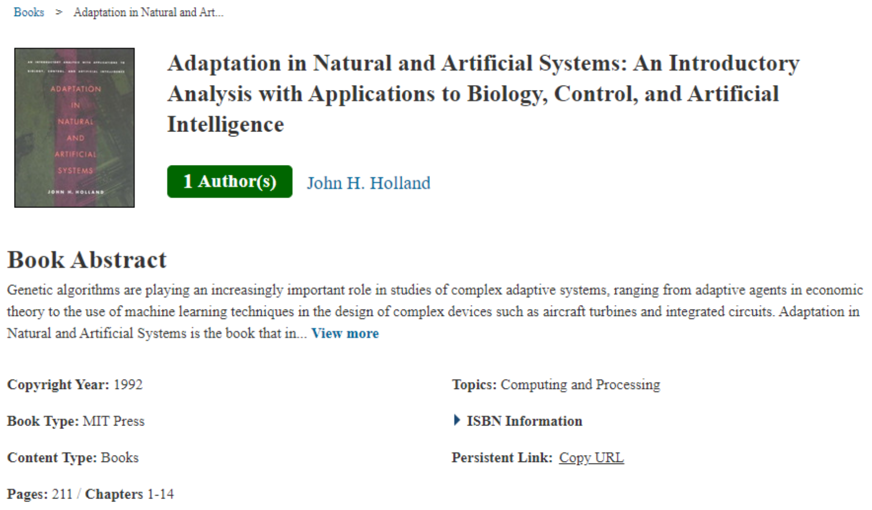
Genetic Algorithm

Prakash Kotecha
Debasis Maharana & Remya Kommadath
Department of Chemical Engineering
Indian Institute of Technology Guwahati

Outline

- Genetic algorithm
- Binary-coded Genetic Algorithm
- Working of binary GA with an example
- Drawbacks of binary GA
- Real-coded Genetic Algorithm
- Working of real GA with an example
- Comparison of algorithms

Genetic Algorithm



Books > Adaptation in Natural and Art...

Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence

1 Author(s) John H. Holland

Book Abstract

Genetic algorithms are playing an increasingly important role in studies of complex adaptive systems, ranging from adaptive agents in economic theory to the use of machine learning techniques in the design of complex devices such as aircraft turbines and integrated circuits. Adaptation in Natural and Artificial Systems is the book that in... View more

Copyright Year: 1992

Book Type: MIT Press

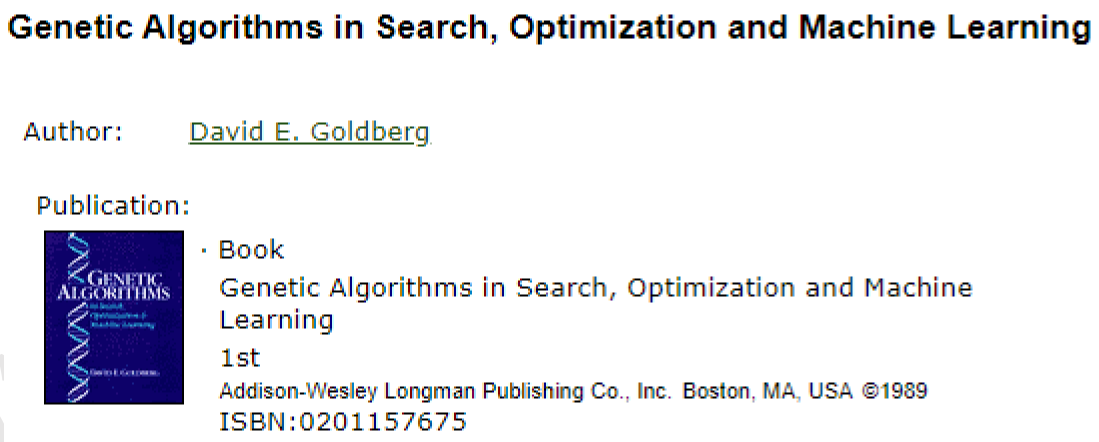
Content Type: Books

Pages: 211 / **Chapters** 1-14

Topics: Computing and Processing

▸ **ISBN Information**

Persistent Link: Copy URL



Genetic Algorithms in Search, Optimization and Machine Learning

Author: David E. Goldberg

Publication:

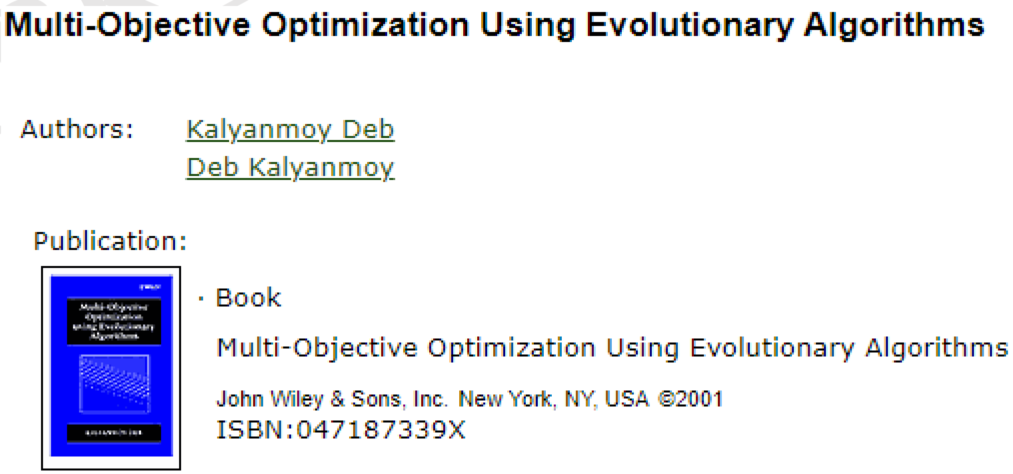
· Book

Genetic Algorithms in Search, Optimization and Machine Learning

1st

Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1989

ISBN:0201157675



Multi-Objective Optimization Using Evolutionary Algorithms

Authors: Kalyanmoy Deb
Deb Kalyanmoy

Publication:

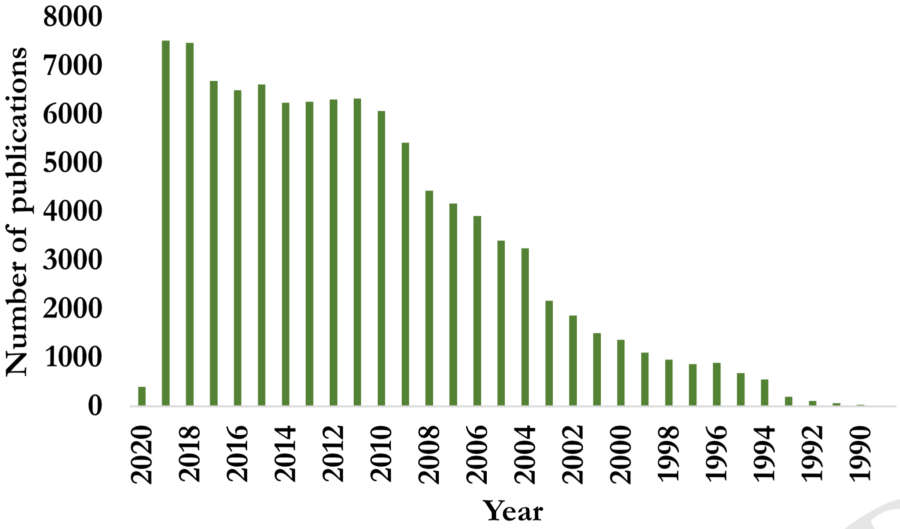
· Book

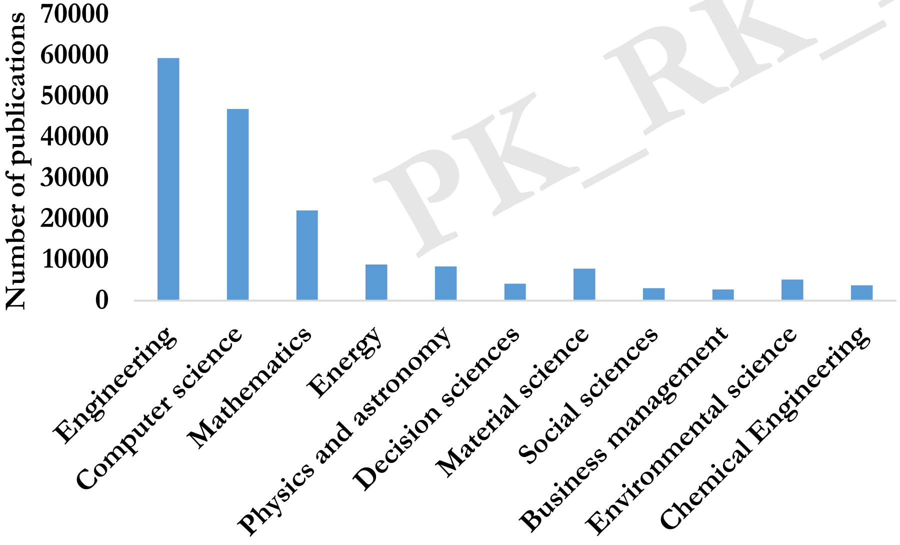
Multi-Objective Optimization Using Evolutionary Algorithms

