# Outline

1. **RGA Operators**
   - Selection Operators
   - Crossover Operators
   - Mutation Operators
2. **Simulations**
   - Rosenbrock Function
   - Himmelblau Function
   - Rastrigin Function
   - Ackley Function
3. **Closure**

# Recap of RGA

- Limitations of BGA for solving problems with continuous search space
- EC Techniques for real-parameter optimization
- Generalized framework for RGA
- Working principles of RGA through an example
  - Initial population in which parameters are coded as real numbers
  - Selection operator: No change in its functioning was observed with respect to BGA because it used the fitness values.
  - Crossover operator: Properties of single-point crossover operator, SBX crossover operator
  - Polynomial mutation operator
  - $(\mu + \lambda)$–strategy for survival stage
- Hand calculations for one generation
- Graphical illustration of RGA for one generation

# Selection Operators

- Selection operators need fitness value of each solution in the population
- Fitness proportionate selection operator
- Tournament selection operator
- Ranking methods
- $(\mu + \lambda)-$ strategy and $(\mu, \lambda)-$ strategy

# Linear Crossover Operator

- Crossover operators for real parameters are also called as blending operators.

## Linear crossover operator

One of the earliest implementations by Wright (1991):

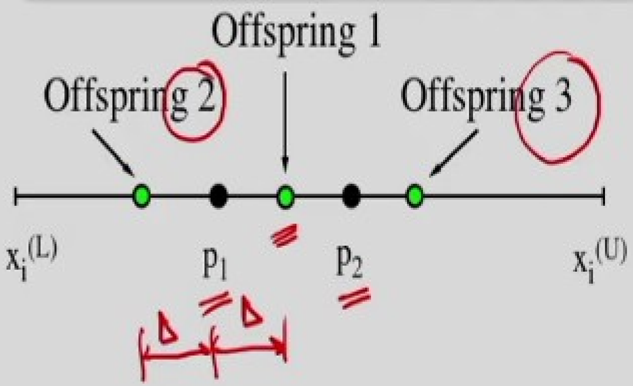
Parent 1: $p_1 = x_i^{(1,t)}$

Parent 2: $p_2 = x_i^{(2,t)}$

Offspring 1: $0.5(x_i^{(1,t)} + x_i^{(2,t)})$

Offspring 2: $(1.5x_i^{(1,t)} - 0.5x_i^{(2,t)})$

Offspring 3: $(-0.5x_i^{(1,t)} + 1.5x_i^{(2,t)})$




Here, $x_i^{(1,t)}$ represents $i$–th decision variable of the randomly chosen first parent in the $t$–th generation.

# Linear Crossover Operator

- Crossover operators for real parameters are also called as blending operators.

## Linear crossover operator

One of the earliest implementations by Wright (1991):

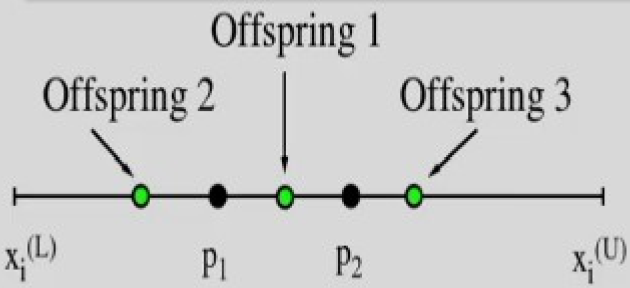
Parent 1: $p_1 = x_i^{(1,t)}$

Parent 2: $p_2 = x_i^{(2,t)}$

Offspring 1: $0.5(x_i^{(1,t)} + x_i^{(2,t)})$ 1

Offspring 2: $(1.5x_i^{(1,t)} - 0.5x_i^{(2,t)})$ 2

Offspring 3: $(-0.5x_i^{(1,t)} + 1.5x_i^{(2,t)})$ 3



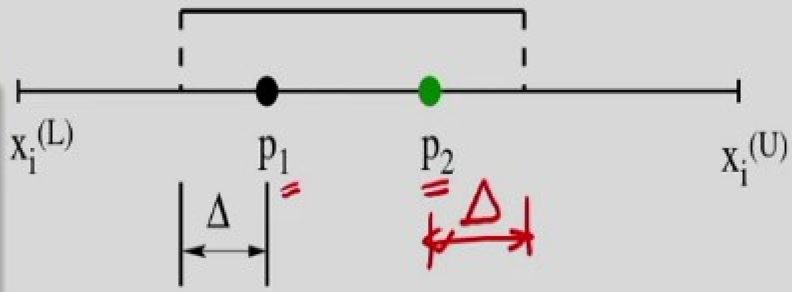

Here, $x_i^{(1,t)}$ represents $i-$th decision variable of the randomly chosen first parent in the $t-$th generation.

- We choose two best offspring solutions among three.

# Blend Crossover Operator

### Blend Crossover and Its Variants

- Blend crossover (BLX-$\alpha$)
- The assumption is $p_1 = x_i^{(1,t)} < x_i^{(2,t)} = p_2$ for two parents.

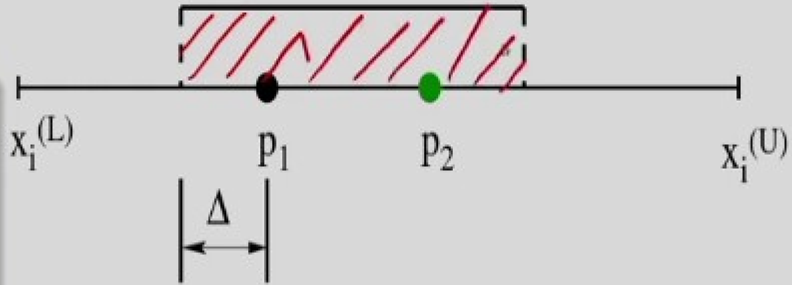



- Here, $\Delta = \alpha(p_2 - p_1) = \alpha(x_i^{(2,t)} - x_i^{(1,t)})$

# Blend Crossover Operator

**Blend Crossover and Its Variants**

- Blend crossover (BLX-$\alpha$)
- The assumption is $p_1 = x_i^{(1,t)} < x_i^{(2,t)} = p_2$ for two parents.



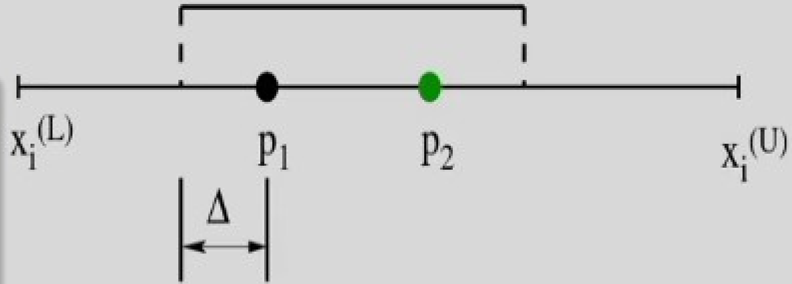

- Here, $\Delta = \alpha(p_2 - p_1) = \alpha(x_i^{(2,t)} - x_i^{(1,t)})$

- BLX-$\alpha$ randomly picks a solution in the range $[x_i^{(1,t)} - \Delta, x_i^{(2,t)} + \Delta)]$

# Blend Crossover Operator

## Blend Crossover and Its Variants

- Blend crossover (BLX-$\alpha$)
- The assumption is $p_1 = x_i^{(1,t)} < x_i^{(2,t)} = p_2$ for two parents.



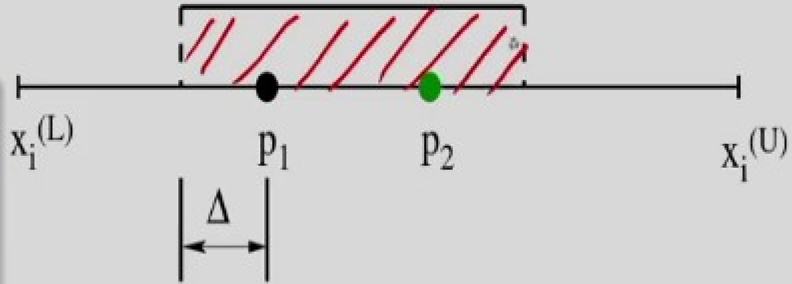

- Here, $\Delta = \alpha(p_2 - p_1) = \alpha(x_i^{(2,t)} - x_i^{(1,t)})$

- BLX-$\alpha$ randomly picks a solution in the range $[x_i^{(1,t)} - \Delta, x_i^{(2,t)} + \Delta)]$
- Offspring: $\underline{x_i^{(1,t+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)}}$, where $\gamma_i = (1+2\alpha)u_i - \alpha$ user-defined
- $u_i$ is a random number between 0 and 1.

# Blend Crossover Operator

**Blend Crossover and Its Variants**

- Blend crossover (BLX-$\alpha$)
- The assumption is $p_1 = x_i^{(1,t)} < x_i^{(2,t)} = p_2$ for two parents.




- Here, $\Delta = \alpha(p_2 - p_1) = \alpha(x_i^{(2,t)} - x_i^{(1,t)})$

- BLX-$\alpha$ randomly picks a solution in the range $[x_i^{(1,t)} - \Delta, x_i^{(2,t)} + \Delta)]$
- Offspring: $x_i^{(1,t+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)}$, where $\gamma_i = (1+2\alpha)u_i - \alpha$
- $u_i$ is a random number between 0 and 1.
- It can be seen from the figure that $\gamma_i$ is uniformly distributed for a fixed value of $\alpha$.

# Blend Crossover Operators

**BLX-$\alpha$**

- The value of $\alpha = 0.5$ is found to be performing better than other $\alpha$ values (Deb, 2001) over many test problems.
- Let us re-write the equation for BLX-$\alpha$

$$x_i^{(1,t+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)}$$

$$\left(x_i^{(1,t+1)} - x_i^{(1,t)}\right) = \gamma_i \boxed{\left(x_i^{(2,t)} - x_i^{(1,t)}\right)}$$

$p_2 - p_1$

  - It suggests that the location of offspring depends on the difference in parent solutions.

# Blend Crossover Operators

**BLX-$\alpha$**

- The value of $\alpha = 0.5$ is found to be performing better than other $\alpha$ values (Deb, 2001) over many test problems.
- Let us re-write the equation for BLX-$\alpha$

$$x_i^{(1,t+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)}$$
$$(x_i^{(1,t+1)} - x_i^{(1,t)}) = \gamma_i \boxed{(x_i^{(2,t)} - x_i^{(1,t)})}$$

  - It suggests that the location of offspring depends on the difference in parent solutions.
- The above condition signifies an adaptive search
  - In initial generations when random population is distributed over the entire search space, the difference is large and an offspring population with a large diversity is expected.
  - When population tends to converge in some region (later generations), difference is small thereby causing focused search.

