Nature-Inspired Computing

Differential Evolution

Dr. Şima Uyar
September 2007

Overview

- Ken Price and Rainer Storn
- 1994 – 1996
- to solve the Chebychev polynomial fitting problem
- population based
- stochastic
- function minimization

Algorithm

```
algorithm DE:
{
  do {
    for all individuals:
    {
      choose target and base vectors;
      randomly choose 2 pop. members;
      compute weighted difference vector;
      add to base vector;
      select between trial vector or old vector;
    }
  } while not_finished;
}
```

Parameters

- NP: population size (minimum 4)
- F: mutation factor (usually in [0,2]
  - e.g. F=0.8
- CR: crossover rate (in [0,1])
  - e.g. CR=0.3
- F is a more sensitive parameter
- higher CR fastens convergence if it occurs

Operators

solution vector:

$x_{i,G} = [x_{1,i,G}, \ldots, x_{D,i,G}]$ i=1,2,...,N

N: size of population (min 4)
G: generation no.
D: parameter count

operators:

- initialization
- mutation
- crossover
- selection

Initialization

- usually random
- parameters have upper and lower bounds

Mutation

- all undergo mutation

```
for each vector x_{i,G}
  randomly select 3 distinct vectors –
         x_{r1,G}, x_{r2,G}, x_{r3,G}
  v_{i,G+1}= x_{r1,G} + F(x_{r2,G} - x_{r3,G}) //method DE/rand/1
```

$v_{i,G+1}$ is called the donor vector

Mutation Schemes - 1

- DE/rand/1

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G})$$

Mutation Schemes - 2

- DE/best/1

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} - x_{r2,G})$$

Mutation Schemes - 3

- DE/best/2

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G})$$

Mutation Schemes - 4

- DE/rand-to best/1

$$v_{i,G+1} = x_{i,G} + \lambda(x_{best,G} - x_{i,G}) + F(x_{r1,G} - x_{r2,G})$$

λ controls the greediness of the scheme and is usually set as λ =F

Crossover - 1

- trial vector $u_{i,G+1}$ generated from $x_{i,G}$ and $v_{i,G+1}$

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{for } j=\langle n\rangle_D, \langle n+1\rangle_D, \ldots, \langle n+L-1\rangle_D \\ x_{ji,G} & \text{otherwise} \end{cases}$$

Crossover - 2

- n and L values decided for each trial vector
- starting values of n chosen randomly in [1,D]
- L is drawn from [0,D-1] with probability $Pr[L=v]=[CR^v]$ $(v>0)$
  - to determine L:

```
L=0;
do {
  L=L+1;
} while ((rand()<CR) AND (L<D));
```

Selection

- $u_{i,G+1}$ is compared to $x_{i,G}$ and the one with the lower objective function value is taken to be member of the next generation G+1

Rules for usage of DE - 1

- at initialization, the population should be well spread out over objective function surface
- usually $CR \in [0,1]$ should be low (e.g. CR=0.3)
  - if no convergence occurs, $CR \in [0.8,1]$ helps
- usually NP=10*D
- usually $F \in [0.5,1]$
- for higher NP values, F should be chosen lower

Rules for usage of DE - 2

- if parameters of the best population members change a lot from generation to generation (even though change in objective function may be slow) $\Rightarrow$ good sign of convergence
- if objective function value of best exhibits plateaus:
  - minimization may take long
  - larger NP may be useful for convergence

Rules for usage of DE - 3

- objective function values of best should not drop too fast or optimization may get stuck at local optima
- choice and design of objective function is crucial