Differential Evolution

Differential Evolution

Differentiation evolution (DE,) is a popular stochastic population based optimization algorithm, originally developed by R. Storn and K. Price in 1997.

Differential Evolution

In some sense, differential evolution a conceptual hybridization between a PSO and GA (though not proposed as such).

- From GA's the following concepts are used:
  - Mutation
  - Cross-Over
  - Selection
- From PSO
  - The use of positions differences between candidate solutions.
  - Though **no** communication is present.

Differential Evolution: Base Version

Much like PSO you start with an initial population uniformly sampled over the search space.

- Let $\Omega(t)$ be the population at generation t in D-dimensional space
- Let $N_t = |\Omega(t)|$ be the population size at time set t
  - In general DE's population size is constant so $N = N_t$
- For **each** individual (called **target** vectors in DE) in the population, the following sequence is performed, in this order,
  - Mutation
  - Crossover
  - Selection
- One complete pass is seen as a generation.

Differential Evolution: Base Version (Mutation)

The mutation operations of DE is considerably different to that of its counter part GA. The mutation scheme for each **target** vector, $\mathbf{x}_i$, is as follows:

- Let $\mathcal{I} = \{1, 2, \cdots N\}$ be the set of indices for each target vector in the population.
- Select $i_1, i_2, i_3 \in \mathcal{I}$ randomly such that $|\{i_1\} \cup \{i_2\} \cup \{i_3\}| = 3$.
- Generate a **mutant** vector

$$\mathbf{m}_i = \mathbf{x}_{i_1} + \beta \left(\mathbf{x}_{i_2} - \mathbf{x}_{i_3}\right) \tag{1}$$

where $\beta \geq 0$ is a constant scalar that controls the differential variation. Originally recommendation is $\beta \in [0, 2]$

Differential Evolution: Base Version (Crossover)

From the each **mutant** vector, $\mathbf{m}_i$, the following procedure is used to generate **trial** vector $\mathbf{t}_i$:

- Generate a random index, $ri \in \mathcal{I}$
- Generate a vector $\mathbf{r} \sim U[0,1]^D$

$$t_{i,j} = \begin{cases} m_{i,j} & \text{if } r_j \leq CR \text{ or } j = ri \\ x_{i,j} & \text{otherwise} \end{cases} \tag{2}$$

for each $j \in 1, 2, \cdots, D$, where D is th problem dimensionality, and CR is the cross over probability.

Differential Evolution: Base Version (Selection)

The base DE's selection is a purely greedy approach. Namely,

- We add the **trail** vector to $\Omega(t+1)$ if $f(\mathbf{t}_i) < f(\mathbf{x}_i)$.
    - Where $f : \mathbb{R}^D \to \mathbb{R}$ is the objective function we are minimization.
- Alternatively we just carry the original **target** vector over to $\Omega(t+1)$

Differential Evolution: Variant scheme

DE are conveniently classified by the form *DE/x/y/z*, where

- x=specifies the vector, $\mathbf{x}_{i_1}$, to be mutated. (sometimes called the target selection method)
  - Which in the base form was random, which is denoted by *rand*
  - If the best target vector was selected this would be denoted by *best*
- y=The number of difference vectors used.
  - which in the base DE was just 1, namely $(\mathbf{x}_{i_2} - \mathbf{x}_{i_3})$
- z=The crossover scheme.
  - which in the base DE was a binary crossover, denoted by *bin*
  - This a minor abuse, as the crossover approach uses is not a **pure** binary crossover.

As such the base DE we have discussed is denoted at *DE/rand/1/bin*

Differential Evolution: Understanding the Influence of DE's Control Parameters

DE/rand/1/bin has three control parameters, the population size N, the differential scaling factor β, and lastly the crossover rate CR.

- **Population size**:The size of the population has a direct influence on the exploration ability of DE algorithms.
    - why?

Differential Evolution: Understanding the Influence of DE's Control Parameters

DE/rand/1/bin has three control parameters, the population size N, the differential scaling factor β, and lastly the crossover rate CR.

- **Population size**:The size of the population has a direct influence on the exploration ability of DE algorithms.
  - why?
  - The larger N the more possible difference vectors we have. Specifically, we have $2\binom{N-1}{2}$ possible vectors for *DE/rand/1/bin*. (why minus 1?)
  - This implies there are more possible direction that can be explored.
  - Recall that the larger N is the quicker you will use up your function evaluations.