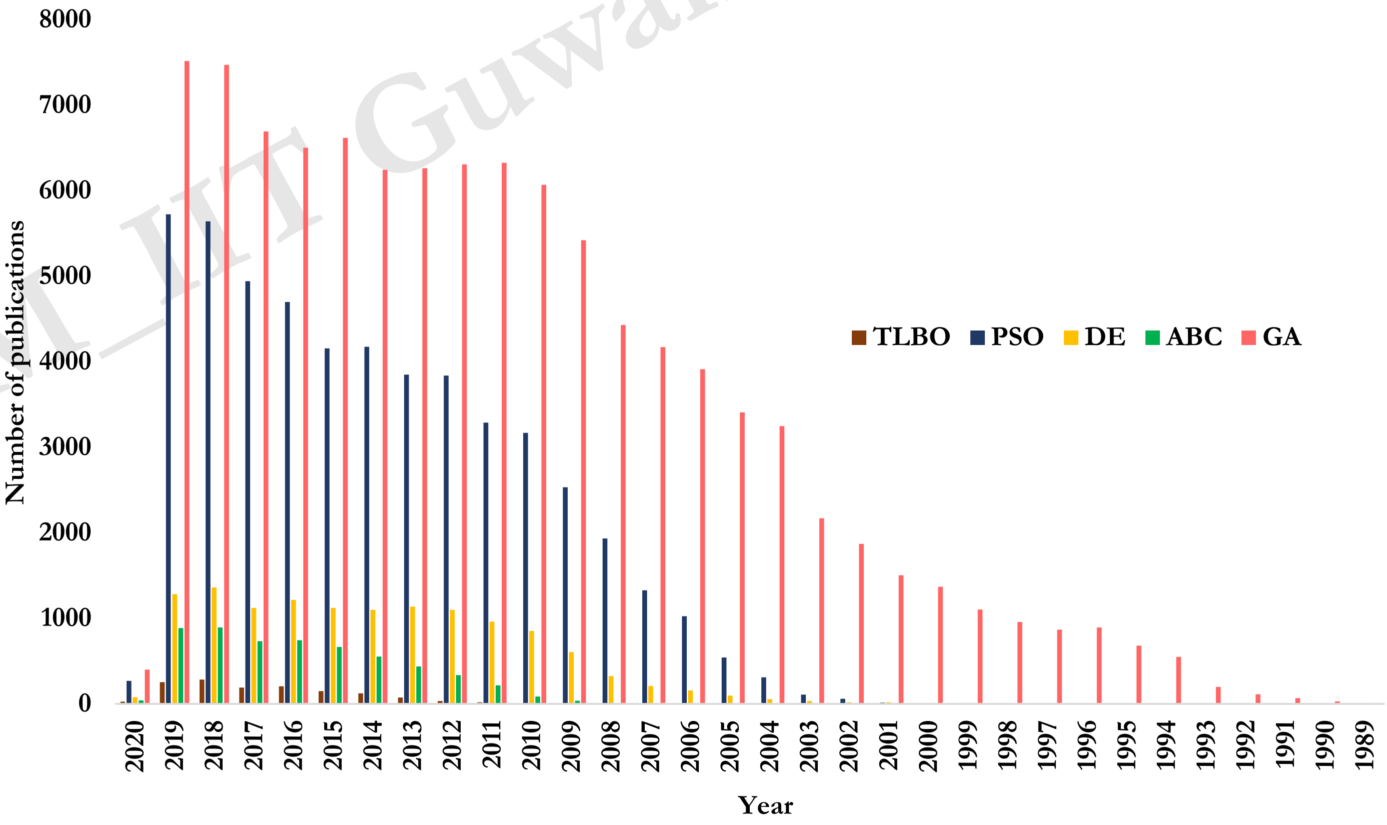
John Wiley & Sons, Inc. New York, NY, USA ©2001

ISBN:047187339X

Genetic Algorithm







Genetic Algorithm (GA)

- Inspired by the principles of natural genetics and selection (Darwin's principle of natural evolution).

- Solution vectors are termed as chromosome.

- Parent: Solution from which new solutions are generated.

- Offspring: Newly generated solutions.

- Offspring are generated through
  - reproduction (selection of good solutions for mating).
  - variation (crossover and mutation).

- Selection of good solutions after variation operator.

- Better candidate has more chance to survive in an environment of limited resources.
  - Good solutions are retained and bad solutions are eliminated.

Genetic Algorithms in search, optimization and machine learning, Addison-Wesley publishing company, 1989

Binary coded GA

- ➢Real variables are encoded into binary variables (0 and 1).
- ➢If the bit length (n) is specified for a variable, then exactly 2^n different solutions are possible between the bounds of the variable.

Let a binary string of length (n) 5 be used to represent a variable x with bounds $[x_{min}, x_{max}]$.

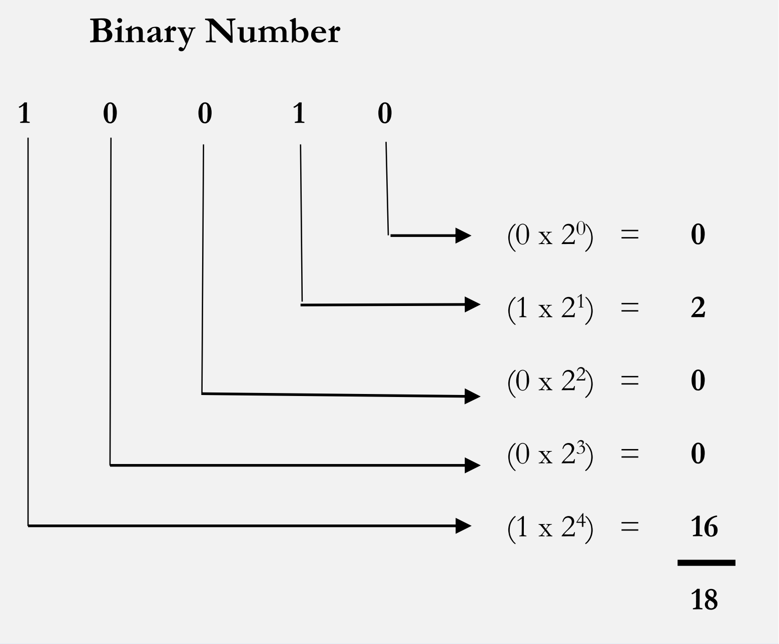
String [0 0 0 0 0] will map to x_{min} and [1 1 1 1 1] will map to x_{max}.

Decoded value of a string can be determined as

$$DV = b_n 2^{n-1} + b_{n-1} 2^{n-2} + \dots + b_2 2^1 + b_1 2^0$$

$$\text{where binary string } s = \begin{bmatrix} b_n & b_{n-1} & \dots & b_2 & b_1 \end{bmatrix}$$

Maximum decoded value of binary string is 31 (2^n - 1) and the minimum is 0.



Binary Number

1 0 0 1 0

$(0 \times 2^0) = 0$

$(1 \times 2^1) = 2$

$(0 \times 2^2) = 0$

$(0 \times 2^3) = 0$

$(1 \times 2^4) = 16$

18

Binary-coded GA

- For a fixed bit length (*n*), the value of a real variable *x* within bounds [x$_{\text{min}}$, x$_{\text{max}}$] is

$$x = x_{min} + \left(\frac{x_{max} - x_{min}}{2^n - 1}\right) DV$$

$$Precision = \frac{x_{\max} - x_{\min}}{2^n - 1}$$

- For evaluating fitness, *x* is used.

Example: Consider a variable *x* with bounds [5, 30] to be represented by a 4 bit binary string.

We will be having 16 (= 2^4) different solutions between 5 and 30.

Let the binary string be, s = [0 1 1 0], then DV = 6

(0 x 2^3) + (1 x 2^2) + (1 x 2^1) + (0 x 2^0) = 6

$$x = 5 + \left(\frac{(30-5)}{2^4 - 1}\right) 6 = 15$$

Precision = 25/15 = 1.67

For higher precision, *n* should be increased.

Binary to real values

$x_{min} = 5$, $x_{max} = 30$

$$x = x_{min} + \underbrace{\left(\frac{x_{max} - x_{min}}{2^n - 1}\right)}_{\text{precision}} DV$$

n = 3, 2^n = 8 possible solutions

$$\text{precision} = \left(\frac{30-5}{2^3-1}\right) = 3.57$$

Binary string	Decoded value	Actual value
000	0	$\boxed{5}$
001	1	$\boxed{8.57}$
010	2	12.14
011	3	15.71
100	4	19.29
101	5	22.86
110	6	26.43
111	7	30