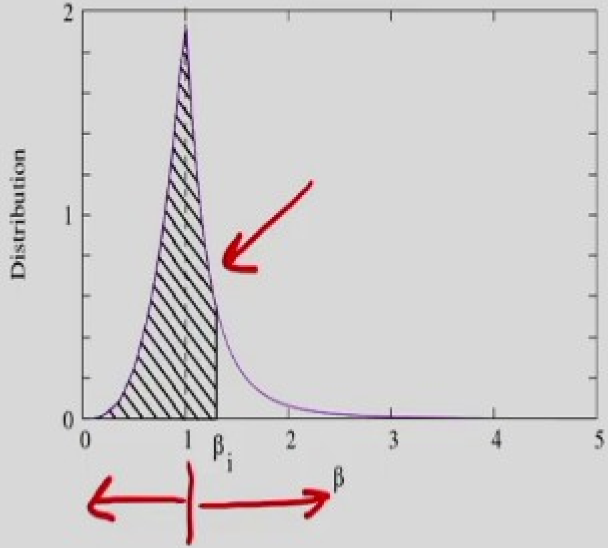
# Simulated Binary Crossover (SBX) Operator

- Probability distribution function:

$$p(\beta_i) = \begin{cases} 0.5(\eta_c+1)\beta_i^{\eta_c}, & \text{if } \beta_i \le 1 \\ 0.5(\eta_c+1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise.} \end{cases}$$

- Calculate $\beta_i$ by equating area under the probability curve equal to $u_i$ (a random number $\in [0,1]$)

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}} & \text{if } u_i \le 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases}$$



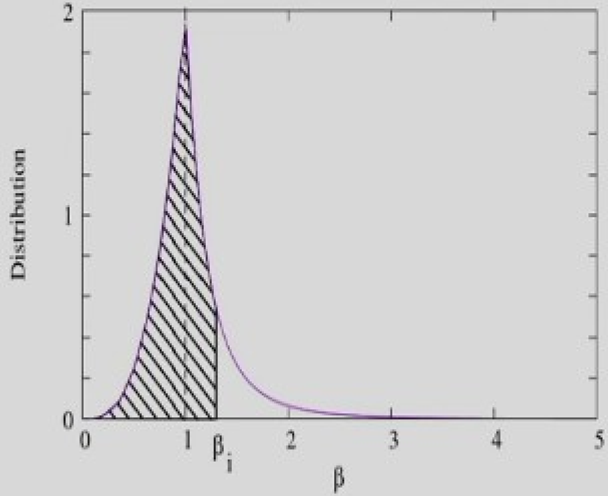
# Simulated Binary Crossover (SBX) Operator

- Probability distribution function:

$$p(\beta_i) = \begin{cases} 0.5(\eta_c+1)\beta_i^{\eta_c}, & \text{if } \beta_i \le 1 \\ 0.5(\eta_c+1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise.} \end{cases}$$

- Calculate $\beta_i$ by equating area under the probability curve equal to $u_i$ (a random number $\in [0,1]$)

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}} & \text{if } u_i \le 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases}$$




- Offspring are:

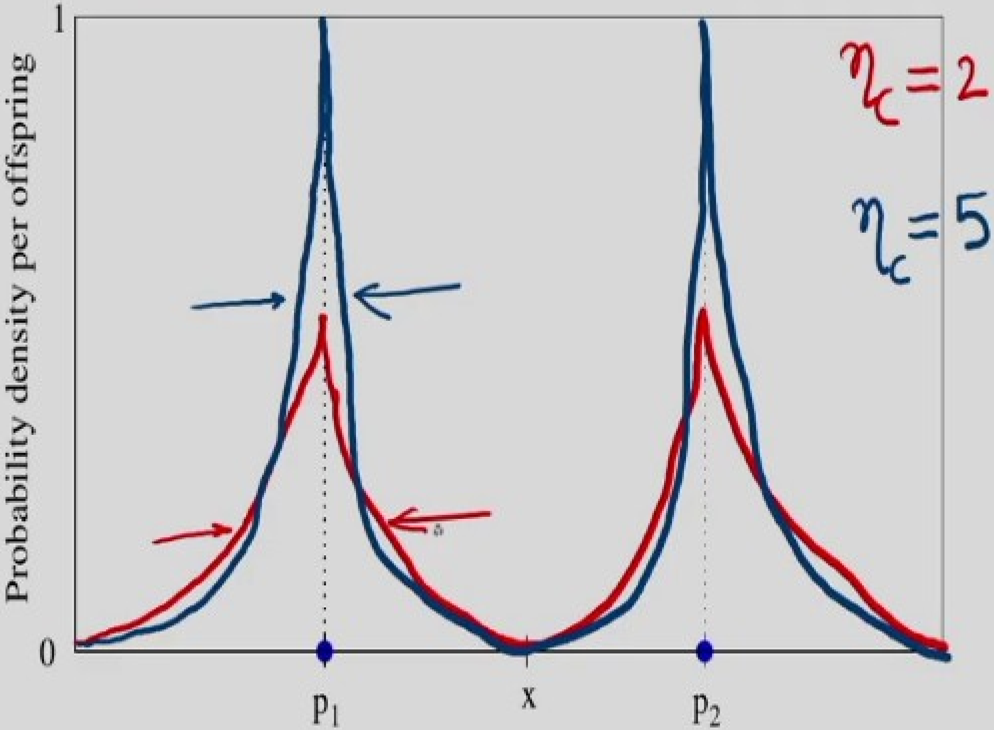
$$x_i^{(1,t+1)} = 0.5\left[(x_i^{(1,t)} + x_i^{(2,t)}) - \beta_i(x_i^{(2,t)} - x_i^{(1,t)})\right]$$

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

- $x_i^{(1,t)} < x_i^{(2,t)}$. $p_1 < p_2$

# Simulated Binary Crossover (SBX) Operator

- Large value of $\eta_c$ indicates higher probability of creating 'near-parent' offspring
- Small $\eta_c$ allows distant solutions to be created as offspring.



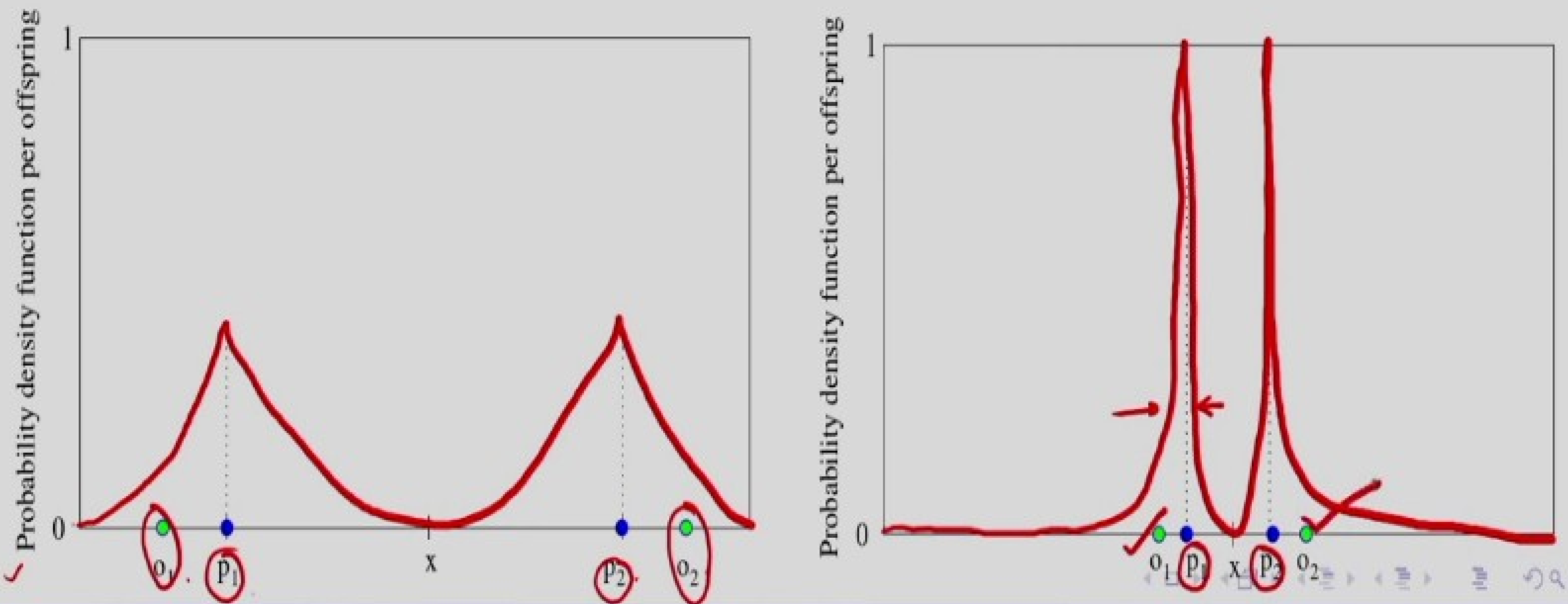
# Simulated Binary Crossover (SBX) Operator

Properties of SBX crossover:

- The difference between the offspring is in proportion to the parent solutions.

$$\left(x_i^{(2,t+1)} - x_i^{(1,t+1)}\right) = \beta_i \left(x_i^{(2,t)} - x_i^{(1,t)}\right)$$

- Near-parent solutions are monotonically more likely to be chosen as offspring than solutions distant from parent.



# SBX Crossover Operator

- The preceding operator creates solution in the range of $[-\infty, \infty]$.
- However, we need to generate solution within the range that is $x_i^{(L)} \leq x_i \leq x_i^{(U)}$.
- Find the cumulative probability

$$
\begin{aligned}
P_1' &= \int_0^{\beta^{(L)}} P(\beta) d\beta, \\
P_2' &= \int_0^{\beta^{(U)}} P(\beta) d\beta,
\end{aligned}
$$

- Here, $\beta^{(L)}$ and $\beta^{(U)}$ are the spread factors for the lower and upper bounds of variable $x_i$
- These spread factors are calculated as

$$\beta^{(L)} = \frac{p_1 + p_2 - 2x_i^{(L)}}{|p_2 - p_1|}$$

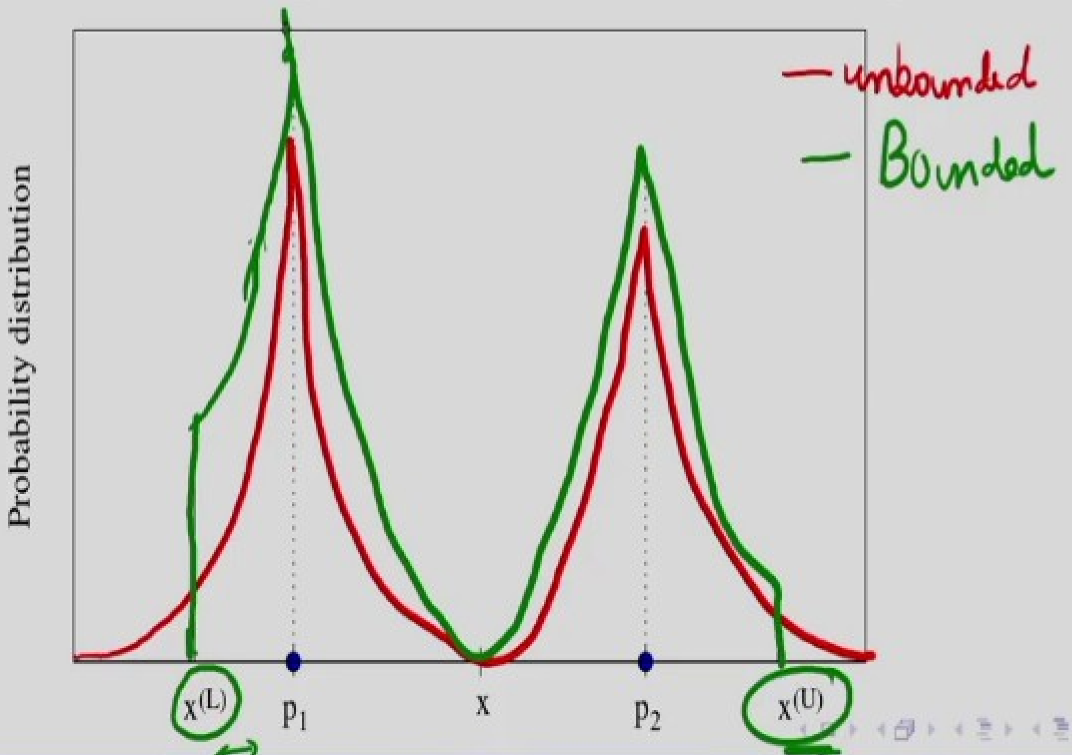
$$\beta^{(U)} = \frac{2x_i^{(U)} - p_1 - p_2}{|p_2 - p_1|}$$

- We can get two spread factors $\beta_1'$ and $\beta_2'$ using modified probability distributions $P(\beta)/P_1'$ and $P(\beta)/P_2'$, respectively.