Differential Evolution:
Understanding the Influence of DE's Control Parameters

- **Scaling factor:**.
  - The smaller the value of β, the smaller the mutation step sizes, and the longer it can take for the algorithm to converge.
  - Larger values for β facilitate exploration, but may cause the algorithm to overshoot good optima.
  - The value of β should be small enough to allow differentials to explore tight valleys, and large enough to maintain diversity.
    - Clearly the optimal β is problem dependant, though $\beta = 0.5$ is often recommended.
  - Common claim is "As the population size increases, the scaling factor should decrease." is claim seems intuitive but there is a caveat.

Differential Evolution: Understanding the Influence of DE's Control Parameters

- **Scaling factor:**.
  - The smaller the value of β, the smaller the mutation step sizes, and the longer it can take for the algorithm to converge.
  - Larger values for β facilitate exploration, but may cause the algorithm to overshoot good optima.
  - The value of β should be small enough to allow differentials to explore tight valleys, and large enough to maintain diversity.
    - Clearly the optimal β is problem dependant, though $\beta = 0.5$ is often recommended.
  - Common claim is "As the population size increases, the scaling factor should decrease." is claim seems intuitive but there is a caveat.
    - The average distance of uniformly distributed random vectors in $\mathbb{R}^D$ approach a constant as $N \to \infty$

Differential Evolution:
Understanding the Influence of DE's Control Parameters

- **Crossover rate (Recombination probability):**.
  - The larger *CR* is the more components of the trial vector $\mathbf{t}_i$, will on average, come from the mutant vector $\mathbf{m}_i$
  - This tells use that
    - ⋆ The higher *CR* is, the exploration more will occur
    - ⋆ If *CR* is very low we basically create trial vectors that are very similar to target vectors (more exploitative)

Differential Evolution: Common DE/x/y/x Variants

Some of the most common variants are described as

- *DE/best/1/z*: Basically $\mathbf{x}_{i_1}$ is set to the best target vector in the population, $\hat{\mathbf{x}}$, so the construction of the mutated vector is done as

$$\mathbf{m}_i = \hat{\mathbf{x}} + \beta\left(\mathbf{x}_{i_2} - \mathbf{x}_{i_3}\right) \tag{3}$$

- *DE/x/d/z*: Instead of using only 1 difference vector we could use any d many difference vectors, so the construction of the mutated vector is done as

$$\mathbf{m}_i = \mathbf{x}_{i_1} + \beta\sum_{k=1}^{d}\left(\mathbf{x}_{i_2,k} - \mathbf{x}_{i_3,k}\right) \tag{4}$$

  which also imposed the restriction that $|\{i_1\} \cup \{i_2, 1\} \cup \{i_3, 1\} \cup \cdots \cup \{i_2, d\} \cup \{i_3, d\}| = 1 + 2d \leq N$.

Differential Evolution: Common DE/x/y/x Variants

- *DE/x/d/z*: There are far more possible difference vectors, specifically in the literature it is claimed
  - that there are $\binom{N-1}{2d}(2d)!$ difference vectors,
  - but this does not represent the maximum number of distinct vector that $\sum_{k=1}^{d}\left(\mathbf{x}_{i_2,k} - \mathbf{x}_{i_3,k}\right)$ could generate, why?

Differential Evolution: Common DE/x/y/x Variants

- *DE/x/d/z*: There are far more possible difference vectors, specifically in the literature it is claimed
    - that there are $\binom{N-1}{2d}(2d)!$ difference vectors,
    - but this does not represent the maximum number of distinct vector that $\sum_{k=1}^{d}\left(\mathbf{x}_{i_2,k} - \mathbf{x}_{i_3,k}\right)$ could generate, why?
    - The order of the pairs does not matter, so divide by $d!$

Differential Evolution: Common DE/x/y/x Variants

- *DE/rand-to-best/d/z*: This strategy combines the rand and best strategies to calculate the trial vector as follows:

$$\mathbf{m}_i = \gamma\hat{\mathbf{x}} + (1-\gamma)\mathbf{x}_{i_1} + \beta\sum_{k=1}^{d}(\mathbf{x}_{i_{2,k}} - \mathbf{x}_{i_{3,k}}) \tag{5}$$

  where $\gamma \in [0,1]$ controls the greediness of the mutation operator.
  - $\gamma = 1 \rightarrow$ *DE/best/d/z*
  - $\gamma = 0 \rightarrow$ *DE/rand/d/z*

Differential Evolution: Common DE/x/y/x Variants

- *DE/current-to-best/1+d/z*: With this strategy, the parent is mutated using at least two difference vectors.