$\boxed{5+0\times3.57}$

$\boxed{5+1\times3.57}$

n = 4, 2^n = 16 possible solutions

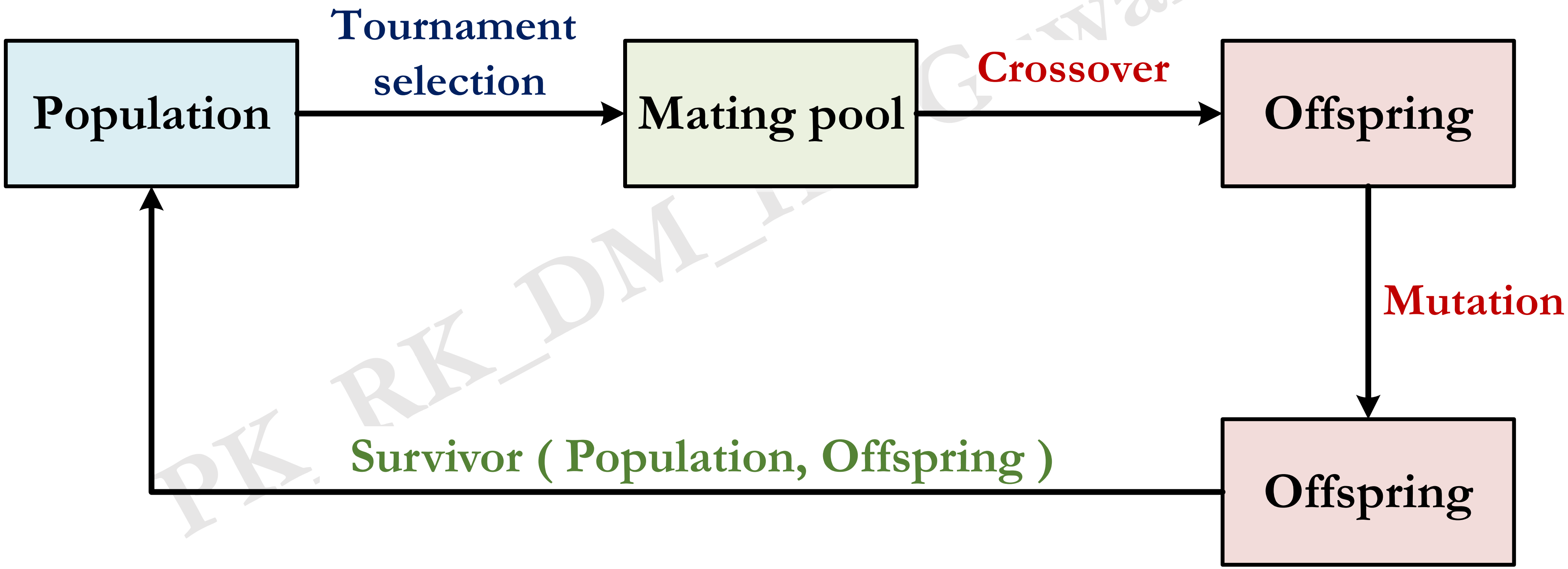
$$\text{precision} = \left(\frac{30-5}{2^4-1}\right) = 1.67$$

$\boxed{5+0\times1.67}$ $\boxed{5+1\times1.67}$

Binary string	Decoded value	Actual value	Binary string	Decoded value	Actual value
0000	0	$\boxed{5}$	1000	8	18.33
0001	1	$\boxed{6.67}$	1001	9	20
0010	2	8.33	1010	10	21.67
0011	3	10	1011	11	23.33
0100	4	11.67	1100	12	25
0101	5	13.33	1101	13	26.67
0110	6	15	1110	14	28.33
0111	7	16.67	1111	15	30

By increasing the binary string length by 1 bit, the precision is increased from **3.57** to **1.67**

GA basic flowchart



Population initialization

- ➢Let population size, $N_p = 6$ and the dimension of the problem, $D = 2$.
- ➢Let the number of bits (n) for representing each variable be 5.
- ➢Size of the population matrix will be $N_p \times nD$, (i.e., $= 6 \times 10$).
- ➢Initial population is generated randomly with 0 or 1.

Equal n

$$P = \begin{array}{ccccc|ccccc} \multicolumn{5}{c}{x_1} & \multicolumn{5}{c}{x_2} \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

Unequal n

$$P = \begin{array}{ccccccc|ccccc} \multicolumn{7}{c}{x_1} & \multicolumn{5}{c}{x_2} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

Reproduction operator

- Identify good (usually above average) solutions in the population.

- Eliminate bad solutions from the population so that multiple copies of good solutions are considered for variation operator.

- Group of solutions selected through reproduction operator comprise the mating pool.

- Reproduction: Selection of good solutions before variation operator.

- Reproduction/selection operators reduces the diversity of the population.

Tournament selection

- **Pool size:** number of members required to be in the mating pool (usually N_p).

- **Tournament size (k):** number of members that participate in a tournament (commonly k = 2).

- Tournaments are played between ‘k’ members and the member with best fitness is selected for mating.

- N_p number of tournaments has to be played.

Let k = 3, the solutions selected randomly from the population ($N_p = 6$) be {4, 2, 6} and their fitness $f_4 = 27$, $f_2 = 89$ and $f_6 = 12$.

For maximization problem: Solution 2 will win.

For minimization problem: Solution 6 will win.

Binary tournament selection

- If tournaments are played systematically, each solution will have two chances to play.
  - best solution will have two copies.
  - worst solution will never be selected.
  - other solutions can have 0, 1 or 2 copies.
- As the tournament size is increased, the selection pressure of each solution increases.

k	Remarks
2	Worst solution will not be in the mating pool
3	Worst and second worst solution will not be in the mating pool
n	The worst n–1 solutions will not be in the mating pool

Variation operator: Single point crossover

- Responsible for generating offspring.
- Two binary strings are chosen randomly from the population to perform crossover.
- Some portion of parent strings are exchanged to generate two offspring.
- Crossover site is a random integer chosen between 1 and nD.
- Crossover of two parents occurs with a probability (p_c).

Let n = 4, D = 2, crossover site = 3 and the randomly chosen parents be

$$parent_1 = \begin{bmatrix} 1 & 0 & 0 & \vert & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad offspring_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$parent_2 = \begin{bmatrix} 0 & 1 & 0 & \vert & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad offspring_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Variation operator: Bit-wise mutation

- Change the bit 1 to 0 and vice versa with a mutation probability (p_m).
- Single solution is involved in mutation.
- Every member in the population can potentially undergo mutation.
- For each bit in a string, probability for mutation is checked.

Let $p_m = 0.3$ and the parent string be

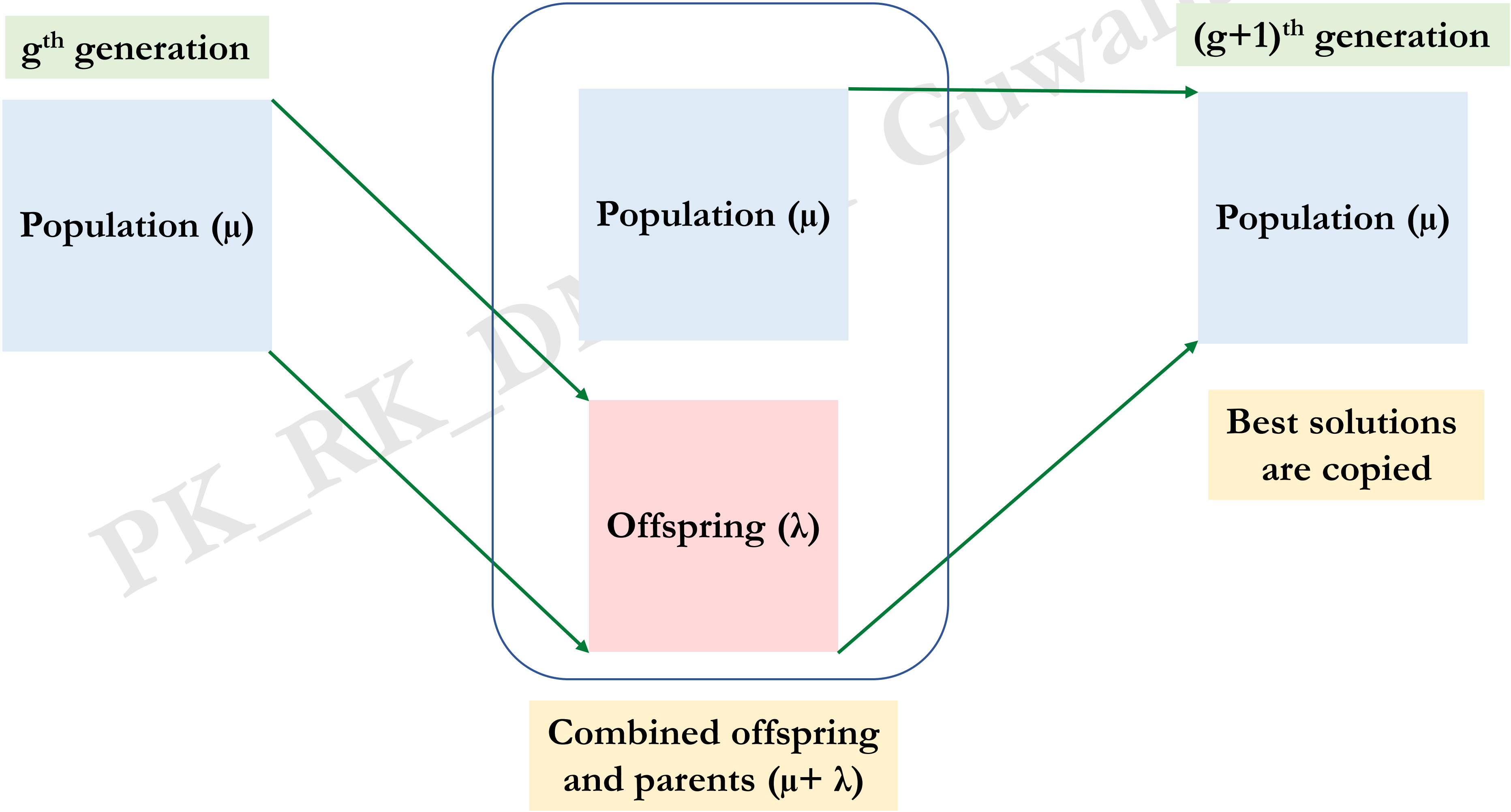
$$offspring = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Let the random number for each bit be r = [0.2 0.5 0.6 0.8 0.7 0.1 0.4 0.9].