## SBX Crossover Operator

- Offspring solutions are

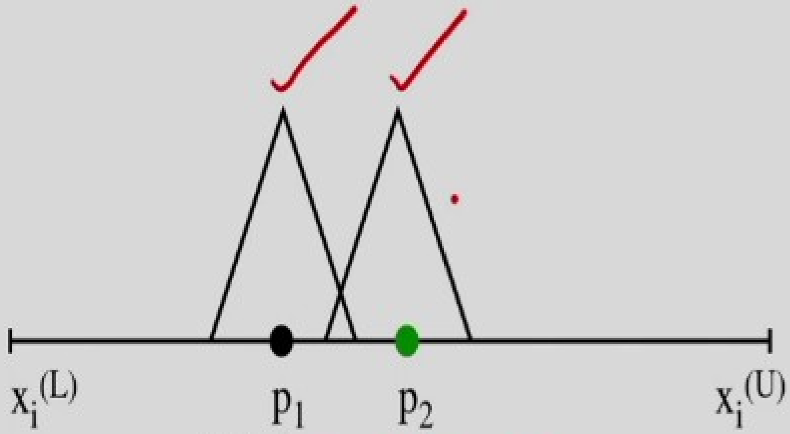
$$\begin{aligned} x_i^{(1,t+1)} &= 0.5\left[(x_i^{(1,t)} + x_i^{(2,t)}) - \beta_1'(x_i^{(2,t)} - x_i^{(1,t)})\right], \\ x_i^{(2,t+1)} &= 0.5\left[(x_i^{(1,t)} + x_i^{(2,t)}) + \beta_2'(x_i^{(2,t)} - x_i^{(1,t)})\right]. \end{aligned}$$



# Fuzzy Recombination (FR) Operator

- FR operator is similar to simulated binary crossover (SBX) operator. When $\eta_c = 1$, SBX and FR operators are the same.

- FR operator used triangular probability distribution.
- Distribution has its apex located at the parent solution and the base is proportional to the difference in parents, i.e., $\Delta = d(x_i^{(2,t)} - x_i^{(1,t)})$.
- FR operator possesses same properties as SBX operator.

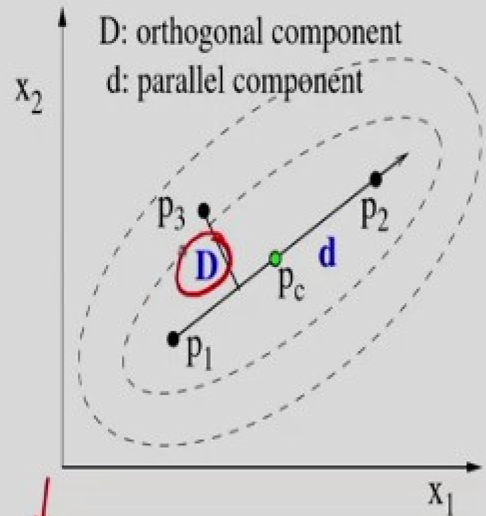



$|\Delta| = d\,(p_2 - p_1)$

$d$ → user-defined

# Unimodal Normally Distributed Crossover (UNDX) Operator

- It is proposed by Ono and Kobayashi, 1997
- In this crossover operator, three or more parent solutions are used to create two or more offspring solutions.
- Offspring are created from an ellipsoidal probability distribution with one axis formed along the line joining two of three solutions.

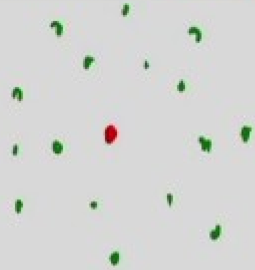



- Two parent solutions $(p_1, p_2)$ are picked randomly and the parallel component is calculated as $d = p_2 - p_1$.
- The center of these parent solutions is $p_c = (p_1 + p_2)/2$.
- The third parent solution $(p_3)$ is picked randomly and an orthogonal component $(D)$ is found.

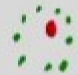
- **Offspring solution** is calculated as $x_o = p_c + \zeta d + \sum_{i=1}^{N-1} \eta_i p_i D$, where $\zeta$, $\eta_i$ are random numbers and $p_i$ is the orthogonal bases that span the subspace perpendicular to $d$.

# Crossover Properties

Case-1



Case-2



**Desired properties expected from crossover operators**

- The population mean should not change.
  - We know that most crossover operators do not use any fitness value. Therefore, a crossover operator do not steer the search in any particular direction.
  - We need a crossover operator that can generate offspring population such that the mean of offspring population should be the same as that of the parent population.
  - However, the variance can change.

# Crossover Properties

**Desired properties expected from crossover operators**

- The population mean should not change.
  - We know that most crossover operators do not use any fitness value. Therefore, a crossover operator do not steer the search in any particular direction.
  - We need a crossover operator that can generate offspring population such that the mean of offspring population should be the same as that of the parent population.
  - However, the variance can change.
- The population diversity should increase.
  - We know that selection operator reduces variance by selecting solutions using fitness values.
  - If crossover operator also reduces variance of the population, it may lead to a premature convergence.
  - Therefore, we need a crossover operator that should increase variance of the population.

# Similarity in Different Crossover Operators

- Beyer and Deb (2000) conducted a study on using three types of crossover operators with their characteristics parameters.
- A flat landscape function was chosen for identical variance growth of operators.

| SBX | BLX | FR |
|---|---|---|
| $\eta_c$ | $\alpha$ | $d$ |
| 2 | 0.662 | 1.095 |
| 3 | 0.500 | 0.707 |
| 5 | 0.419 | 0.433 |
| 10 | 0.3801 | 0.226 |

- The study suggested that these crossover operators can perform similar for some characteristic parameter values.

# Random Mutation Operator

- Michalewicz, 1992 proposed random mutation operator to create solution randomly from the entire search space.

$$y_i^{(1,t)} = r_i(x_i^{(U)} - x_i^{(L)}),$$

where $r_i$ is a random number $\in [0, 1]$

- Independent of the parent solutions and equivalent to random initialization.

# Random Mutation Operator

- Michalewicz, 1992 proposed random mutation operator to create solution randomly from the entire search space.

$$y_i^{(1,t)} = r_i(x_i^{(U)} - x_i^{(L)}),$$

where $r_i$ is a random number $\in [0, 1]$

- Independent of the parent solutions and equivalent to random initialization.

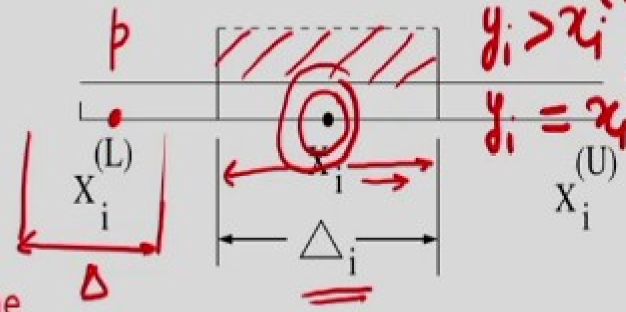
$y_i < x_i^{(L)}$
$y_i = x_i^{(L)}$

- A solution in the vicinity of parent solution with a uniform probability distribution (dashed-line)

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

where $\Delta_i$ is a user-defined maximum perturbation allowed in the $i$−th decision variable.

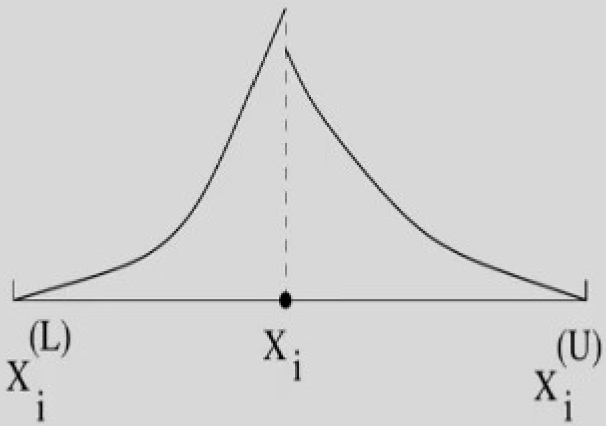
$y_i > x_i^{(U)}$
$y_i = x_i^{(U)}$




- Care should be taken if $y_i^{(1,t)}$ takes value outside of the prescribed lower and upper limits.

# Non-Uniform Mutation Operator

- Michalewicz, 1992 proposed a non-uniform mutation operator with the idea that
  - probability of creating a solution closer to the parent is more than the probability of creating one away from it.




- As generations ($t$) proceed, this probability of creating solutions closer to the parent gets higher and higher.

$$y_i^{(1,t)} = x_i^{(1,t)} + \delta_i^t,$$

where

$$\delta_i^t = \begin{cases} (x_i^{(U)} - x_i^{(1,t)}).\left(1 - r_i^{(1-t/T)^b}\right) & \text{probability} \leq 0.5 \\ (x_i^{(1,t)} - x_i^{(L)}).\left(1 - r_i^{(1-t/T)^b}\right) & \text{otherwise} \end{cases}$$

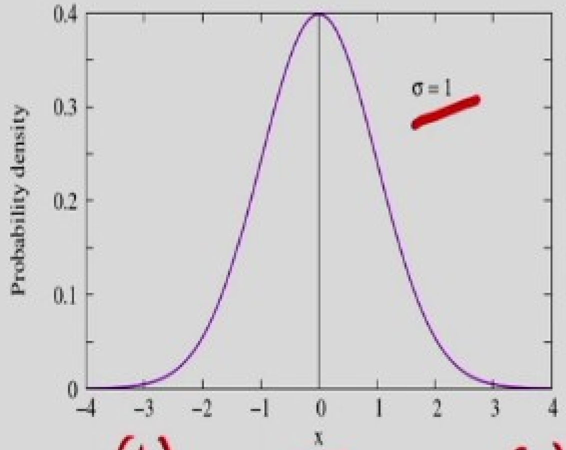
- $T$ is the maximum allowed generations and $b$ is a non-negative user defined parameter.

