**Schema Theorem for Binary Coded GA**

- Proposed by Prof. John Holland

- An attempt to give GA a mathematical foundation

Let us consider a population of binary-strings created at random

1 1 0 0 1 1
0 1 0 1 0 0
¦ ¦ ¦ ¦ ¦ ¦
1 1 0 1 1 1
1 1 0 0 0 0

Let us assume the following two schemata (templates):

$H_1$: * 1 0 * * *
$H_2$: * 1 0 * 0 0 ( * could be either 1 or 0)

$H_1$:  * 1 0 * * *
$H_2$:  * 1 0 * 0 0

- Order of schema $O(H)$:

  No. of fixed positions (bits) present in a schema

  For example: $O(H_1) = 2$; $O(H_2) = 4$

- Defining length of schema $\delta(H)$:

  Distance between the first and last fixed positions in a string

  For example: $\delta(H_1) = 3-2 = 1$; $\delta(H_2) = 6-2 = 4$

- <u>Effect of Selection</u>:

Let m(H, t) = No. of strings belonging to schema H at $t^{th}$ Gen.

m(H, t+1) = No. of strings belonging to schema H at $(t+1)^{th}$ Gen.

$$E\,[m(H, t+1)] = m(H, t)\,\frac{f(H)}{\overline{f}}$$

f (H) = Schema fitness or Avg. fitness of the strings represented by schema H

$\overline{f}$ = Avg. fitness of the entire population at t-th Gen.

- <u>Effect of Crossover (Single-point)</u>:

Let $P_c$= Probability of crossover and

L = String length

A schema is destroyed if crossover site falls within the defining length

Probability of destruction = $p_c \dfrac{\delta(H)}{L-1}$

Probability of survival = $1 - p_c \dfrac{\delta(H)}{L-1}$

- Effect of Mutation (Bit–wise Mutation):

To protect a schema, mutation should not occur at the fixed bits

Let $p_m$ : probability of mutation

probability of destruction of a single bit = $p_m$

probability of survival of a single bit = $1- p_m$

Probability of survival of the whole schema,

$$p_s = (1- p_m)\ (1- p_m) \ldots\ldots\ldots O(H)$$

$$= (1- p_m)^{O(H)}$$

$$= 1- O(H)\, p_m \qquad \text{as } p_m \ll 1$$

Considering the contributions of all three operators,

$$E\left[m(H,t+1)\right] = m(H,t)\,\frac{f(H)}{\bar{f}}\left[1 - p_c\,\frac{\delta(H)}{L-1} - O(H)\,p_m\right] \quad \text{(neglecting 2}^{\text{nd}}\text{ order term)}$$

**Building-Block Hypothesis:**

The schemata having low order, short defining length and fitness considerably more than average fitness of the population will have more and more representations in future generations

## Limitations of Binary Coded GA

- Unable to yield any arbitrary precision in the solution → Real Coded GA

- Hamming Cliff problem → creates an artificial hindrance to the gradual search of GA → Gray Coded GA

| | | |
|---|---|---|
| 14 : | 0 1 1 1 0 | |
| | | 1 change |
| 15 : | 0 1 1 1 1 | |
| | | 5 changes |
| 16 : | 1 0 0 0 0 | |

**Real Coded GA:**

Chromosome:

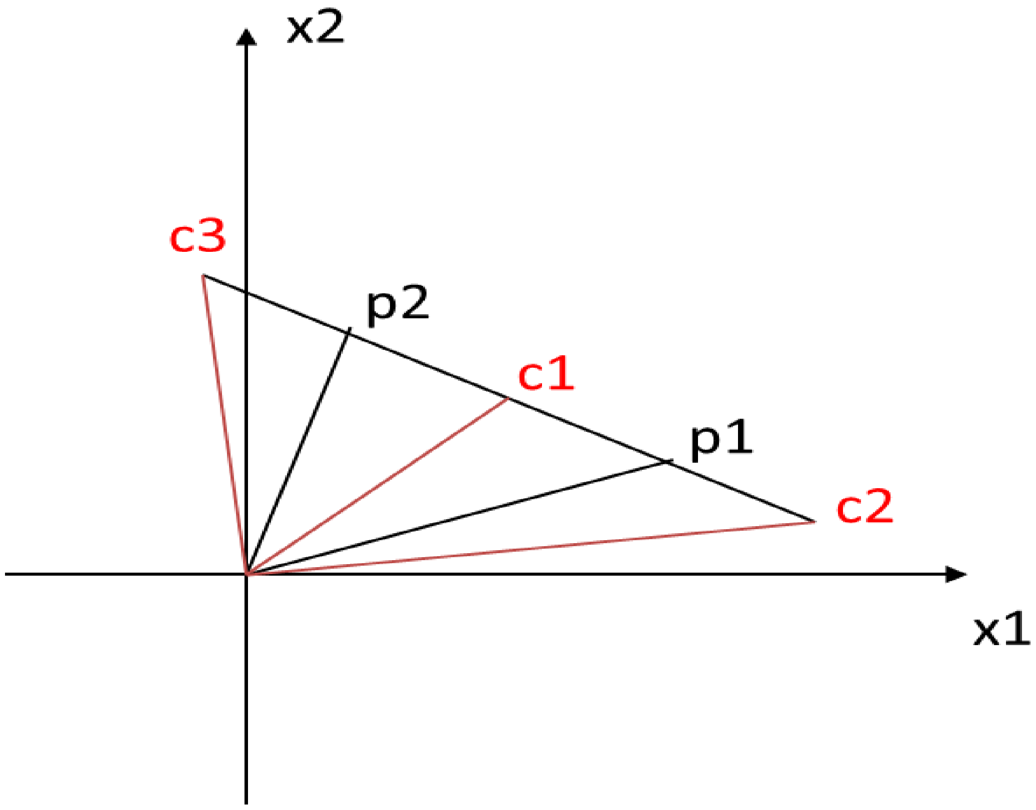
| x1 | x2 | x3 | x4 | x5 | x6 |
|---|---|---|---|---|---|
| 5.82 | 1.10 | 9.22 | 3.61 | 8.30 | 2.99 |

- ✓ Selection: Same as Binary Coded GA

- ✓ Crossover: <u>Single point</u>

  <u>Linear Crossover</u>

$$\left.\begin{aligned} Ch1 &= 0.5*Pr1+0.5*Pr2 \\ Ch2 &= 1.5*Pr1-0.5*Pr2 \\ Ch3 &= -0.5*Pr1+1.5*Pr2 \end{aligned}\right\} \text{Best Two}$$




$$C2, C3 = \frac{P1 + P2}{2} \pm (P1 - P2)$$

## Blend Crossover

$$Ch1 = (1-\gamma)*Pr1+ \gamma *Pr2$$

$$Ch2 = \gamma *Pr1+(1- \gamma)*Pr2$$

$\gamma = 2*r – 0.5$ where r is a uniform random number in [0,1]

i.e. $\gamma$ is a uniform random number in [-0.5,1.5]

## Simulated Binary Crossover

✓ Mutation: Replace a value with a random value in the entire range or in a random neighbourhood
(neighbourhood may shrink as generation increases)

- Mutation probability in RCGA is more than that in BCGA

Let $m$ be the string length for a variable $x1$ in BCGA

Probability that this variable survives mutation is $(1 - P_m)^m$

$$\approx (1 - mP_m)$$

Hence, $(1 - P_m^R) = (1 - mP_m^B)$

$$P_m^R = m\, P_m^B$$

**Numerical Example:**

maximize $f(x_1, x_2) = - x_1^2 - 2x_2^2 - x_1x_2$ ; $-5 \leq x_1, x_2 \leq 5.0$

Assume the mating pool to be (4,0); (-2,2); (3,1); (3,1)

Consider mating pairs as 1-3 and 2-4

Obtain the next generation by linear crossover.

Assume Pc = 1.0 and Pm = 0.