Mutation occurs for 1st ($r_1 < p_m$) and 6th ($r_6 < p_m$) variable and the offspring is

$$offspring = \begin{bmatrix} \textcircled{0} & 0 & 1 & 0 & 1 & \textcircled{1} & 0 & 0 \end{bmatrix}$$

Survival of the fittest: (μ+ λ) selection



Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc, 2001

Working of genetic algorithm: Sphere function

Consider $\min \quad f(x)=\sum_{i=1}^{4} x_i^2; \quad 0 \le x_i \le 30, \quad i=1,2$

$f(x)=x_1^2+x_2^2$

- Decision variables: x_1 and x_2

- Step 1: Fix the population size, maximum iterations, bit length, crossover probability, mutation probability

$$N_p=4, \quad T=10, \quad n=4, \quad p_c=0.8, \quad p_m=0.3$$

$$x=x_{\min}+\frac{x_{\max}-x_{\min}}{2^n-1}(DV)$$

$$x=0+\frac{30-0}{2^4-1}(DV)=2\times DV$$

- Step 2: Generate random binary solutions

$$\begin{array}{cc} & \begin{array}{cccccccc} & & x_1 & & & & x_2 & \end{array} \\ P= & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array} \quad P_D=\begin{bmatrix} 9 & 12 \\ 3 & 7 \\ 4 & 5 \\ 6 & 10 \end{bmatrix} \quad P_A=\begin{bmatrix} 18 & 24 \\ 6 & 14 \\ 8 & 10 \\ 12 & 20 \end{bmatrix} \quad f=\begin{bmatrix} 900 \\ 232 \\ 164 \\ 544 \end{bmatrix}$$

Selection: Tournament selection

- Step 3: Select two random candidates for tournament.

Let the two candidates be

$$P_2 = \begin{bmatrix} 6 & 14 \end{bmatrix} \quad f_2 = 232$$

$$P_4 = \begin{bmatrix} 12 & 20 \end{bmatrix} \quad f_4 = 544$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$P_A = \begin{bmatrix} 18 & 24 \\ 6 & 14 \\ 8 & 10 \\ 12 & 20 \end{bmatrix} \qquad f = \begin{bmatrix} 900 \\ 232 \\ 164 \\ 544 \end{bmatrix}$$

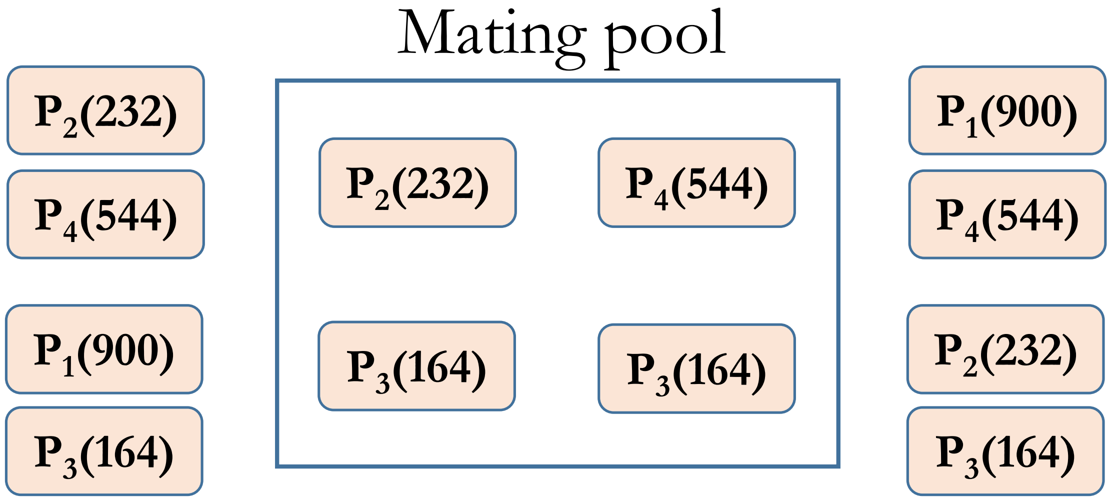
- Step 4: Compare fitness to select winner.

$$f_2 < f_4 \rightarrow \boxed{f_2}$$

- Step 5: After four tournaments

Best solution has two copies.

Worst solution has zero copies.



Crossover: Single point crossover

$p_c = 0.8$

- Step 6: Randomly select a pair of parents for crossover.

$$\text{Parent} = \begin{array}{c c} \begin{bmatrix} \boxed{0 & 0 & 1 & 1 & 0 & 1 & 1 & 1} \\ \boxed{0 & 1 & 0 & 0 & 0 & 1 & 0 & 1} \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{matrix} P_2 \\ P_3 \\ P_4 \\ P_3 \end{matrix} \end{array}$$

Let the two parents be

$$\text{Parent}_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{Parent}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

- Step 7: Generate a random number to check if crossover is to be performed.

Let $r = 0.2$

$r < p_c \rightarrow$ perform crossover

- Step 8: Select a random crossover site.

Let $r = 3$

$$\text{Parent}_1 = \begin{bmatrix} 0 & 0 & 1 & | & \boxed{1\ \ 0\ \ 1\ \ 1\ \ 1} \end{bmatrix} \qquad \text{offspring}_1 = \begin{bmatrix} 0 & 0 & 1 & | & \boxed{0\ \ 0\ \ 1\ \ 0\ \ 1} \end{bmatrix}$$

$$\text{Parent}_2 = \begin{bmatrix} 0 & 1 & 0 & | & \boxed{0\ \ 0\ \ 1\ \ 0\ \ 1} \end{bmatrix} \qquad \text{offspring}_2 = \begin{bmatrix} 0 & 1 & 0 & | & \boxed{1\ \ 0\ \ 1\ \ 1\ \ 1} \end{bmatrix}$$

Crossover: Single point crossover

$p_c = 0.8$

$$\text{Parent} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

- Step 9: Randomly select a pair of parents for crossover

Let the two parents be

$$\text{Parent}_3 = [0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$$

$$\text{Parent}_4 = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1]$$

- Step 10: Select a random number to check if crossover has to be performed

Let $r = 0.6$

$r < p_c \rightarrow$ perform crossover

- Step 11: Select a random crossover site

Let $r = 5$

$$\text{Parent}_3 = [0 \quad 1 \quad 1 \quad 0 \quad 1 \,|\, \boxed{0 \quad 1 \quad 0}] \qquad \text{offspring}_3 = [0 \quad 1 \quad 1 \quad 0 \quad 1 \,|\, \boxed{1 \quad 0 \quad 1}]$$

$$\text{Parent}_4 = [0 \quad 1 \quad 0 \quad 0 \quad 0 \,|\, \boxed{1 \quad 0 \quad 1}] \qquad \text{offspring}_4 = [0 \quad 1 \quad 0 \quad 0 \quad 0 \,|\, \boxed{0 \quad 1 \quad 0}]$$

Mutation: bit-wise mutation

- Step 12: Select first offspring for mutation

$$\text{offspring}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$p_m = 0.3$

$$\text{offspring} = O = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- Step 13: Select random numbers to check if mutation is to be performed

$$\text{Let } r = \begin{bmatrix} 0.6 & 0.1 & 0.5 & 0.4 & 0.9 & 0.7 & 0.3 & 0.8 \end{bmatrix}$$

$r < p_m \rightarrow$ perform mutation

$$\text{offspring}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \Rightarrow \text{offspring}_1 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Mutation: bit-point mutation

- Step 14: Perform mutation for all offspring

$p_m = 0.3$

	Offspring	Random number for mutation	New offspring
O_2	[0 1 0 1 0 1 1 1]	[0.5 0.1 0.6 0.3 0.5 0.4 0.6 0.3]	[0 0 0 1 0 1 1 1]
O_3	[0 1 1 0 1 1 0 1]	[0.4 0.5 0.1 0.8 0.5 0.7 0.4 0.2]	[0 1 0 0 1 1 0 0]
O_4	[0 1 0 0 0 0 1 0]	[0.8 0.6 0.3 0.9 0.7 0.5 0.4 0.6]	[0 1 0 0 0 0 1 0]

$$f\left(x\right) = x_1^2 + x_2^2$$

$$O = \begin{array}{cccccccc} & x_1 & & & & x_2 & & \\ \end{array}$$

$$O = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$x = x_{\min} + \frac{x_{\max} - x_{\min}}{2^n - 1}\left(DV\right)$$

$$\Rightarrow x = 2 \times DV$$

$$O_A = \begin{bmatrix} 12 & 10 \\ 2 & 14 \\ 8 & 24 \\ 8 & 4 \end{bmatrix}$$

$$f = \begin{bmatrix} 244 \\ 200 \\ 640 \\ 80 \end{bmatrix}$$

Survival

- Step 15: Combine all the solutions and select best N_p ($N_p = 4$) solutions

$$P = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad f = \begin{bmatrix} 900 \\ 232 \\ \boxed{164} \\ 544 \end{bmatrix}$$

$$O = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad f_O = \begin{bmatrix} 244 \\ 200 \\ 640 \\ \boxed{80} \end{bmatrix}$$

$$P_{combined} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad f_{combined} = \begin{array}{c} \begin{bmatrix} 900 \\ \boxed{232} \\ \boxed{164} \\ 544 \\ 244 \\ \boxed{200} \\ 640 \\ \boxed{80} \end{bmatrix} \end{array} \begin{array}{l} \\ \Leftarrow 4 \\ \Leftarrow 2 \\ \\ \\ \Leftarrow 3 \\ \\ \Leftarrow 1 \end{array}$$

$$P = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \quad f = \begin{bmatrix} \boxed{80} \\ 164 \\ 200 \\ 232 \end{bmatrix}$$

Population for the next iteration

Pseudocode

```
Input: Fitness function, lb, ub, $N_p$, T, n, $p_c$, $p_m$, k
1. Initialize a random population ( P ) of binary string (size: $N_p$ x nD)
2. Evaluate fitness ( f ) of P        ⬅ FE = $N_p$
  for t = 1 to T
    Perform tournament selection of tournament size, k
    for i = 1 to $N_p/2$
      Randomly choose two parents
        if $r < p_c$
          Select the crossover site
          Generate two offspring using single-point crossover
        else
          Copy the selected parents and their fitness to offspring population
        end
    end
    for i = 1 to $N_p$
        Generate D random numbers between 0 and 1
        Perform bit-wise mutation of $i^{th}$ solution in offspring population
        Evaluate the fitness of offspring        ⬅ Max FE = $N_p$
    end
    Combine population and offspring population to perform (μ + λ)
  end
```

In one iteration, max $\#FE = N_p$

For T iterations, max $\#FE = N_p + N_pT$

crossover site = 3

$$parent_1 = [1\ \ 0\ \ 0\ |\ 0\ \ 0\ \ 1\ \ 0\ \ 1]$$
$$parent_2 = [0\ \ 1\ \ 0\ |\ 1\ \ 0\ \ 1\ \ 1\ \ 1]$$
$$Offspring_1 = [1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1]$$
$$Offspring_2 = [0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1]$$

Generation

$r_6 < p_m$

$$parent = [1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0]$$
$$offspring = [1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0]$$

Survival of fittest

Drawbacks of binary GA

- Binary GA makes the search space discrete.