# Normally Distributed Mutation Operator

- Simple and popular method is to use a zero-mean Gaussian probability distribution:

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

- $\sigma_i$ is a fixed user defined parameter. It must be set correctly in a problem.
- Special care must be taken to respect boundary limits on decision variables.




$$y_i < x_i^{(L)} \quad y_i = x_i^{(L)} \qquad y_i > x_i^{(U)} \quad y_i = x_i^{(U)}$$

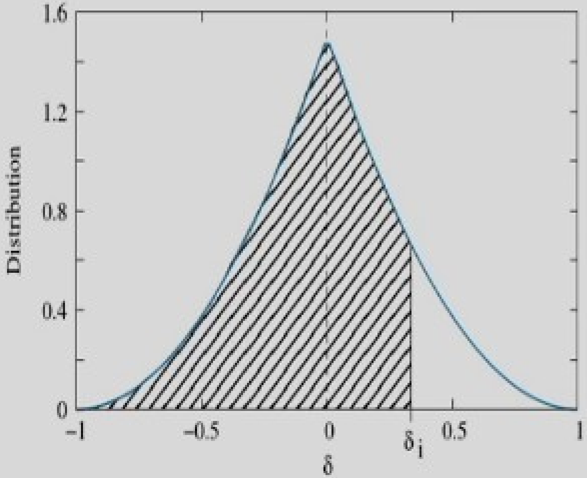
# Polynomial Mutation operator

- Use polynomial distribution for perturbing a solution

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{(U)} - x_i^{(L)})\bar{\delta}_i$$

- $\bar{\delta}_i$ is calculated from polynomial probability distribution

$$P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m} :$$




- Calculate $\bar{\delta}_i$ by equating area under the probability curve equal to $r_i$ (a random number $\in [0, 1]$)

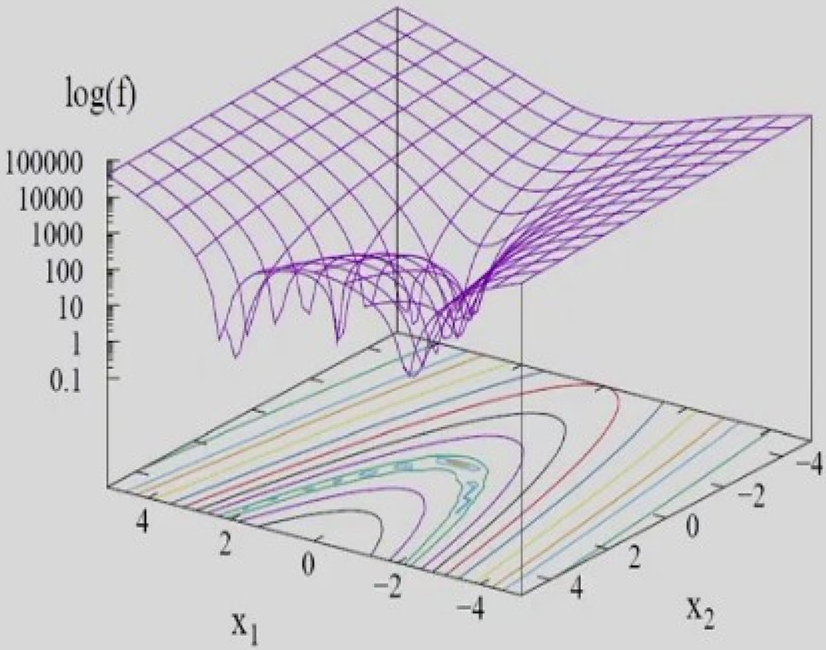
$$\bar{\delta}_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5, \\ 1 - [2(1 - r_i)]^{1/(\eta_m+1)}, & \text{if } r_i \geq 0.5. \end{cases}$$

