

Jugal K. Kalita
Requirements on a Practical Minimization Technique (Storn and Price 1997)

- Ability to handle non-differentiable, nonlinear and multimodal cost functions.
- Parallelizability to cope with computation intensive cost functions.
- Ease of use, in particular, there should be few control variables or parameters.
- Good convergence properties, i.e., consistent convergence to the global minimum (optimum) in consecutive independent trials.
Motivation for Introducing Differential Evolution (Storn and Price 1997)

- DE was designed to be a stochastic direct search method. In other words, it satisfies the first requirement since it can be used to minimize any function for which a global minimum exists.
- DE uses a population of vectors where the stochastic perturbation of the vectors can be done independently. In other words, parallelization is easy.
Ease of use: To satisfy the third requirement, it is good if the minimization method is self-organizing so that it requires very little input from the user. DE alters the search space using information from within the vector population (like Genetic Algorithms). DE’s self-organizing scheme takes the difference of two randomly chosen population vectors to perturb an existing vector. The perturbation is done to every population vector.

Convergence: DE has been found to converge very well in a large number of practical problems although theoretical discussions are not extensive.
Differential Evolution: Basic Components

- DE is a parallel population-based direct search method where the population is comprised of $NP$ vectors each of dimension $D$.
  - $NP$ does not change during the algorithm’s life time.
- The initial population is chosen randomly and should cover the entire parameter space.
  - The usual assumption is that all features or attributes vary in a uniform manner, unless otherwise stated.
  - If an initial or preliminary solution can be computed or is available, the initial population can be obtained by adding normally distributed random deviations to the preliminary solution.
Differential Evolution: Basic Components...

- **Mutation**: DE generates new individuals in the population by adding weighted difference between two population vectors to a third vector (the mutated vector).

- **Crossover**: The mutated vector’s parameters are then mixed with the parameters of another pre-determined vector, the target vector, to yield a trial vector.

- **Selection**: If the trial vector yields a lower cost function value than the target vector, the trial vector replaces the target vector in the next generation.

- Each population vector must serve once as the target vector so that NP competitions take place in one generation.
Mutation

- Let the population in generation $G$ be
  \[ x_{i,G} \in \{1, 2, \ldots, NP \} \]
- Each member of the population serves as a target vector.
- For each target vector $x_{i,G}$ in generation $G$, a mutant vector is generated for generation $G+1$ according to
  \[ v_{i,G+1} = x_{r_1,G} + F \times (x_{r_2,G} - x_{r_3,G}) \]
  with random indexes $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$. The three indexes are mutually different. The three indexes are also different from index $i$. $F \in [0, 2]$ and controls the amplification of the differential variation $\left(x_{r_2,G} - x_{r_3,G}\right)$. 

University of Colorado, Colorado Springs, USA
Mutation

Figure: An example of a 2-D cost function showing its contour lines and the process of generation $v_{i,G+1}$. 
The goal is to increase the diversity of the perturbed population vectors.
To achieve this goal, the trial vector is computed in the following manner.
The trial vector for generation $G + 1$ is given as

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \cdots u_{Di,G+1})$$
Crossover...

- Each component of the trial vector for generation $G+1$ is formed in the following manner.

$$u_{ji,G+1} = \begin{cases} 
  v_{ji,G+1} & \text{if } (\text{randb}(j) \leq CR) \text{ or } j = \text{rnbr}(i) \\
  x_{ji,G} & \text{if } (\text{randb}(j) > CR) \text{ or } j \neq \text{rnbr}(i).
\end{cases}$$

- Here, $\text{randb}(j)$ is the $j$th evaluation of a uniform random number generator with outcome $\in [0, 1]$. $CR$ is the crossover constant $\in [0, 1]$, given by the user. $\text{rnbr}(i)$ is a randomly chosen index $\in 1, 2, \cdots D$, which ensures that $u_{i,G+1}$ gets at least one parameter from $v_{i,G+1}$. 

University of Colorado, Colorado Springs, USA
Selection

To decide whether or not it should become a member of generation $G+1$, the trial vector $u_{i,G+1}$ is compared with the target vector $x_{i,G}$. If vector $u_{i,G+1}$ yields a smaller value than $x_{i,G}$, then $x_{i,G+1}$ is set to $u_{i,G+1}$; otherwise, the old value $x_{i,G}$ is retained.
Mutation

**Figure**: Illustration of the crossover process for $d = 7$ parameters or features
The Basic DE Algorithm

Initialize all agents $X$ with random positions in search space

repeat
  for Each agent $X$ in the population do
    Pick three agents $r_1, r_2, r_3$
    Pick a random index between 1 and $NP$
    Compute the agent’s potentially new position
    for For each component of the agent do
      Copy or not copy based on the cases discussed earlier
    end for
  end for
until termination criterion is met, (e.g. number of iterations performed, or adequate fitness reached),
Pick the agent from the population that has the lowest cost and return it as the best found candidate solution.
Variants of DE

- The scheme $DE/x/y/z$ is used to describe variants of DE.
- $x$ specifies the vector to be mutated; examples: $rand$ for random, $best$ for the lowest cost vector from the current population.
- $y$ is the number of difference vectors used.
- $z$ is the crossover scheme.
- The scheme described earlier can be written as $DE/rand/1/bin$ where $bin$ is crossover using random numbers discussed earlier.
Another variant that (Price 1996) found useful is \( DE/best/2/bin \) where

\[
v_{i,G+1} = x_{best,G} + F \times \left( x_{r_1,G} + x_{r_2,G} - x_{r_3,G} - x_{r_4,G} \right)
\]