- Unable to achieve any arbitrary precision.

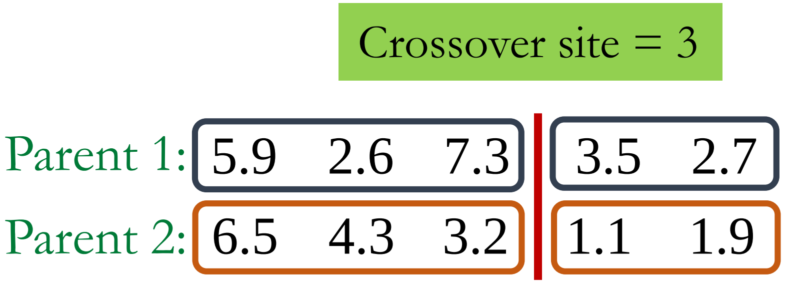
$$x = \frac{x^{max} - x^{min}}{2^n - 1}$$

  - If n bits are used to represent a decision variable, then 2^n different values are possible between the lower and upper bound of the variable.
  - To increase the precision, n has to be increased.
  - Increase in n results in larger dimension and population size.

- Hamming cliffs: transition to neighboring solution (in real space) needs change in multiple bits (Example: 01111 (= 15) to 10000 (= 16)).

Real coded Genetic Algorithm

- Encoding of real variables to binary is not required.
- Decision variables can be directly used to compute the objective function value.
- Selection operator used in binary GA can also be employed in real GA.
- Naive crossover operators such as single point crossover might fail to perform well.
  - Search within the current values of the decision variables.
  - Depends on mutation operator for a new value of decision variable.
- Modification in the variation operator is required to explore the search space.



Crossover site = 3

Parent 1: 5.9 2.6 7.3 | 3.5 2.7

Parent 2: 6.5 4.3 3.2 | 1.1 1.9

Simulated Binary Crossover (SBX)

- Simulates the single-point crossover on binary strings.

- Requires two parents to generate two offspring.

- For a fixed η_c, the offspring have a spread which is proportional to that of parents.

$$O_1 - O_2 = \beta\left(P_a^{'} - P_b^{'}\right)$$

η_c is distribution index

- Compute β

$$\beta = \begin{cases} (2u)^{1/(\eta_c+1)} & if\ u \le 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{1/(\eta_c+1)} & otherwise \end{cases}$$

- Generate offspring

$$O_a = 0.5\left[(1+\beta)P_a^{'} + (1-\beta)P_b^{'}\right]$$
$$O_b = 0.5\left[(1-\beta)P_a^{'} + (1+\beta)P_b^{'}\right]$$

$P_a^{'}$	Parent 1	O_a	Offspring 1
$P_b^{'}$	Parent 2	O_b	Offspring 2

Simulated Binary Crossover for Continuous Search Space, Complex systems, 9(2), 115-148,1995
Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc, 2001

Simulated Binary Crossover (SBX)

- Crossover is performed with high probability.
- Two offspring are symmetric about the parents.
- Avoids the bias towards any particular parent in a single crossover operation.
- For a constant β:
  - **Distant parents** result in largely spread offspring and
  - **Near parents** result in closer offspring

Case 1: Let the parents be $P_a' = 2 \; and \; P_b' = 8$

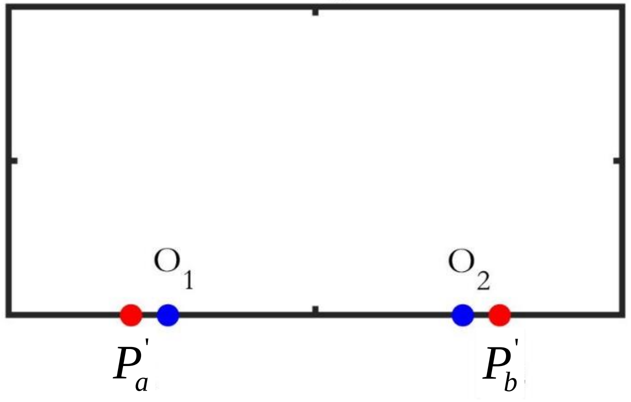
$$O_1 = 0.5\left[(1+0.8)\times 2 + (1-0.8)\times 8\right] = 2.6$$
$$O_2 = 0.5\left[(1-0.8)\times 2 + (1+0.8)\times 8\right] = 7.4$$

Case 2: Let the parents be $P_a' = 4 \; and \; P_b' = 5$

$$O_1 = 0.5\left[(1+0.8)\times 4 + (1-0.8)\times 5\right] = 4.1$$
$$O_2 = 0.5\left[(1-0.8)\times 4 + (1+0.8)\times 5\right] = 4.9$$

$$O_1 - O_2 = \beta\left(P_a' - P_b'\right)$$

Distant parent



Near Parent

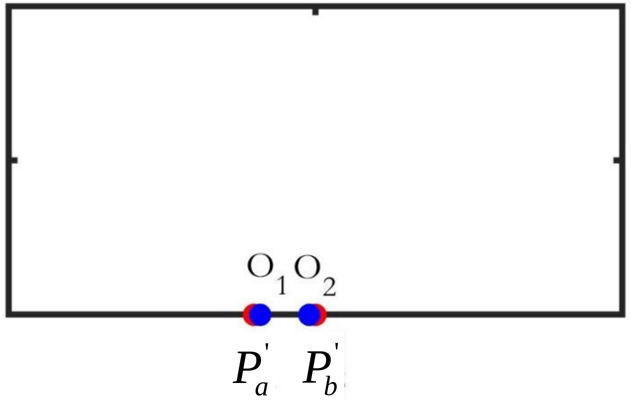


Illustration: Impact of varying β

- Consider two solutions $P_a^{'} = 2$ and $P_b^{'} = 5$

$$O_a = 0.5\left[(1+\beta)P_a^{'} + (1-\beta)P_b^{'}\right]$$
$$O_b = 0.5\left[(1-\beta)P_a^{'} + (1+\beta)P_b^{'}\right]$$

$$O_a - O_b = \beta\left(P_a^{'} - P_b^{'}\right)$$

- Case 1: Contracting crossover (β<1)

Take β = 0.6

$$O_1 = 0.5\left[(1+0.6)\times 2 + (1-0.6)\times 5\right] = 2.6$$
$$O_2 = 0.5\left[(1-0.6)\times 2 + (1+0.6)\times 5\right] = 4.4$$

Offspring are closer

- Case 2: Stationary crossover (β=1)

Take β = 1

$$O_1 = 0.5\left[(1+1)\times 2 + (1-1)\times 5\right] = 2$$
$$O_2 = 0.5\left[(1-1)\times 2 + (1+1)\times 5\right] = 5$$

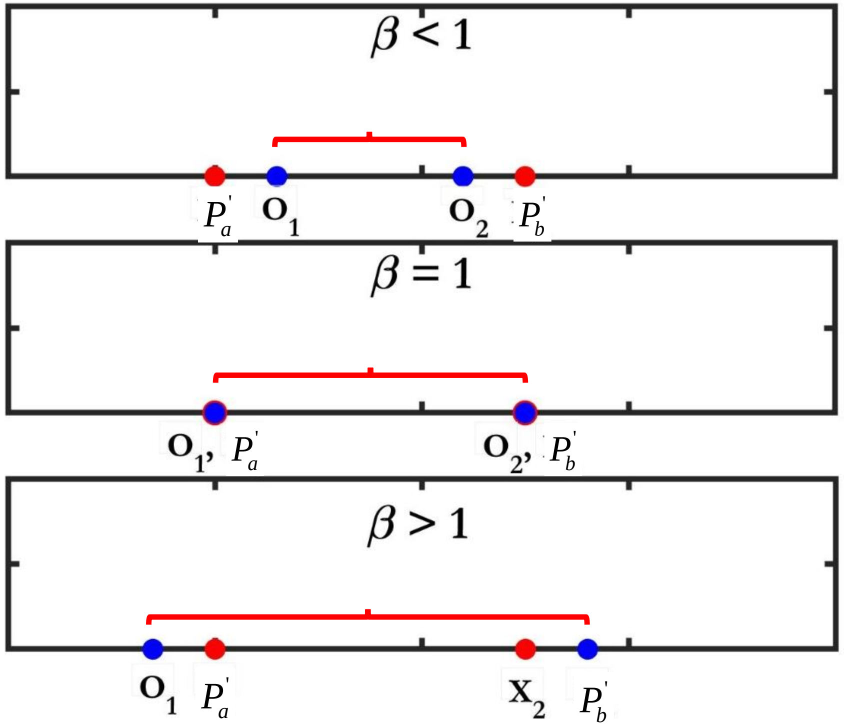
Offspring and parents are identical

- Case 3: Expanding crossover (β>1)

Take β = 1.4

$$O_1 = 0.5\left[(1+1.4)\times 2 + (1-1.4)\times 5\right] = 1.4$$
$$O_2 = 0.5\left[(1-1.4)\times 2 + (1+1.4)\times 5\right] = 5.6$$

Offspring are far apart



Polynomial Mutation

- Mutation is performed with low probability.