# Rosenbrock Function

## Rosenbrock Function

Minimize $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right),$

bounds $-5 \leq x_i \leq 5$ and $i = 1, \ldots, n.$

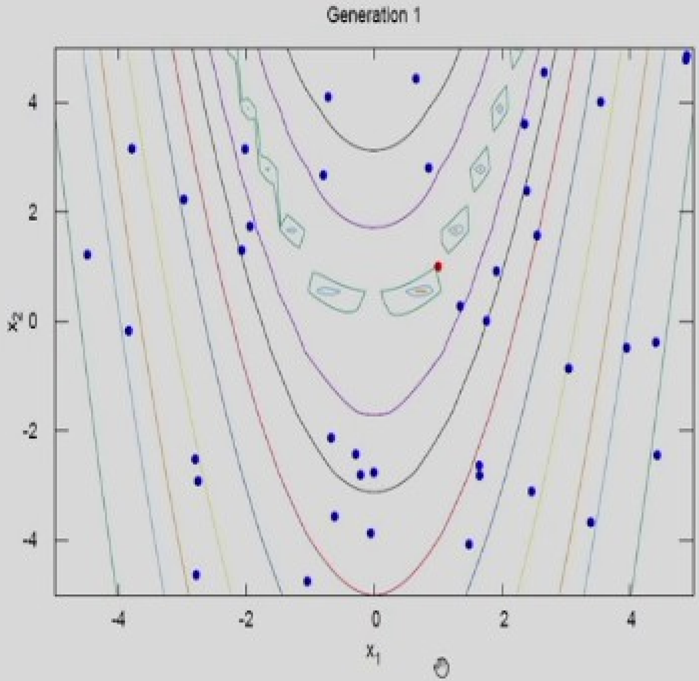



- Optimal solution is $x^* = (1, \ldots, 1)^T$ and $f(x^*) = 0$

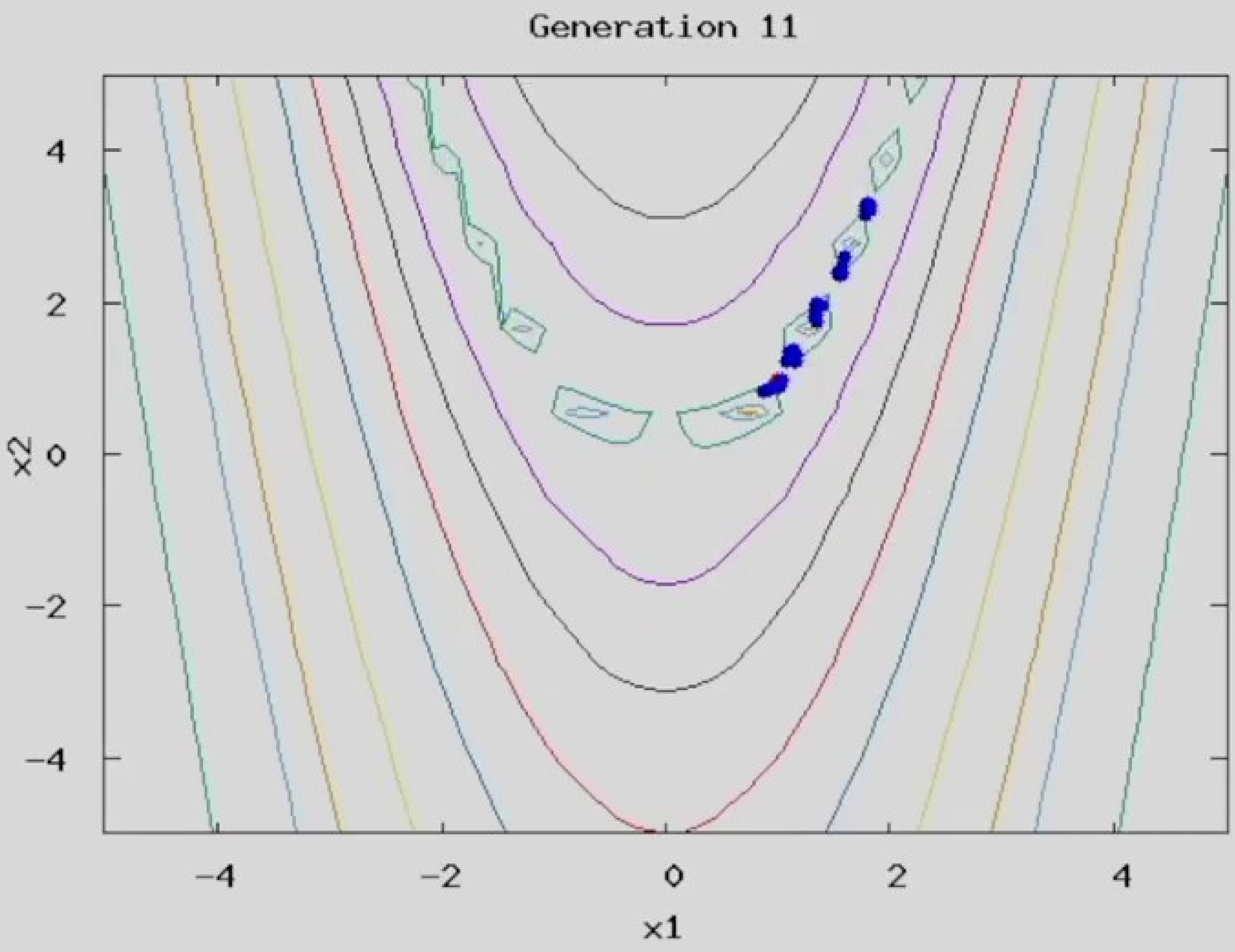
# Rosenbrock Function

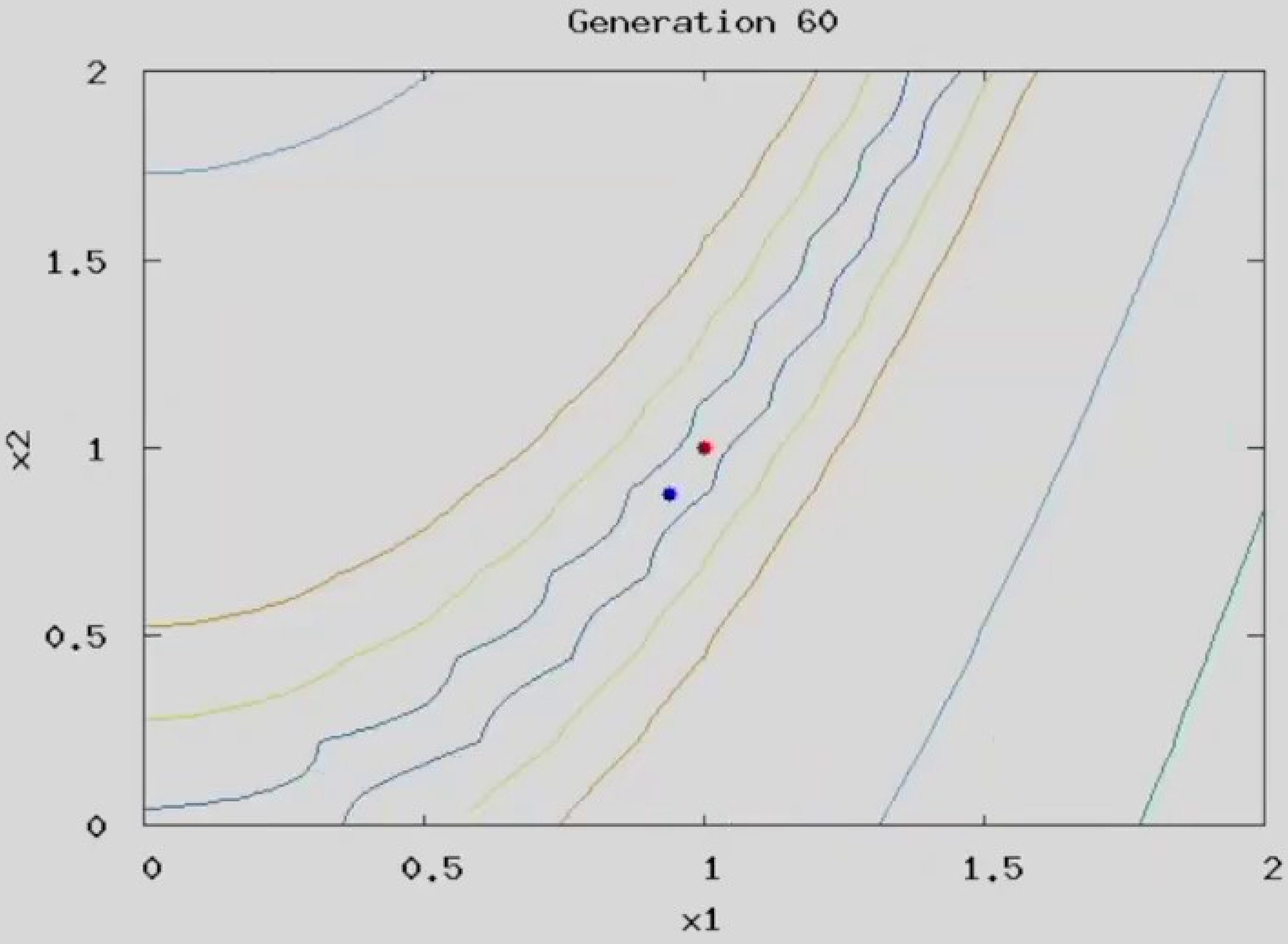
## RGA Parameters

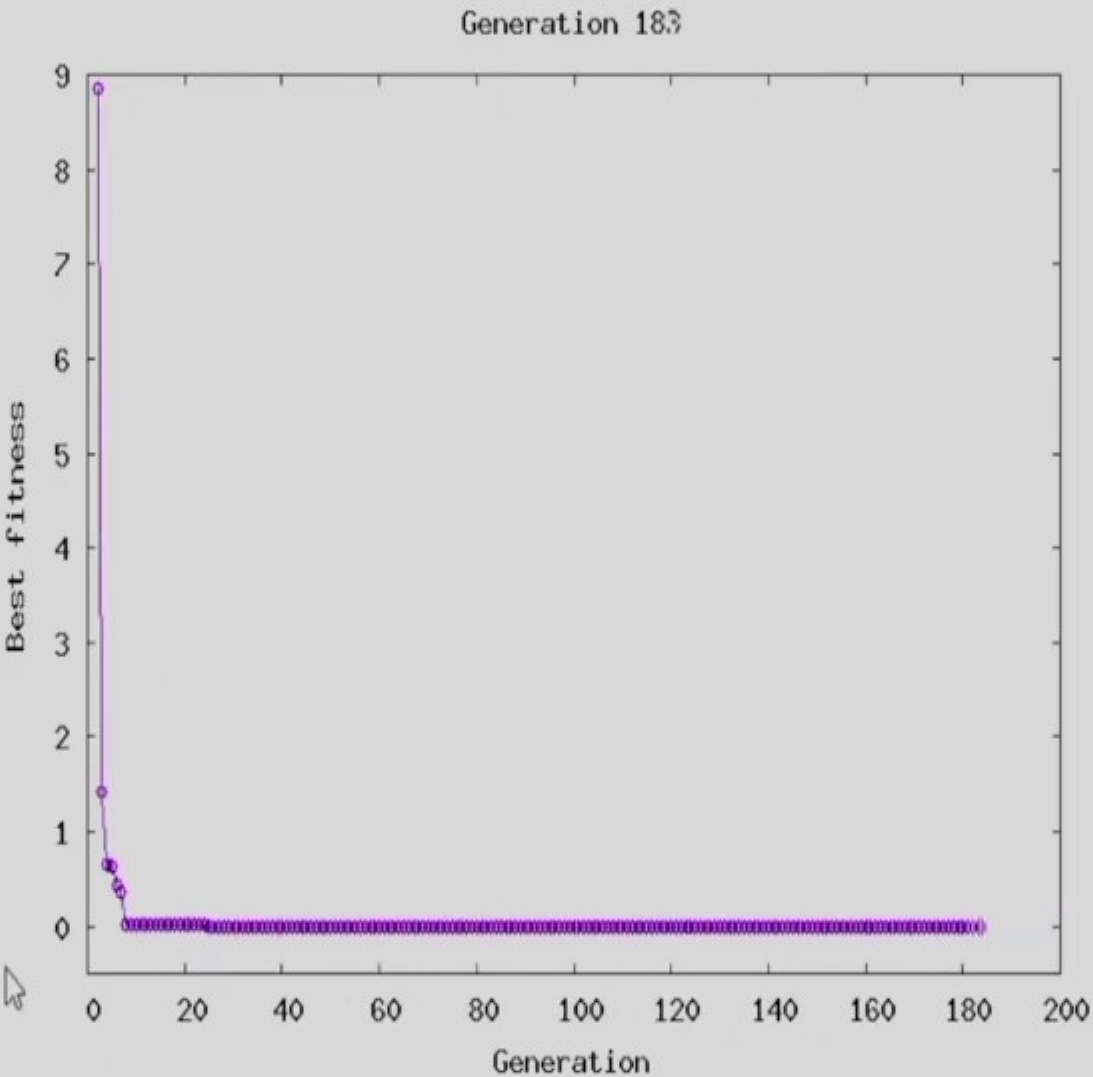
- Number of variables: $n = 2$
- Population size: $N = 40$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.5$
- Binary tournament selection operator
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)-$strategy




- Simulation Link
- Progress Link


Generation 11
4
2
0
−2
−4
x2
−4
−2
0
2
4
x1


Generation 60
2
1.5
1
0.5
0
x2
0
0.5
1
1.5
2
x1

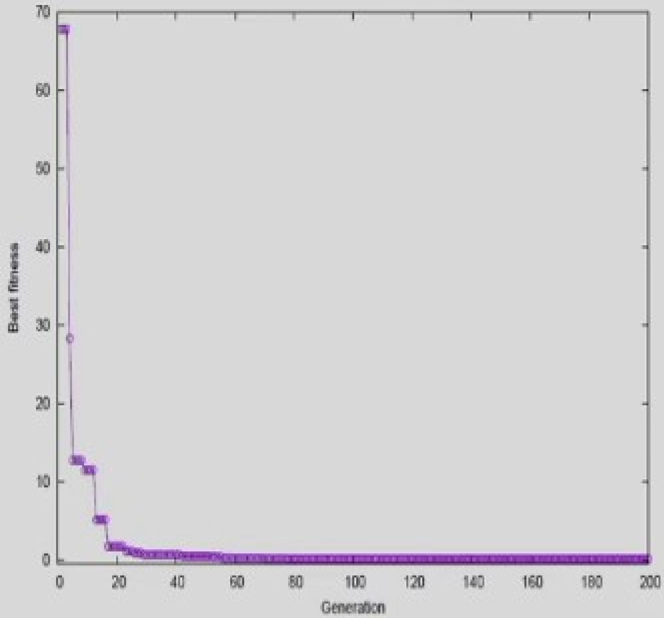


Generation 183
Best fitness
Generation
0
1
2
3
4
5
6
7
8
9
0
20
40
60
80
100
120
140
160
180
200

# Rosenbrock Function

## RGA Parameters

- Number of variables: $n = 4$
- Population size: $N = 40$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.25$
- Binary tournament selection operator
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)$–strategy

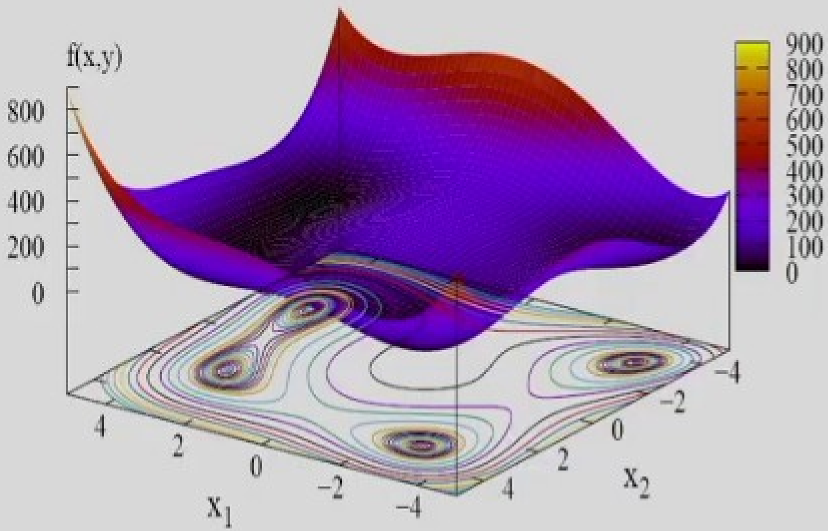



- Progress Link

# Himmelblau Function

## Himmelblau Function

Minimize $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,$
bounds $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5.$

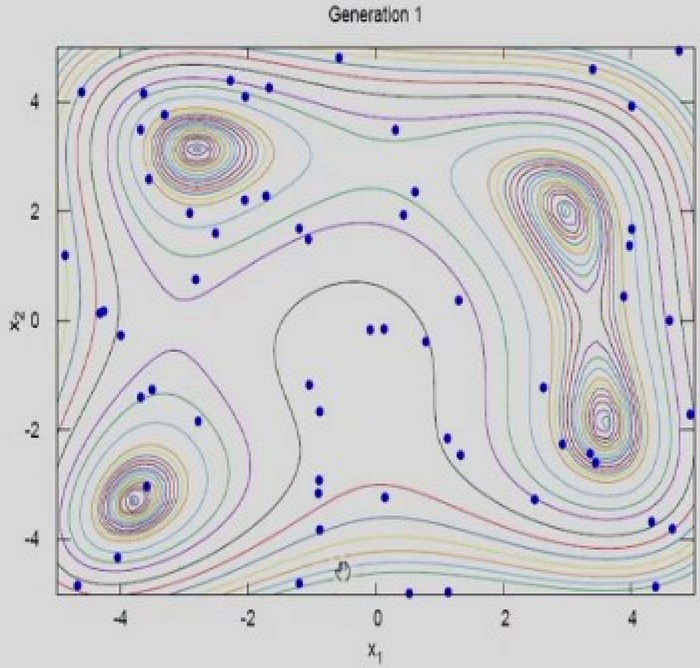



- Multi-modal function: it has 4 minimum points
- First optimal solution is $x^* = (3, 2)^T$ and $f(x^*) = 0$
- Second optimal solution is $x^* = (-2.805, 3.131)^T$ and $f(x^*) = 0$
- Third optimal solution is $x^* = (-3.779, -3.283)^T$ and $f(x^*) = 0$
- Fourth optimal solution is $x^* = (3.584, -1.848)^T$ and $f(x^*) = 0$

