Real coded Genetic Algorithms

Prof. (Dr.) Rajib Kumar Bhattacharjya

Professor, Department of Civil Engineering

Indian Institute of Technology Guwahati, India

Room No. 005, M Block

Email: rkbc@iitg.ernet.in, Ph. No 2428

16 September 2015

Real coded Genetic Algorithms

- Disadvantage of binary coded GA
  - more computation
  - lower accuracy
  - longer computing time
  - solution space discontinuity
  - hamming cliff

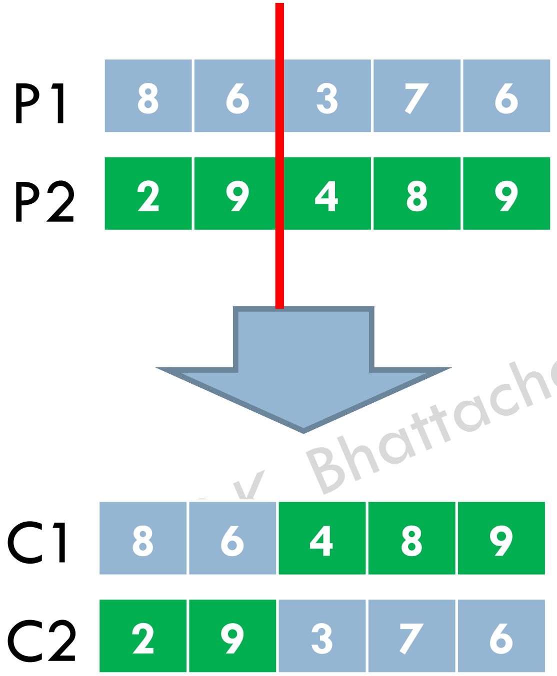
Real coded Genetic Algorithms

- The standard genetic algorithms has the following steps
  1. Choose initial population
  2. Assign a fitness function
  3. Perform elitism
  4. Perform selection
  5. Perform crossover
  6. Perform mutation

- In case of standard Genetic Algorithms, steps 5 and 6 require bitwise manipulation.

Real coded Genetic Algorithms

Simple crossover: similar to binary crossover



Real coded Genetic Algorithms

Linear Crossover

- Parents: **(**$x_1,\ldots,x_n$ **)** and **(**$y_1,\ldots,y_n$ **)**
- Select a single gene (*k*) at random
- Three children are created as,

$$(x_1, ..., x_k, 0.5 \cdot y_k + 0.5 \cdot x_k, ..., x_n)$$

$$(x_1, ..., x_k, 1.5 \cdot y_k - 0.5 \cdot x_k, ..., x_n)$$

$$(x_1, ..., x_k, \text{-}0.5 \cdot y_k + 1.5 \cdot x_k, ..., x_n)$$

- From the three children, best two are selected for the next generation

Real coded Genetic Algorithms

Single arithmetic crossover

- Parents: **(**$x_1, \ldots, x_n$ **)** and **(**$y_1, \ldots, y_n$ **)**
- Select a single gene (*k*) at random
- child_1 is created as,

$$(x_1, \ldots, x_k, \alpha \cdot y_k + (1-\alpha) \cdot x_k, \ldots, x_n)$$

- reverse for other child. e.g. with $\alpha = 0.5$

0.5	0.7	0.7	0.5	0.2	0.8	0.3	0.9	0.4
0.1	0.3	0.1	0.3	0.7	0.2	0.5	0.1	0.2

0.5	0.7	0.7	0.5	0.2	0.5	0.3	0.9	0.4
0.1	0.3	0.1	0.3	0.7	0.5	0.5	0.1	0.2

Real coded Genetic Algorithms

Simple arithmetic crossover

- Parents: $(x_1,\ldots,x_n)$ and $(y_1,\ldots,y_n)$
- Pick random gene *(k)* after this point mix values
- child$_1$ is created as:

$$(x_1, \ldots, x_k, \alpha \cdot y_{k+1} + (1-\alpha)\cdot x_{k+1}, \ldots, \alpha \cdot y_n + (1-\alpha)\cdot x_n)$$

- reverse for other child. e.g. with $\alpha = 0.5$

0.5	0.7	0.7	0.5	0.2	0.8	0.3	0.9	0.4
0.1	0.3	0.1	0.3	0.7	0.2	0.5	0.1	0.2

0.5	0.7	0.7	0.5	0.2	0.5	0.4	0.5	0.3
0.1	0.3	0.1	0.3	0.7	0.5	0.4	0.5	0.3

Real coded Genetic Algorithms

R.K. Bhattacharjya/CE/IITG

Whole arithmetic crossover

- Most commonly used
- Parents: **(**$x_1,\ldots,x_n$ **)** and **(**$y_1,\ldots,y_n$ **)**
- child$_1$ is:

$$\alpha \cdot \bar{x} + (1-\alpha) \cdot \bar{y}$$

- reverse for other child. e.g. with α = 0.5

0.5	0.7	0.7	0.5	0.2	0.8	0.3	0.9	0.4
0.1	0.3	0.1	0.3	0.6	0.2	0.5	0.1	0.2

0.3	0.5	0.4	0.4	0.4	0.5	0.4	0.5	0.3
0.3	0.5	0.4	0.4	0.4	0.5	0.4	0.5	0.3

Simulated binary crossover

- Developed by Deb and Agrawal, 1995)

$$x_i^{(1,t+1)} = 0.5\left[\left(1+\beta_{qi}\right)x_i^{(1,t)} + \left(1-\beta_{qi}\right)x_i^{(2,t)}\right]$$

$$x_i^{(2,t+1)} = 0.5\left[\left(1-\beta_{qi}\right)x_i^{(1,t)} + \left(1+\beta_{qi}\right)x_i^{(2,t)}\right]$$

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{n_c+1}} , & if\ u_i \le 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{n_c+1}} , & otherwise \end{cases}$$

Where, u_i a random number

n_c is a parameter that controls the crossover process. A high value of the parameter will create near-parent solution

Random mutation

$$y_i^{(1,t+1)} = u_i\left(x_i^u - x_i^l\right)$$

Where u_i is a random number between [0,1]

$$y_i^{(1,t+1)} = x_i^{1,t+1} + (u_i - 0.5)\Delta_i$$

Where, Δ_i is the user defined maximum perturbation

Normally distributed mutation

A simple and popular method

$$y_i^{(1,t+1)} = x_i^{1,t+1} + N(0, \sigma_i)$$

Where $N(0, \sigma_i)$ is the Gaussian probability distribution with zero mean

Polynomial mutation

Deb and Goyal,1996 proposed

$$y_i^{1,t+1} = x_i^{1,t+1} + \left(x_i^u - x_i^l\right)\delta_i$$

$$\delta_i = \begin{cases} (2u_i)^{1/(\eta_m+1)} - 1 & \text{If } u_i < 0.5 \\ 1 - \left(2(1-u_i)\right)^{1/(\eta_m+1)} & \text{If } u_i \geq 0.5 \end{cases}$$

THANKS