DE vs. EA

• Similarities:
  – A stochastic population-based search algorithm.
  – Use mutation, crossover and selection operators.

• Differences:
  – Mutate, cross then select (most EAs apply these operators in reverse order)
  – Use distance and direction information from the current population to guide the search process

Vector Algebra

• DE uses distance and direction information from the current population to guide the search process.
  • In x dimension,
    \[ \overrightarrow{AB} = 3 - 5 = -2 \]
  • Distance is 2;
  • Direction is the direction of A to B.

Geometrical Illustration of DE

\[ v_i(t) = x_{r0}(t) + \beta \times (x_{r1}(t) - x_{r2}(t)) \]

Target vector

Donor vector

Trial vector

Base vector

Two randomly selected vectors

Trial vector after Crossover
DE Variants

- **DE variants notation:** `DE/x/y/z`
  - `x` refers to the method used to select the base vector (`x_0`);
  - `y` indicates the number of vector pairs (difference vectors) used for differential mutation.
  - `z` indicates the crossover point selection method.

- **Example:** `DE/rand/1/bin`
  - the base vector `x_0` is randomly selected; 1 pair of vector is used for differential mutation; binomial crossover point selection.

Variation on X

- **`DE/rand/1/z`:** the base vector `x_0` is randomly selected; 1 pair of vector is used to for differential mutation:
  \[ v_i(t) = x_{r0}(t) + \beta \times (x_{r1}(t) - x_{r2}(t)) \]

- **`DE/best/1/z`:** The best vector \( \hat{x}(t) \) in the population is used as the base vector; 1 pair of vector is used for differential mutation:
  \[ v_i(t) = \hat{x}(t) + \beta \times (x_{r1}(t) - x_{r2}(t)) \]

Best vs. Random

- **Best vector:**
  - *Exploitation* of known best solution.
  - Lead to faster convergence.
  - Reduce the search space exploration.
  - Less efficient for complex problems with many local optima.

- **Random vector:**
  - Has the opposite effect of using best vector as the base vector.

Best and Random Combination

- **`DE/rand-to-best/1/z`:** each dimension of the base vector is selected either from the best or the random vector based on the parameter `γ`
  \[ v_i(t) = γ \times \hat{x}(t) + (1 - γ) \times x_{r0}(t) + \beta \times (x_{r1} - x_{r2}) \]

- Larger `γ` is greedier, as the donor vector is based more on the best solution. When `γ` is 1, it is equal to `DE/best/γ/z`. 
Variation on Y

- The larger the number of vector pairs used to compute the difference vector, the more directions can be explored per generation.

\[ v_i(t) = x_{r_0}(t) + \beta \sum_{k=1}^{n} (x_{r_1,k}(t) - x_{r_2,k}(t)) \]

- Examples: DE/rand/2/z, DE/best/3/z

Number of Difference Vectors

- One difference vector: \( \beta \times (x_{i1} - x_{i2}) \)
- Two difference vectors: \( \beta \times (x_{i1} - x_{i2}) + \beta \times (x_{i3} - x_{i4}) \)
- Large number of pairs:
  - Explore more directions in the search space.
  - Lead to slower convergence.
  - More efficient for complex problems with many local optima.

Weight Factor

- In [Storn and Price, 1997], the weight factor \( \beta \) is between 0 and 2.
- In [Qin and Suganthan, 2005] and [Ali and Torn, 2004], the standard DE they described has scale factor \( F \) between 0 and 1.
- The smaller the value of \( \beta \), the smaller the mutation step size, and the longer it will be for the population to converge.
- As the population size increases, the weight factor should decrease: more exploitation.
- Suggested \( \beta \) is 0.5

Z Variation

- The z can be of different method to select crossover points.
- Binominal (uniform)
  - The crossover points are randomly selected from the set of all possible crossover points, based on a pre-defined probability \( p_c \).
- Exponential (sort like two-point)
  - Start with a randomly selected crossover point and continue to select adjacent points to add to \( T \), treating the vector as a circular array, until either \( U(0,1) > p_c \) or \( |T| = n \).
Probability of Combination

• Probability of recombination ($p_r$): it controls the number of elements of the parent $x_i(t)$ that will change. The higher the $p_r$ is, the more variation is introduced in the new population, thereby increasing diversity and exploration.

• According to [Lopez Cruz, van Willigenburg and van Straten, 2003], increasing $p_r$ often results in faster convergence, while decreasing $p_r$ increases search robustness, i.e. the population is less likely to converge prematurely.

Useful Links

• [http://www.icsi.berkeley.edu/~storn/code.html](http://www.icsi.berkeley.edu/~storn/code.html)