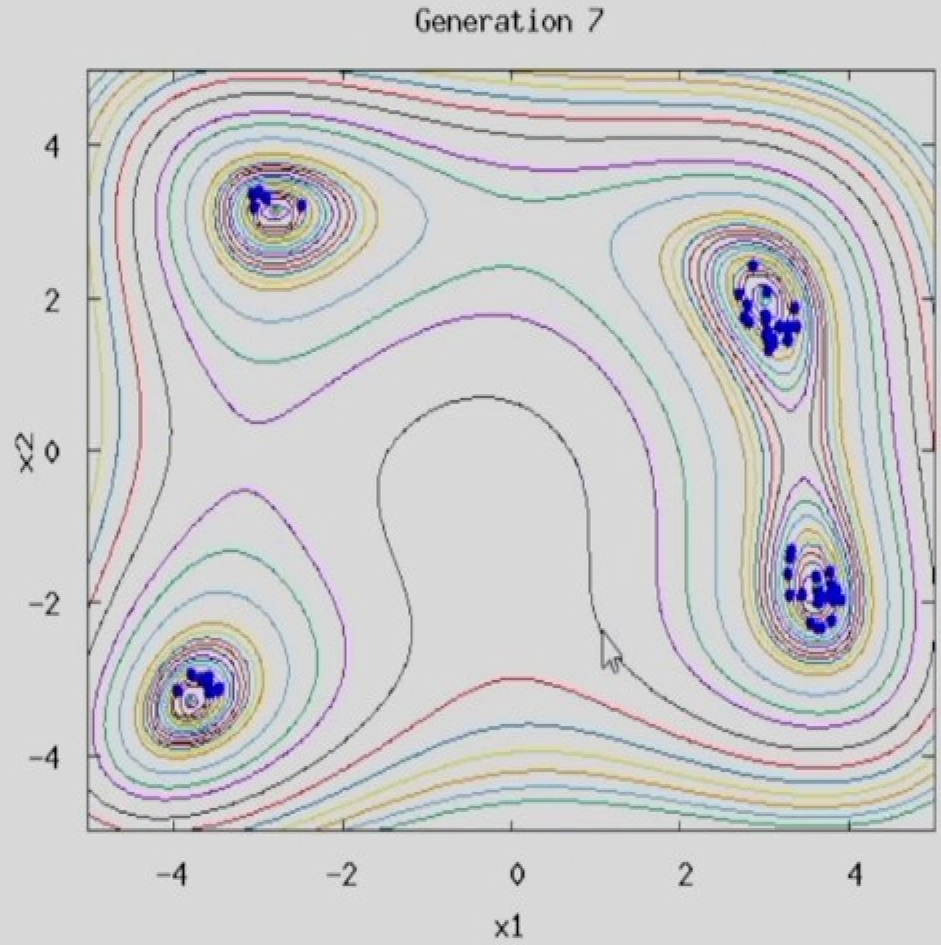
# Himmelblau Function

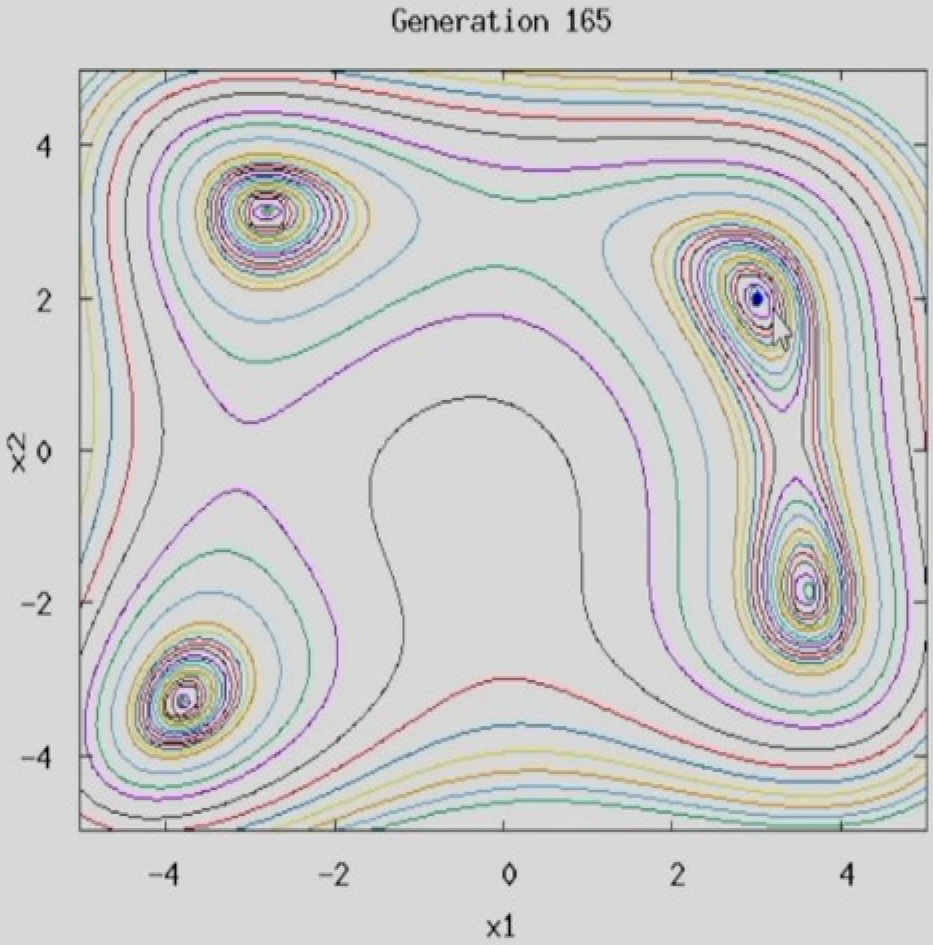
## RGA Parameters

- Population size: $N = 60$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.5$
- Binary tournament selection operator
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)$–strategy




- Simulation Link
- Progress Link


Generation 7
4
2
0
-2
-4
x2
-4
-2
0
2
4
x1

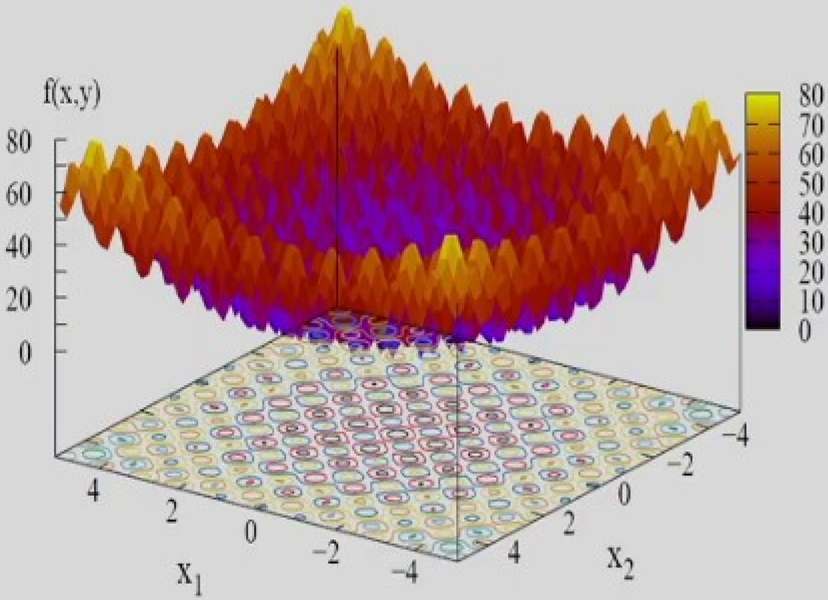

Generation 165
4
2
0
−2
−4
x2
−4
−2
0
2
4
x1

# Rastrigin Function

## Rastrigin Function

Minimize $\quad f(x_1,\ldots,x_n) = 10n + \sum_{i=1}^{N}(x_i^2 - 10\cos(2*\pi x_i)),$