- Compute δ as

$$\delta = \begin{cases} (2r)^{1/(\eta_m+1)} - 1 & \text{if } r < 0.5 \\ 1 - \left[2(1-r)\right]^{1/(\eta_m+1)} & \text{if } r \geq 0.5 \end{cases}$$

η_m is distribution index

- Generate offspring

$$O = O + (ub - lb)\delta$$

O	Offspring solution
ub	upper bound
lb	lower bound

- One offspring is generated from an offspring.

Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc, 2001

Working of genetic algorithm: Sphere function

Consider $\min \quad f(x)=\sum_{i=1}^{4} x_i^2; \quad 0 \le x_i \le 10, \quad i=1,2,3,4$

Decision variables: x_1, x_2, x_3 and x_4

$$f(x)=x_1^2+x_2^2+x_3^2+x_4^2$$

- Step 1: Fix the population size, crossover probability, mutation probability, maximum iterations, distribution index for crossover and mutation

$$N_P=6,\ p_c=0.8,\ p_m=0.2, T=10, \eta_c=20, \eta_m=20$$

- Step 2: Generate random solutions within the domain of the decision variables

$$P=\begin{bmatrix}4 & 0 & 0 & 8\\ 3 & 1 & 9 & 7\\ 0 & 3 & 1 & 5\\ 2 & 1 & 4 & 9\\ 6 & 2 & 8 & 3\\ 5 & 8 & 1 & 3\end{bmatrix} \qquad f=\begin{bmatrix}80\\ 140\\ 35\\ 102\\ 113\\ 99\end{bmatrix}$$

Selection: Tournament selection

- Step 3: Select two random candidates for tournament

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \\ 5 & 8 & 1 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \\ 99 \end{bmatrix}$$

Let the two candidates be

$$P_3 = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \quad f_3 = 35$$

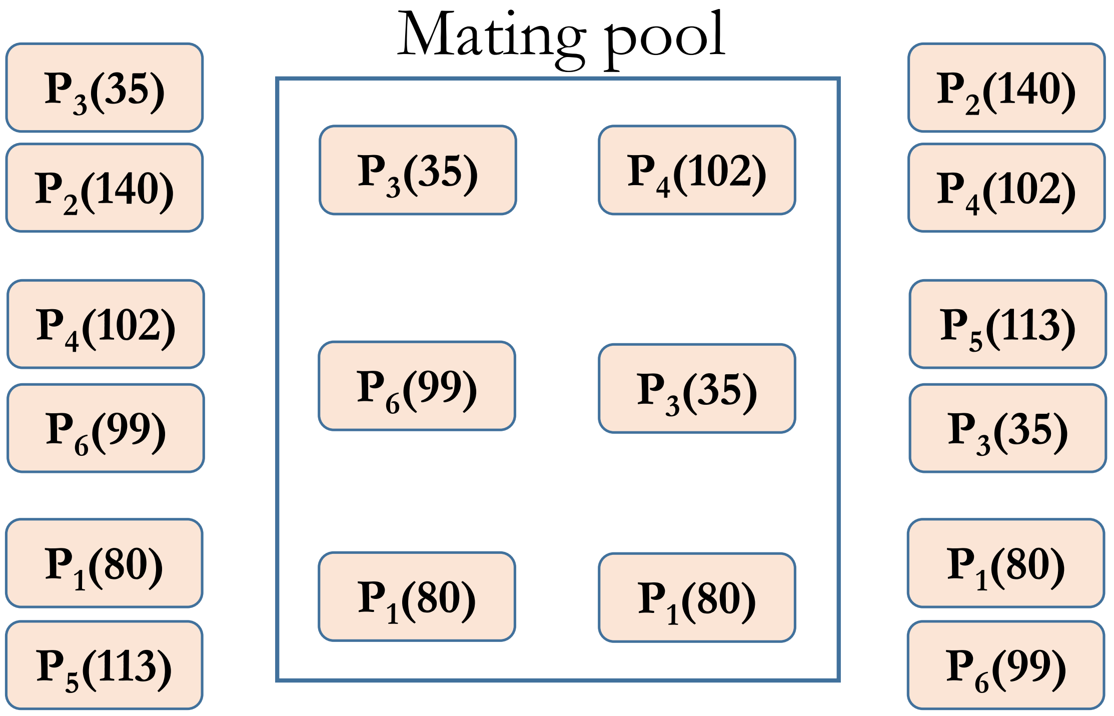
$$P_2 = \begin{bmatrix} 3 & 1 & 9 & 7 \end{bmatrix} \quad f_2 = 140$$

- Step 4: Compare fitness to select winner

$$f_3 < f_2 \rightarrow f_3$$

- Step 5: After six tournaments

Best solution has two copies
Worst solution has zero copies



Mating pool

P$_3$(35)
P$_2$(140)

P$_4$(102)
P$_6$(99)

P$_1$(80)
P$_5$(113)

P$_3$(35) P$_4$(102)

P$_6$(99) P$_3$(35)

P$_1$(80) P$_1$(80)

P$_2$(140)
P$_4$(102)

P$_5$(113)
P$_3$(35)

P$_1$(80)
P$_6$(99)

Procedure for SBX crossover

Input: P, D, p_c, η_c

1. Randomly select a pair of parents (say P_a' and P_b') from mating pool.

2. Generate a random number (r) between 0 and 1.

3. If $r \geq p_c$, then copy the parent solutions as offspring.

4. If $r < p_c$, generate D random numbers (u) for each variable

5. Determine β of each variables.

6. Generate two offspring (O_a and O_b) using

$$\beta = \begin{cases} (2u)^{1/(\eta_c+1)} & \text{if } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{1/(\eta_c+1)} & \text{otherwise} \end{cases}$$

$$O_a = 0.5\left[(1+\beta)P_a' + (1-\beta)P_b'\right]$$
$$O_b = 0.5\left[(1-\beta)P_a' + (1+\beta)P_b'\right]$$

Crossover

$p_c = 0.8, \quad \eta_c = 20$

$$\text{Parent} = \begin{bmatrix} 0 & 3 & 1 & 5 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \\ 2 & 1 & 4 & 9 \\ 0 & 3 & 1 & 5 \\ 4 & 0 & 0 & 8 \end{bmatrix}$$

- Step 6: Select first pair of parents randomly for crossover

Let the two parents be

$\text{Parent}_1 = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix}$ and $\text{Parent}_3 = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix}$

- Step 7: Select a random number to check if crossover is to be performed

Let $r = 0.2$

$r < p_c \rightarrow$ Perform crossover

- Step 8: Select a random number for each variable to perform crossover

Let $u = \begin{bmatrix} \boxed{0.2} & 0.6 & 0.1 & \boxed{0.8} \end{bmatrix}$

$\boxed{(u \le 0.5)}$ $\boxed{(u > 0.5)}$

$$\beta = (2 \times 0.2)^{1/(20+1)} = 0.96 \qquad \beta = \left(\frac{1}{2 \times (1 - 0.8)}\right)^{1/(20+1)} = 1.04$$

$$\beta = \begin{cases} (2u)^{1/(\eta_c+1)} & \text{if } u \le 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{1/(\eta_c+1)} & \text{otherwise} \end{cases}$$

Crossover

$$\beta = \begin{bmatrix} 0.96 & 1.01 & 0.93 & 1.04 \end{bmatrix}$$

- Step 9: Create two offspring

$$\boxed{O_1 = 0.5\left[\left(1+\beta\right)P_a^{'} + \left(1-\beta\right)P_b^{'}\right]}$$

$$Parent = P^{'} = \begin{bmatrix} 0 & 3 & 1 & 5 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \\ 2 & 1 & 4 & 9 \\ 0 & 3 & 1 & 5 \\ 4 & 0 & 0 & 8 \end{bmatrix}$$

$$\text{offspring}_1 = 0.5 \times \begin{pmatrix} \left(1+\begin{bmatrix} 0.96 & 1.01 & 0.93 & 1.04 \end{bmatrix}\right) \times \boxed{\begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix}} \\ \left(1-\begin{bmatrix} 0.96 & 1.01 & 0.93 & 1.04 \end{bmatrix}\right) \times \boxed{\begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix}} \end{pmatrix}$$

$$= \begin{bmatrix} 0.08 & 3.01 & 0.97 & 4.94 \end{bmatrix}$$

$$\text{offspring}_2 = 0.5 \times \begin{pmatrix} \left(1-\begin{bmatrix} 0.96 & 1.01 & 0.93 & 1.04 \end{bmatrix}\right) \times \boxed{\begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix}} \\ \left(1+\begin{bmatrix} 0.96 & 1.01 & 0.93 & 1.04 \end{bmatrix}\right) \times \boxed{\begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix}} \end{pmatrix}$$

$$\boxed{O_2 = 0.5\left[\left(1-\beta\right)P_a^{'} + \left(1+\beta\right)P_b^{'}\right]}$$

$$= \begin{bmatrix} 3.92 & -0.02 & 0.03 & 8.06 \end{bmatrix}$$

- Step 10: Check for bound violation

$$0 \le x_i \le 10$$

$$\text{offspring}_2 = \begin{bmatrix} 3.92 & \boxed{-0.02} & 0.03 & 8.06 \end{bmatrix} \rightarrow \begin{bmatrix} 3.92 & \boxed{0} & 0.03 & 8.06 \end{bmatrix}$$