**Answer:**

(3.5, 0.5); (2.5, 1.5); (0.5, 1.5); (-4.5, 2.5)

## **Constraints Handling in GA** (also applicable to PSO)

optimize $f(\underline{x})$

subject to

$$g_j(\underline{x}) \leq 0 \;,\; j = 1,2,\ldots,m$$

$$h_k(\underline{x}) = 0 \;,\; k = 1,2,\ldots,p$$

$$\underline{x} = [x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T$$

$$\underline{x}_{min} \leq \underline{x} \leq \underline{x}_{max}$$

Let $m+p = q$

Functional constraints

$$\Phi_k(\underline{x}) \;,\; k = 1,2,\ldots,q$$

## Penalty Function Approach

Fitness function of $i^{th}$ solution

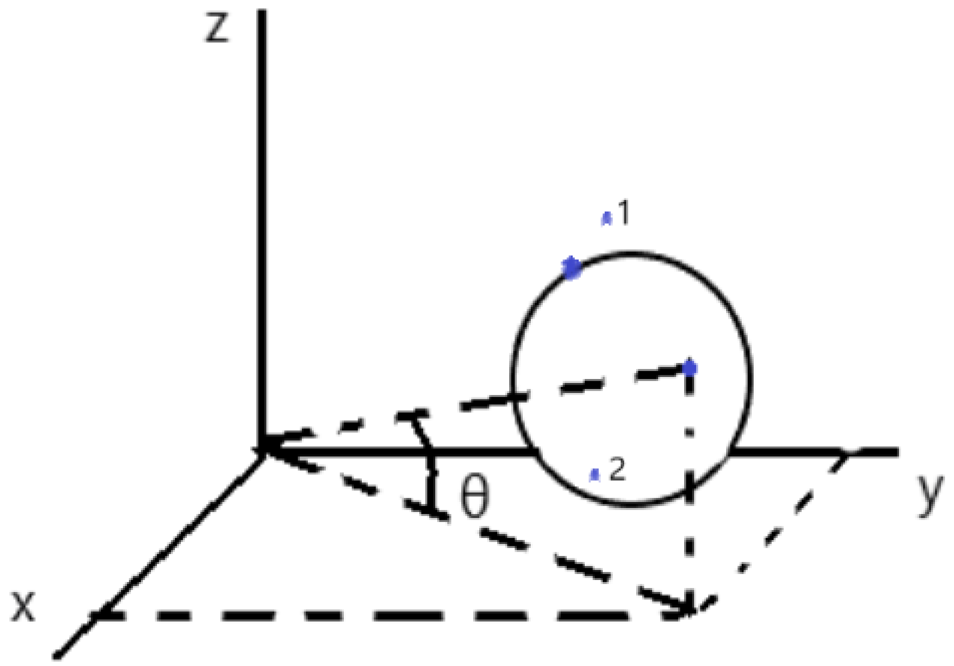
$F_i(X) = f_i(X) \pm P_i$ (+ for minimization problems)

where $P_i$ indicates penalty given by

$$P_i = C\sum_{k=1}^{q}\left\{\varphi_{ik}(X)\right\}^2$$

C indicates penalty coefficient

**Example:**



## Static Penalty

Fitness of i-th solution

$$F_i(X) = f_i(X) + \sum_{k=1}^{q} C_{k,r} \{\varphi_{ik}(X)\}^2$$

where $C_{k,r}$ : $r^{th}$ level violation of $k^{th}$ constraint

(amount of violation is divided into various pre-defined levels)

## Dynamic Penalty

Fitness
$$F_i(X) = f_i(X) + (C.t)^{\alpha} \sum_{k=1}^{q} \left|\varphi_{ik}(X)\right|^{\beta}$$

where C, α, β are user-defined constants

t = number of generations

✓ Penalty increasing with generation number (pressurizing GA)

## Adaptive Penalty

Fitness $$F_i(X) = f_i(X) + \lambda(t)\sum_{k=1}^{q}\{\phi_{ik}(X)\}^2$$

where t : number of generations

$$\lambda(t+1) = \begin{cases} \frac{1}{\beta_1}.\lambda(t), & \text{if best soln. of last } N_f \text{ GEN were feasible} \\ \beta_2.\lambda(t), & \text{if infeasible} \end{cases}$$

if neither, $\lambda(t+1) = \lambda(t)$ ( where $\beta_1 \neq \beta_2$ and $\beta_1, \beta_2 > 1$ )