bounds $\quad -5.12 \le x_i \le 5.12$ and $i = 1,\ldots,n.$

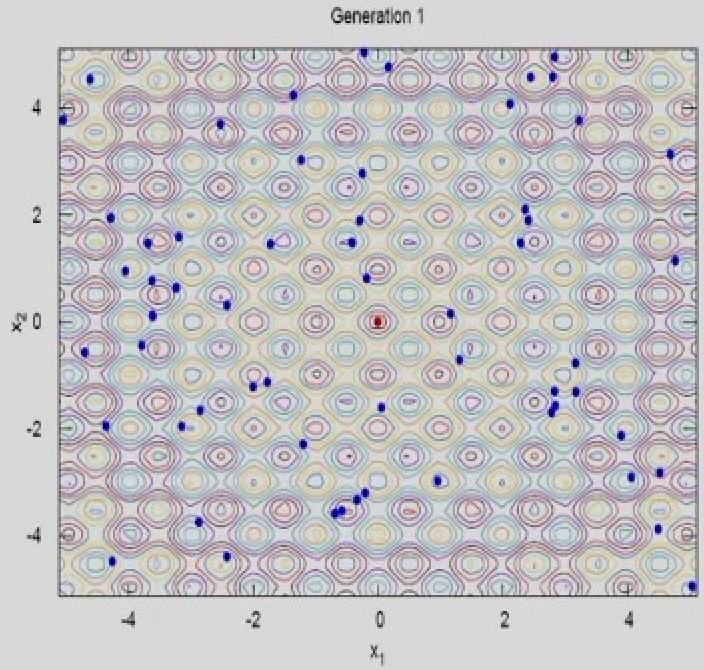



- Optimal solution is $x^* = (0,\ldots,0)^T$ and $f(x^*) = 0$

# Rastrigin Function

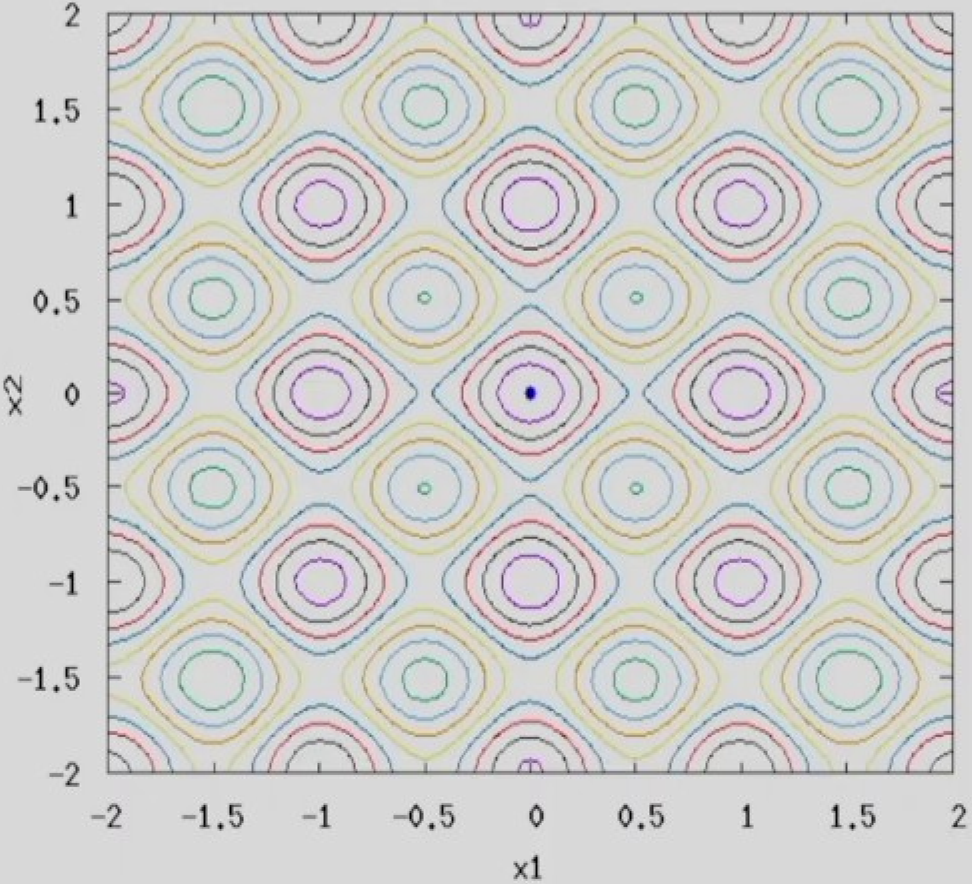
## RGA Parameters

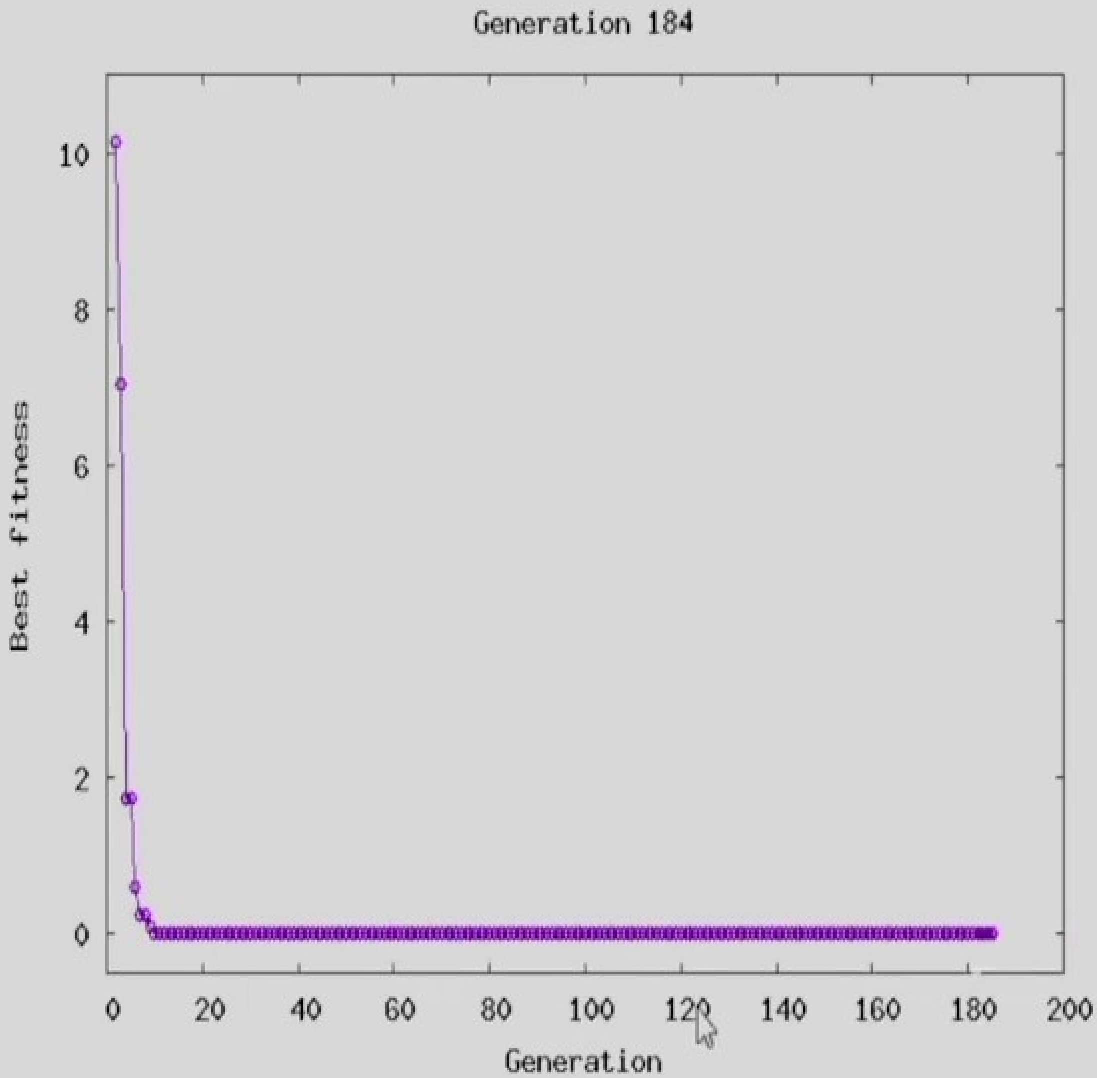
- Number of variable: $n = 2$
- Population size: $N = 60$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.5$
- Binary tournament selection operator
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)$–strategy




- Simulation Link
- Progress Link

Generation 185

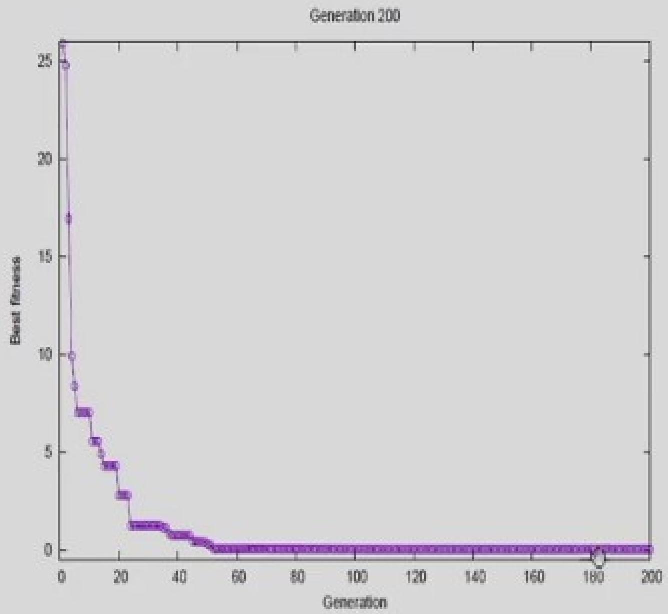



Generation 184
Best fitness
10
8
6
4
2
0
0
20
40
60
80
100
120
140
160
180
200
Generation

# Rastrigin Function

## RGA Parameters

- Number of variable: $n = 4$
- Population size: $N = 60$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.5$
- Binary tournament selection operator
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)$–strategy

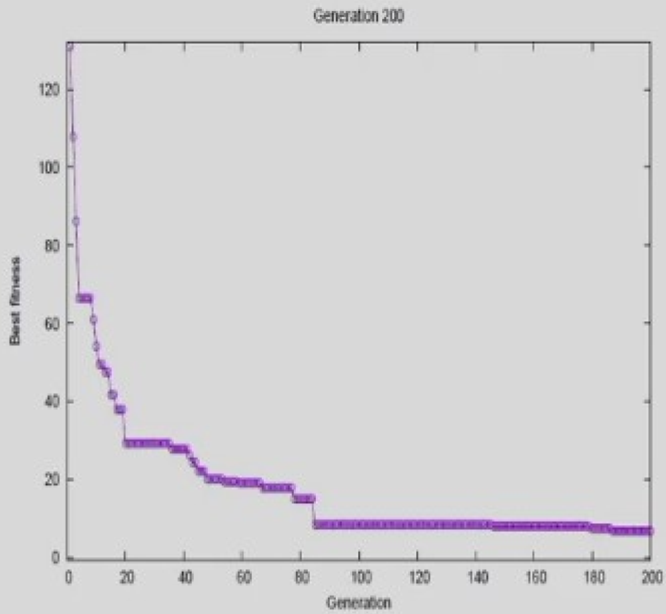



- Simulation ▸ Link

# Rastrigin Function

## RGA Parameters

- Number of variable: $n = 10$
- Population size: $N = 60$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.1$
- Binary tournament selection operator
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)$–strategy



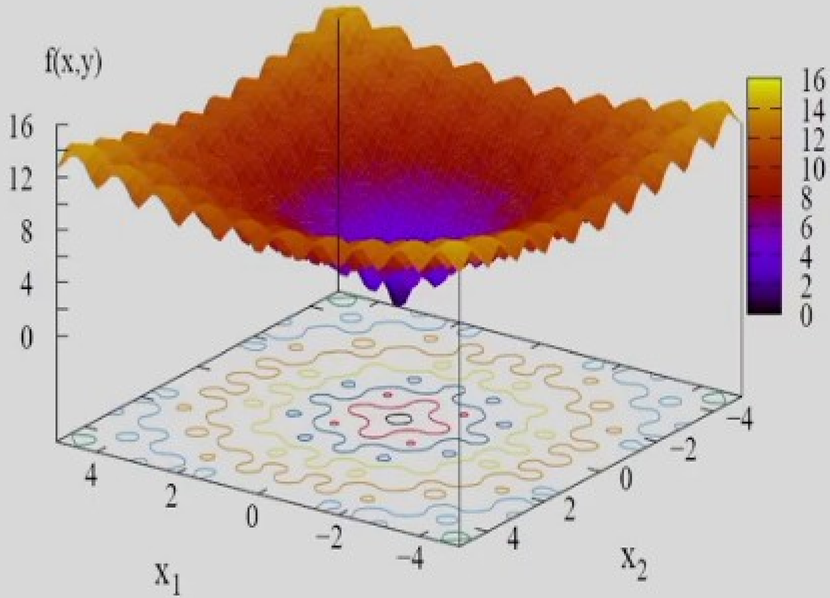

- Simulation Link

# Ackley Function

## Ackley Function

Minimize $f(x_1, x_2) = -20\exp\left(-0.2\sqrt{(0.5(x_1^2 + x_2^2))}\right)$
$- \exp(0.5(\cos(2\pi x_1) + \cos(2\pi x_2))) + \exp(1) + 20,$