No bound violation in offspring_1

Crossover

$p_c = 0.8, \quad \eta_c = 20$

- Step 11: Select second pair of parents randomly for crossover

Let the two parents be

$$\text{Parent}_2 = [5 \quad 8 \quad 1 \quad 3] \quad \text{and} \quad \text{Parent}_6 = [4 \quad 0 \quad 0 \quad 8]$$

$$\text{Parent} = \begin{bmatrix} 0 & 3 & 1 & 5 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \\ 2 & 1 & 4 & 9 \\ 0 & 3 & 1 & 5 \\ 4 & 0 & 0 & 8 \end{bmatrix}$$

- Step 12: Select a random number to check if crossover is to be performed

Let $r = 0.9$

$r > p_c \rightarrow$ no crossover

- Step 13: Copy the parents as offspring solutions

$$\text{offspring}_3 = \text{Parent}_2 = [5 \quad 8 \quad 1 \quad 3]$$

$$\text{offspring}_4 = \text{Parent}_6 = [4 \quad 0 \quad 0 \quad 8]$$

$$\text{offspring} = O = \begin{bmatrix} 0.08 & 3.01 & 0.97 & 4.94 \\ 3.92 & 0 & 0.03 & 8.06 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \end{bmatrix}$$

Crossover

$p_c = 0.8, \eta_c = 20$

$$\text{Parent} = \begin{bmatrix} 0 & 3 & 1 & 5 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \\ 2 & 1 & 4 & 9 \\ 0 & 3 & 1 & 5 \\ 4 & 0 & 0 & 8 \end{bmatrix}$$

- Step 14: Select third pair of parents randomly for crossover

Let the two parents be $\text{Parent}_4 = [2 \quad 1 \quad 4 \quad 9]$ $\quad \text{Parent}_5 = [0 \quad 3 \quad 1 \quad 5]$

- Step 15: Select a random number to check if crossover is to be performed

Let $r = 0.4$

$r < p_c \rightarrow$ Perform crossover

- Step 16: Select a random number for each variable to perform crossover

Let $u = [0.3 \quad 0.1 \quad 0.8 \quad 0.6]$

$$\beta = [\ \underbrace{\boxed{0.98 \quad 0.93}}_{}\overset{}{} \ \boxed{1.04 \quad 1.01}\ ]$$

$(u \le 0.5)$ $\quad (u > 0.5)$

$$\beta = \begin{cases} (2u)^{1/(\eta_c+1)} & if\ u \le 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{1/(\eta_c+1)} & \text{otherwise} \end{cases}$$

- Step 17: Create two offspring

$\text{offspring}_5 = [1.98 \quad 1.07 \quad 4.06 \quad 9.02]$

$\text{offspring}_6 = [0.02 \quad 2.93 \quad 0.94 \quad 4.98]$

$$O_1 = 0.5\left[(1+\beta)P_a^{'} + (1-\beta)P_b^{'}\right]$$

$$O_2 = 0.5\left[(1-\beta)P_a^{'} + (1+\beta)P_b^{'}\right]$$

Procedure for polynomial mutation

Input: P, D, p_m, η_m

1. Generate a random number (u) between 0 and 1
2. If u ≥ p_m, then no change in the offspring
3. If u < p_m, generate D random numbers (r) corresponding to each variable
4. Determine δ of each variables

$$\delta = \begin{cases} (2r)^{1/(\eta_m+1)} - 1 & \text{if } r < 0.5 \\ 1 - \left[2(1-r)\right]^{1/(\eta_m+1)} & \text{if } r \geq 0.5 \end{cases}$$

5. Modify offspring using $O = O + \left(x^u - x^l\right) \times \delta$

Mutation

- Step 18: Select first offspring for mutation

$$\text{offspring}_1 = [0.08 \quad 3.01 \quad 0.97 \quad 4.94]$$

$$O = \begin{bmatrix} 0.08 & 3.01 & 0.97 & 4.94 \\ 3.92 & 0 & 0.03 & 8.06 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \\ 1.98 & 1.07 & 4.06 & 9.02 \\ 0.02 & 2.93 & 0.94 & 4.98 \end{bmatrix}$$

- Step 19: Select a random number to check if mutation happens

Let $r = 0.1$

$r < p_m \rightarrow$ Perform mutation

- Step 20: Select a random number for each variable to perform mutation

Let $r = [\boxed{0.6} \quad 0.1 \quad \boxed{0.2} \quad 0.8]$

$p_m = 0.2, \eta_m = 20$

$(r \geq 0.5)$

$$\delta = 1 - (2 \times (1 - 0.6))^{1/(20+1)}$$
$$= 0.01$$

$(r < 0.5)$

$$\delta = (2 \times 0.2)^{1/(20+1)} - 1$$
$$= -0.04$$

$$\delta = \begin{cases} (2r)^{1/(\eta_m+1)} - 1 & \text{if } r < 0.5 \\ 1 - [2(1-r)]^{1/(\eta_m+1)} & \text{if } r \geq 0.5 \end{cases}$$

$$\delta = [0.01 \quad -0.07 \quad -0.04 \quad 0.04]$$

Mutation

$$\delta = \begin{bmatrix} 0.01 & -0.07 & -0.04 & 0.04 \end{bmatrix}$$

$$0 \le x_i \le 10, \quad i = 1,2,3,4$$

- Step 21: Generate new offspring

$$O_1 = O_1 + \left(x^u - x^l\right) \times \delta$$

$$\text{offspring}_1 = \begin{bmatrix} 0.08 & 3.01 & 0.97 & 4.94 \end{bmatrix} + \left(\begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}\right) \times \begin{bmatrix} 0.01 & -0.07 & -0.04 & 0.04 \end{bmatrix}$$

$$= \begin{bmatrix} 0.18 & 2.31 & 0.57 & 5.34 \end{bmatrix}$$

$$O = \begin{bmatrix} 0.18 & 2.31 & 0.57 & 5.34 \\ 3.92 & 0 & 0.03 & 8.06 \\ 5 & 8 & 1 & 3 \\ 4 & 0 & 0 & 8 \\ 1.98 & 1.07 & 4.06 & 9.02 \\ 0.02 & 2.93 & 0.94 & 4.98 \end{bmatrix}$$

- Step 22: Select second offspring for crossover

$$\text{offspring}_2 = \begin{bmatrix} 3.92 & 0 & 0.03 & 8.06 \end{bmatrix}$$

- Step 23: Select a random number to check if mutation happens

Let $r = 0.2$

$r = p_m \rightarrow$ No mutation

- Step 24: No change in the offspring

Mutation

- Step 25: Perform mutation for rest of the offspring

$p_m = 0.2, \eta_m = 20$

Offspring	Random number for mutation probability	Random number for mutation	New offspring
[5 8 1 3]	0.1	[0.5 0.1 0.6 0.3]	[5 7.3 1.1 2.8]
[4 0 0 8]	0.6	No mutation ($r > p_m$)	[4 0 0 8]
[1.98 1.07 4.06 9.02]	0.3	No mutation ($r > p_m$)	[1.98 1.07 4.06 9.02]
[0.02 2.93 0.94 4.98]	0.8	No mutation ($r > p_m$)	[0.02 2.93 0.94 4.98]

- Step 26: Evaluate fitness of all the offspring solutions

$$O = \begin{bmatrix} 0.18 & 2.31 & 0.57 & 5.34 \\ 3.92 & 0 & 0.03 & 8.06 \\ 5 & 7.3 & 1.1 & 2.8 \\ 4 & 0 & 0 & 8 \\ 1.98 & 1.07 & 4.06 & 9.02 \\ 0.02 & 2.93 & 0.94 & 4.98 \end{bmatrix} \qquad f_O = \begin{bmatrix} 34.21 \\ 80.33 \\ 87.34 \\ 80 \\ 102.91 \\ 34.27 \end{bmatrix}$$

Survival

- Step 27: Combine all the solutions and select best N_p ($N_p = 6$) solutions

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \\ 5 & 8 & 1 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \\ 99 \end{bmatrix}$$

$$O = \begin{bmatrix} 0.18 & 2.31 & 0.57 & 5.34 \\ 3.92 & 0 & 0.03 & 8.06 \\ 5 & 7.3 & 1.1 & 2.8 \\ 4 & 0 & 0 & 8 \\ 1.98 & 1.07 & 4.06 & 9.02 \\ 0.02 & 2.93 & 0.94 & 4.98 \end{bmatrix} \qquad f_O = \begin{bmatrix} 34.21 \\ 80.33 \\ 87.34 \\ 80 \\ 102.91 \\ 34.27 \end{bmatrix}$$

$$P_{combined} = \begin{bmatrix} 0.18 & 2.31 & 0.57 & 5.34 \\ 0.02 & 2.93 & 0.94 & 4.98 \\ 0 & 3 & 1 & 5 \\ 4 & 0 & 0 & 8 \\ 4 & 0 & 0 & 8 \\ 3.92 & 0 & 0.03 & 8.06 \\ 5 & 7.3 & 1.1 & 2.8 \\ 5 & 8 & 1 & 3 \\ 2 & 1 & 4 & 9 \\ 1.98 & 1.07 & 4.06 & 9.02 \\ 6 & 2 & 8 & 3 \\ 3 & 1 & 9 & 7 \end{bmatrix} \qquad f_{combined} = \begin{bmatrix} 34.21 \\ 34.27 \\ 35 \\ 80 \\ 80 \\ 80.33 \\ 87.34 \\ 99 \\ 102 \\ 102.91 \\ 113 \\ 140 \end{bmatrix}$$