$$\mathbf{m}_i = \mathbf{x}_i + \beta(\hat{\mathbf{x}} - \mathbf{x}_i) + \beta \sum_{k=1}^{d} (\mathbf{x}_{i_1,k} - \mathbf{x}_{i_2,k}) \tag{6}$$

- One difference vector is calculated from the best vector and the parent vector, while the rest of the difference vectors are calculated using randomly selected vectors:

Differential Evolution: Common DE/x/y/x Variants

- *DE/x/y/exp*: the other common mutation strategy for DE is exponential crossover
- Unlike binary cross over where individual components are selected, in exponential crossover
    - a sequence of adjacent crossover points are selected
    - In the approach the vector components are treated as a circular list.
        - so component D and component 1 are adjacent.
    - The approach is as follows:
        - Randomly select a index $j \in \{1, \cdots, D\}$, this point will be part of the crossover
        - Then $(j+1)$ will be included if $r \leq CR$ where $r \sim U(0,1)$
        - If $j+1$ was included repeat for $j+2$ etc.
        - Until we fail to include or the whole vector is included.
    - The exponential crossover is effective when linkages exist between the neighbouring decision variables.

Differential Evolution: Less Common Variants

- *DE/2-opt/1/z*: This approach is a slight spin on the *DE/rand/1/z* version of DE.
- The mutant vector is constructed

$$\mathbf{m}_i = \begin{cases} \mathbf{x}_{i_1} + \beta\left(\mathbf{x}_{i_2} - \mathbf{x}_{i_3}\right) & \text{if } f(\mathbf{x}_{i_1}) < f(\mathbf{x}_{i_2}) \\ \mathbf{x}_{i_2} + \beta\left(\mathbf{x}_{i_1} - \mathbf{x}_{i_3}\right) & \text{else} \end{cases} \tag{7}$$

- This approach is a compromise in-between the exploitative extreme of *DE/best/1/z* and the explorative extreme of *DE/rand/1/z*.
- The approach can be applied for any y.

Differential Evolution: Less Common Variants

- *DE/Proximity-based/1/z* (ProDE): Epitropakis et al. proposed a proximity induced mutation scheme for DE,
  - where neighbours of a parent vector, rather than the random ones will be used to generate the mutant vector.
    - this is not to be confused with PSO's neighbourhood structure
  - First computes the pair-wise distance between all members of a population, and stores them in a matrix say **R** where $r_{i,j} = d(\mathbf{x}_i, \mathbf{x}_j)$ (in practise use an upper triangular one rather)
  - From **R** we build the probability matrix $\mathbf{R}^p$ using

$$r^p_{i,j} = 1 - \frac{r_{i,j}}{\sum_{k=1}^{N} r_{i,k}} \qquad (8)$$

  - You then select $i_1, i_2, i_3 \in \mathcal{I}/\{i\}$ using non replacement roulette wheel selection based on $\mathbf{R}^p$

Differential Evolution: Binary Optimization

DE was designed for continuous optimization.

- However, it can be applied to binary problems. The two most common method as near identical with those used to PSO binary problems.
- The approaches are
  - *binDE/x/y/z*
  - *AMDE/x/y/z*

Differential Evolution: Binary Optimization

binDE/x/y/z

- Operates the same as $DE/x/y/z$ expect the fitness of trial and target vectors by constructing a transformed vector

$$y_{i,j} = \begin{cases} 0 & \text{if } Sig(x_{i,j}) \geq 0.5 \\ 1 & \text{if } Sig(x_{i,j}) < 0.5 \end{cases} \tag{9}$$

Where

$$Sig(x_{i,j}) = \frac{1}{1 + e^{-1}} \tag{10}$$

then the fitness of $\mathbf{x_i}$ is $f(\mathbf{y_i})$

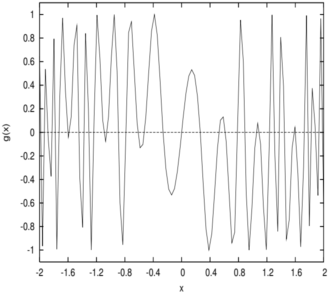
Differential Evolution: Binary Optimization

AMDE/x/y/z

- Just as in Angular modulated PSO, the DE optimizes in 4-dimensional continuous space and then converts the target and trail vectors to a binary string before evaluating their fitness.

$$g(x) = \sin(2\pi(x - a) \times b \times \cos(2\pi(x - a) \times c)) + d \qquad (11)$$

sampled at evenly spaced positions, x


g(x)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-2
-1.6
-1.2
-0.8
-0.4
0
0.4
0.8
1.2
1.6
2
x