bounds $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5.$

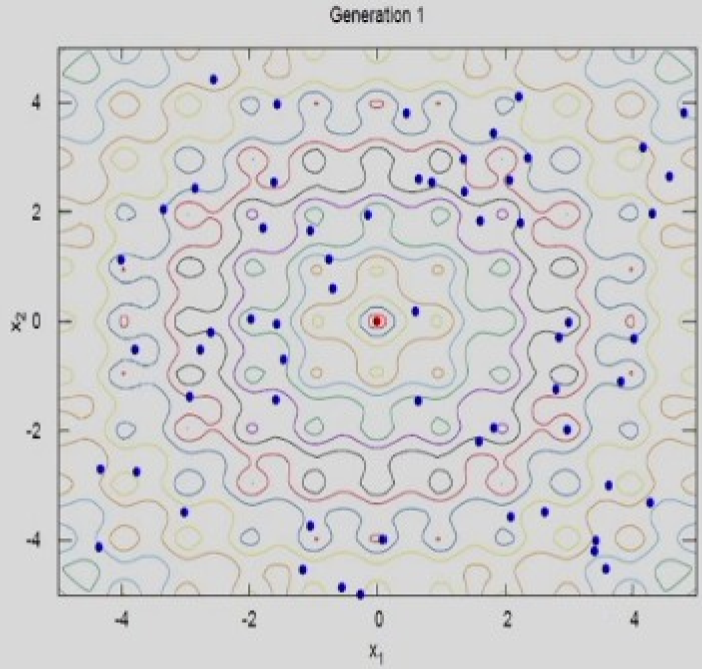



- Optimal solution is $x^* = (0,0)^T$ and $f(x^*) = 0$

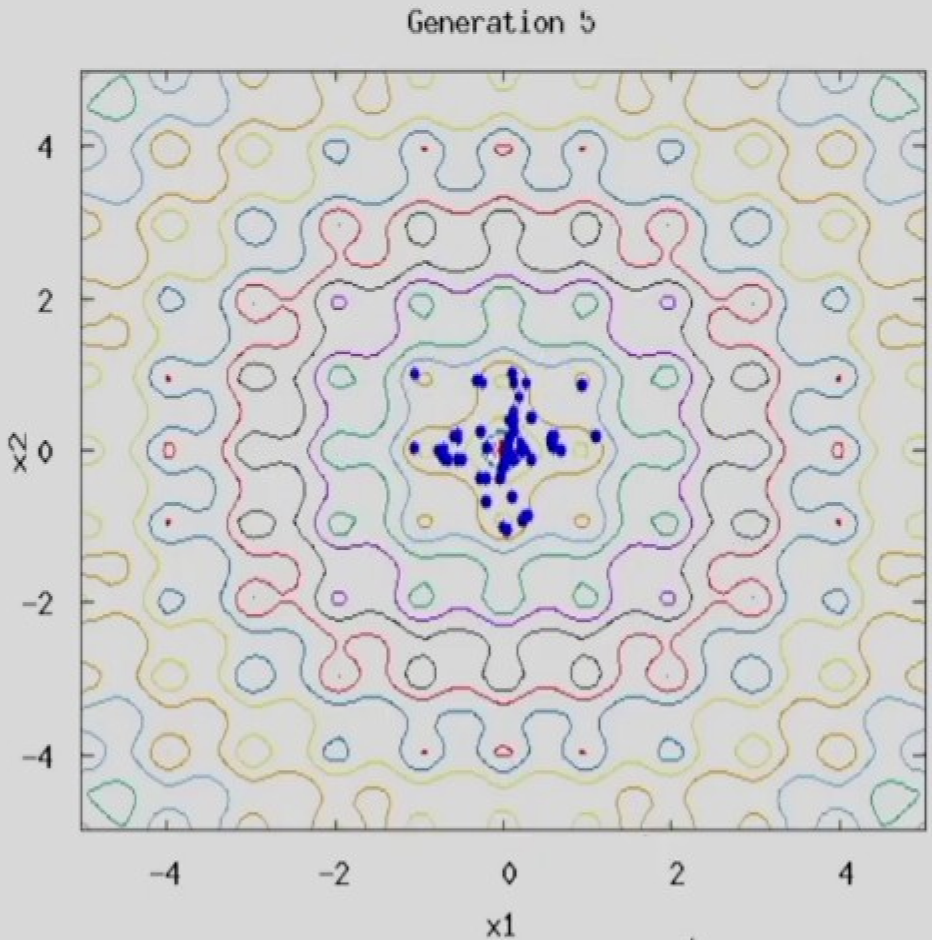
# Ackley Function

## RGA Parameters

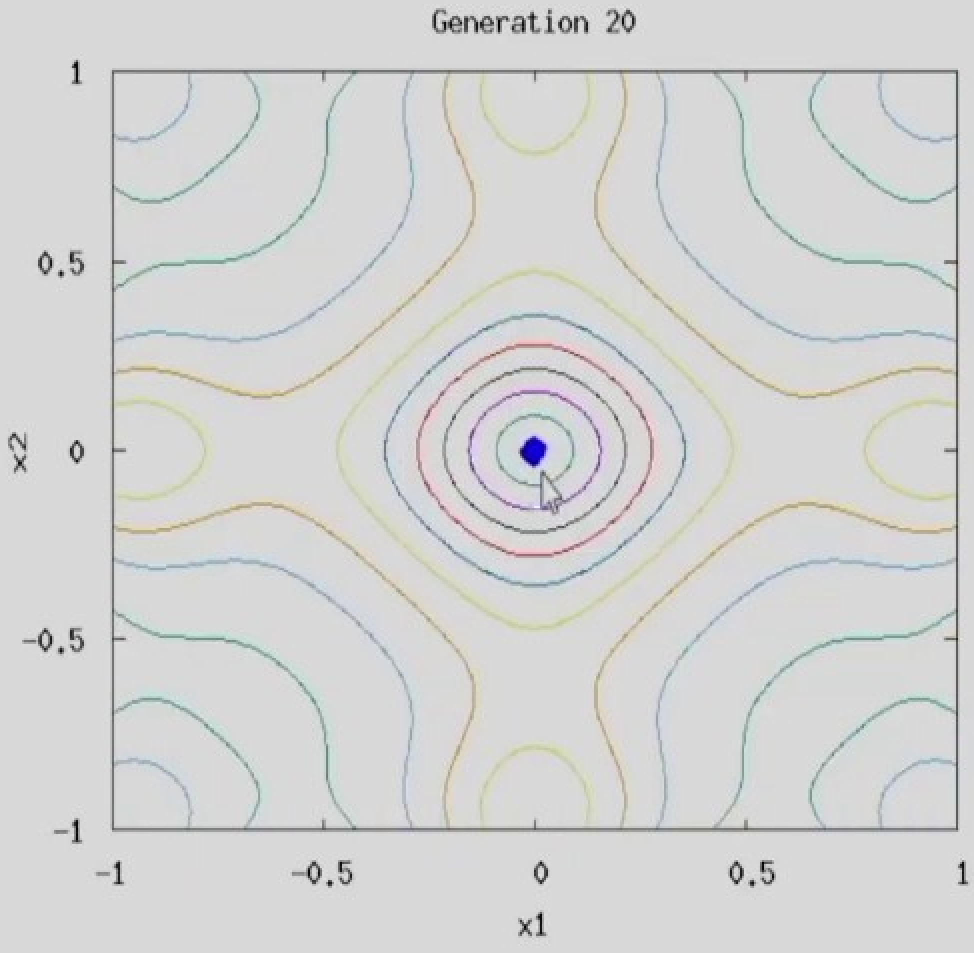
- Population size: $N = 60$
- No. of generations: $T = 200$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation: $p_m = 1/n = 0.5$
- SBX crossover operator: $\eta_c = 15$
- Polynomial mutation operator: $\eta_m = 20$
- $(\mu + \lambda)$–strategy

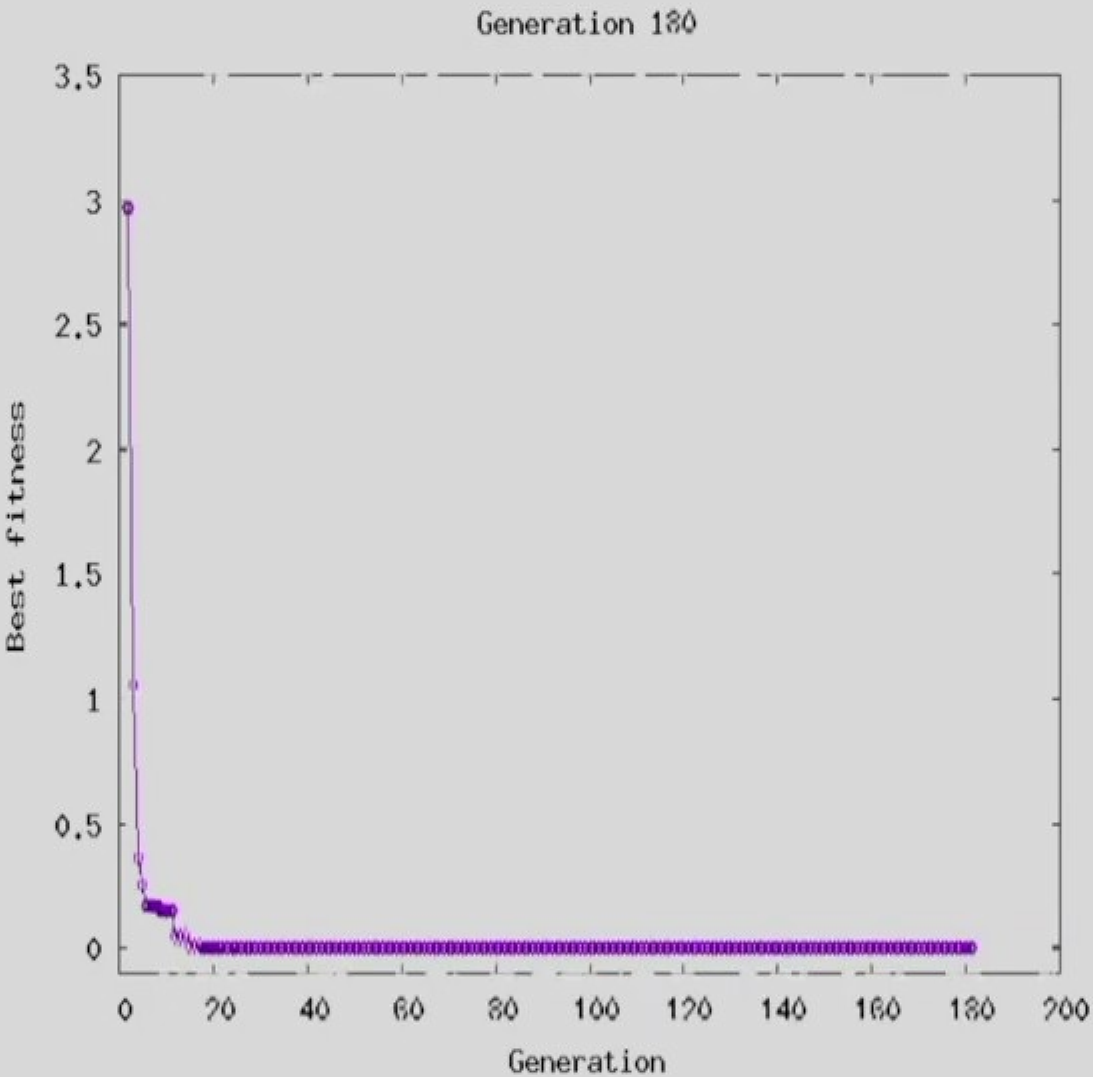



- Simulation Link
- Progress Link



Generation 5
4
2
0
-2
-4
-4
-2
0
2
4
x1
x2


Generation 20
1
0.5
0
-0.5
-1
x2
-1
-0.5
0
0.5
1
x1


Generation 180
Best fitness
3.5
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
140
160
180
200
Generation

# Closure

- Various Operators for RGA
- Selection operators
- Crossover operators
  - Linear crossover operator
  - Blend crossover operator
  - SBX crossover operator
  - Fuzzy recombination operator
  - Unimodal normally distributed crossover operator
- Properties of crossover operator
- Similarity among crossover operators on a flat landscape function
- Mutation operators
  - Random mutation operator
  - Non-uniform mutation operator
  - Normally distributed mutation operator
  - Polynomial mutation operator
- Simulations and application of RGA
  - Rosenbrock function
  - Himmelblau function
  - Rastrigin function
  - Ackley function