Satisfaction of termination criterion

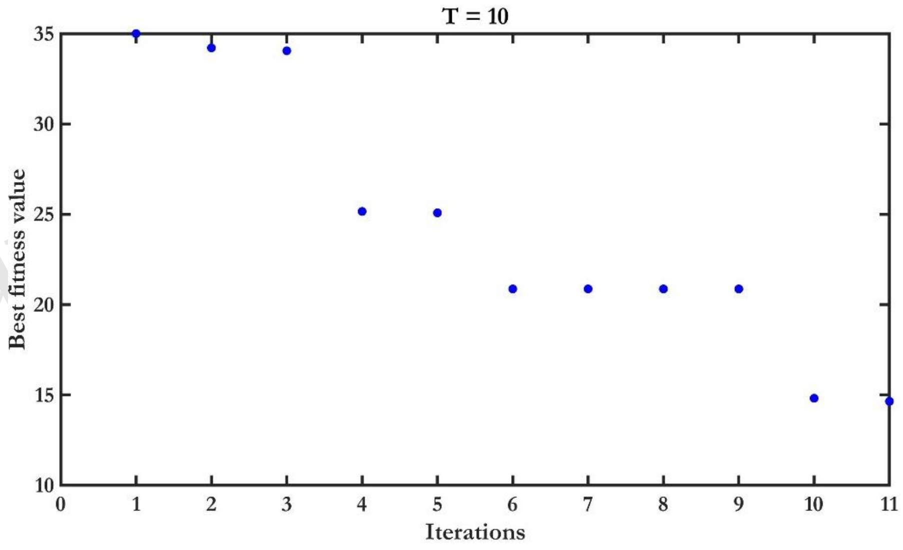
$$\min \quad f(x) = \sum_{i=1}^{4} x_i^2; \quad 0 \le x_i \le 10, \quad i = 1,2,3,4$$

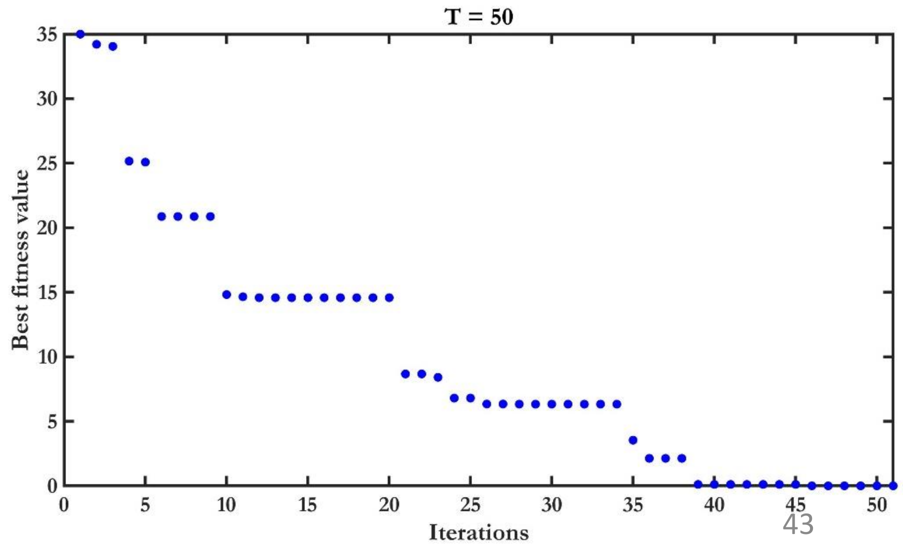
After completion of 10 iterations

$$P = \begin{bmatrix} 0.3 & 1.23 & 0.61 & 3.56 \\ 0.3 & 1.24 & 0.61 & 3.58 \\ 0.3 & 1.24 & 0.61 & 3.59 \\ 0 & 1.74 & 0.61 & 4.18 \\ 0 & 1.74 & 0.61 & 4.18 \\ 0 & 1.74 & 0.61 & 4.18 \end{bmatrix} \qquad f = \begin{bmatrix} 14.65 \\ 14.82 \\ 14.89 \\ 20.87 \\ 20.87 \\ 20.87 \end{bmatrix}$$

The minimum value of the function is **0**

The optima is achieved by increasing T





Pseudocode

```
Input: Fitness function, lb, ub, $N_p$, T, $p_c$, $p_m$, $\eta_c$, $\eta_m$, k
1.  Initialize a random population ( P )
2.  Evaluate fitness ( f ) of P                FE = $N_p$
   for t = 1 to T
      Perform tournament selection of tournament size, k
      for i = 1 to $N_p/2$
        Randomly choose two parents
            if $r < p_c$
              Generate two offspring using SBX-crossover
              Bound the offspring and store them in offspring population
            else
              Copy the selected parents and their fitness to offspring population
            end
      end
      for i = 1 to $N_p$
            if $r < p_m$
              Perform polynomial mutation of $i^{th}$ solution in offspring population
              Bound the mutated offspring
            else
              No change in $i^{th}$ solution of offspring
            end
      end
   Evaluate the fitness                        Max #FE = $N_p$
   Combine population and offspring population to perform ($\mu + \lambda$)
   end
```

In one iteration, max #FE = N_p

For T iterations, max #FE = $N_p + N_p T$

Generation

$$\beta = \begin{cases} (2u)^{1/(\eta_c+1)} & \text{if } u \le 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{1/(\eta_c+1)} & \text{otherwise} \end{cases}$$

$$O_1 = 0.5\left[(1+\beta)X_1 + (1-\beta)X_2\right]$$

$$O_2 = 0.5\left[(1-\beta)X_1 + (1+\beta)X_2\right]$$

$$\delta = \begin{cases} (2r)^{1/(\eta_m+1)} & \text{if } r < 0.5 \\ 1-\left[2(1-r)\right]^{1/(\eta_m+1)} & \text{if } r \ge 0.5 \end{cases}$$

$$y = O + (ub - lb)\delta$$

Survival of fittest

Pseudocode

BGA

```
Input: Fitness function, lb, ub, $N_p$, T, n, $p_c$, $p_m$, k
1.  Initialize a random population ( P ) of binary string (size: $N_p$ x nD)
2.  Evaluate fitness ( f ) of P
  for t = 1 to T
    Perform tournament selection of tournament size, k
    for i = 1 to $N_p/2$
      Randomly choose two parents
        if $r < p_c$
          Select the crossover site
          Generate two offspring using single-point crossover
        else
          Copy the selected parents as offspring
        end
    end
    for i = 1 to $N_p$
      Generate D random numbers between 0 and 1
      Perform bit-wise mutation of $i^{th}$ offspring
    end
    Evaluate the fitness of offspring
    Combine population and offspring population to perform $(\mu + \lambda)$
  end
```

RGA

```
Input: Fitness function, lb, ub, $N_p$, T, $p_c$, $p_m$, $\eta_c$, $\eta_m$, k
1.  Initialize a random population ( P )
2.  Evaluate fitness ( f ) of P
  for t = 1 to T
    Perform tournament selection of tournament size, k
    for i = 1 to $N_p/2$
      Randomly choose two parents
        if $r < p_c$
          Generate two offspring using SBX-crossover
          Bound the offspring and store them in offspring population
        else
          Copy the selected parents and their fitness to offspring population
        end
    end
    for i = 1 to $N_p$
        if $r < p_m$
          Perform polynomial mutation of $i^{th}$ offspring
          Bound the mutated offspring
        else
          No change in $i^{th}$ offspring
        end
    end
  Evaluate the fitness of offspring
  Combine population and offspring population to perform $(\mu + \lambda)$
  end
```

Real variables are encoded to binary in BGA
Encoding is not required in RGA

In RGA mutation probability is checked only once
In BGA, it is checked for each variable

Comparison of techniques

	TLBO	PSO	DE	ABC	GA
Phases	Teacher, Learner	No phases (Position and velocity update)	No phases (Mutation and crossover)	Employee, Onlooker and Scout	No phases (Mutation and crossover)
Convergence	Monotonic	Monotonic (with g_{best} & p_{best})	Monotonic	Monotonic (with globalized memory)	Monotonic
Parameters	Population size, termination criteria	Population size, termination criteria, w, c_1 and c_2	Population size, termination criteria, P_c, F	Population size, termination criteria, limit	Population size, termination criteria, p_c, p_m, and other parameters of variation operators (η_c and η_m)
Generation of solution	Using other solutions, mean and best solution	Using velocity vector, p_{best} and g_{best}	Using other solutions	Using other solutions	Using other solutions
Best solution	Part of population	Need not be part of population	Part of population	Need not be part of population	Part of population
Fitness function	Objective function	Objective function	Objective function	Inversely related to objective function	Objective function
Population update	Twice	Once	Once	Twice or thrice (Scout phase)	Once
Selection	Greedy	Always accept new solution into the population (μ, λ)	Greedy	Greedy and (μ, λ) (in scout phase)	Survival of the fittest (μ+λ)
#FE	$N_p + 2N_pT$	$N_p + N_pT$	$N_p + N_pT$	Max #FE = $N_p + 2N_pT + T$	Max #FE = $N_p + N_pT$

Further reading

- Genetic Algorithms in search, optimization and machine learning, Addison-Wesley publishing company, 1989

- An introduction to genetic algorithms, **Sadhana**, 24, Parts 4 & 5, 293-315, 1999

- Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons Inc., 2001

- Simulated Binary Crossover for Continuous Search Space, **Complex systems**, 9(2), 115-148, 1995

- A fast and elitist multiobjective genetic algorithm: NSGA-II, **IEEE Transactions on Evolutionary Computation**, 6(2), 182-197, 2002

Thank You !!!