



GENETIC ALGORITHMS

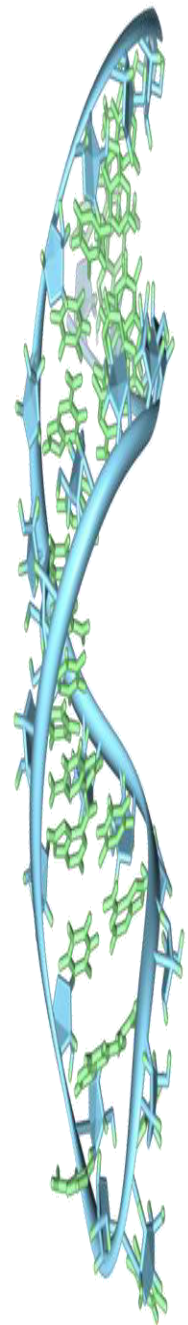
Prof. Ganapati Panda, FNAE, FNASc.

School of Electrical Sciences

Indian Institute of Technology Bhubaneswar

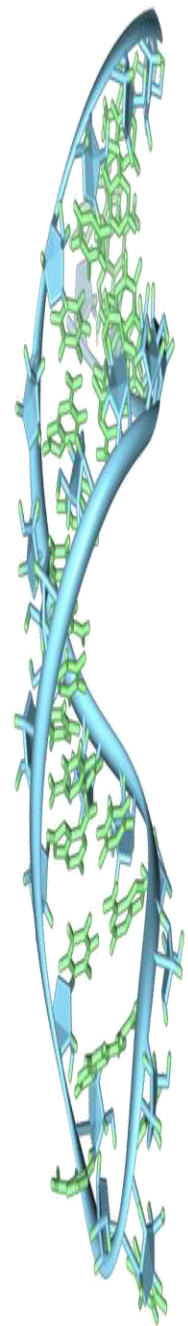
The Genetic Algorithms (GA)

- ❖ Based on the mechanics of biological evolution.
- ❖ Initially developed by John Holland, University of Michigan (1970's)
 - To understand processes in natural systems.
 - To design artificial systems retaining the robustness and adaptation properties of natural systems.
- ❖ Provide efficient techniques for optimization and machine learning applications.
- ❖ It is a stochastic algorithm.
- ❖ It does not use gradient information.
- ❖ Widely used in business, science and engineering.



Genetic Algorithms: Basic idea

- ❖ Start with a population of candidate solutions.
- ❖ Variation: Introduce variation by applying two operators: crossover and mutation.
- ❖ Survival of the fittest: Use a fitness criterion to bias the evolution towards desired features.



Genetic Algorithms Techniques

❖ GAs are a particular class of evolutionary algorithms.

The techniques common to all GAs are:

- **Inheritance** - offspring acquire characteristics
- **Mutation** - change, to avoid similarity
- **Natural Selection** - variations improve survival
- **Crossover** - recombination

❖ GAs are best used when the objective function is:

- Discontinuous
- Highly nonlinear
- High dimension
- Stochastic
- Has unreliable or undefined derivatives

Biological Terminology

❖ **Gene** – a single encoding of part of the solution space, i.e. either single bits or short blocks of adjacent bits that encode an element of the candidate solution

1

❖ **Chromosome** – a string of genes that represents a solution

1	0	1	1	0
---	---	---	---	---

❖ **Population** – the number of chromosomes available to test

0	1	1	0	1
1	1	0	0	1
1	0	1	0	1
1	0	1	0	1
1	0	0	0	1
0	0	1	1	1

Mapping: Nature vs. computer

NATURE	COMPUTER
POPULATION	SET OF SOLUTIONS
INDIVIDUAL	SOLUTION TO A PROBLEM
FITNESS	QUALITY OF SOLUTION
CHROMOSOME	ENCODING FOR A SOLUTION
GENE	PART OF ENCODING OF A SOLUTION
REPRODUCTION	CROSSOVER

Features of Genetic Algorithms

- ❖ Not too fast but cover large search space
 - Capable of quickly finding promising regions of the search space but may take a relatively long time to reach the optimal solution.
- ❖ Good heuristics for combinatorial problems
- ❖ Usually emphasize combining information from good parents (crossover)
- ❖ Different GAs use different
 - Representations
 - Mutations
 - Crossovers
 - Selection mechanisms

Representation

❖ GAs on primarily two types of representations:

- Binary Coded

[0110, 0011, 1101,]

- Real Coded

[13.2, -18.11, 5.72, ...]

❖ Binary-Coded (genotype) GAs must decode a chromosome into a real value (phenotype), for evaluating the fitness value.

❖ Real-Coded GAs can be regarded as GAs that operate on the actual real value (phenotype).

- For Real-Coded GAs, no genotype-to-phenotype mapping is needed.

Selection

- ❖ A proportion of the existing population is selected to breed a new breed of generation.
- ❖ Parents with better fitness have better chances to produce offspring.

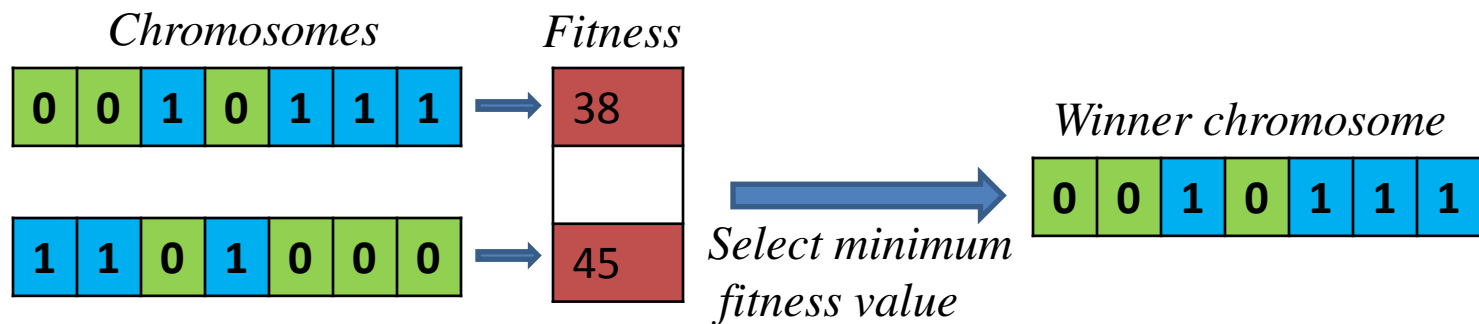
Rank selection

- ❖ Rank selection first ranks the population and then every chromosome receives fitness from this ranking.
 - Selection is based on this ranking rather than absolute differences in fitness.
 - The best will have fitness 1, second best 2 etc. and the worst will have fitness N (number of chromosomes in population).

Tournament selection

Runs a "*tournament*" among a few individuals chosen at random from the population and selects the winner (the one with the best fitness) for crossover

- Two entities are picked out of the pool, their fitness is compared, and the better is permitted to reproduce.
- Selection pressure can be easily adjusted by changing the tournament size.
- Deterministic tournament selection selects the best individual in each tournament.



Roulette-wheel selection

- Assign each individual part of the wheel according to proportion of fitness value.
- ✓ The probability that a individual will be selected is given by the ratio of its fitness to the sum of fitness of all individuals of the current population.

$$P(h_i) = \frac{f(h_i)}{\sum_{j=1}^N f(h_j)}$$

where h_i is the i^{th} individual of current population and size of population is N .

- Spin wheel (N) times to select (N) individuals

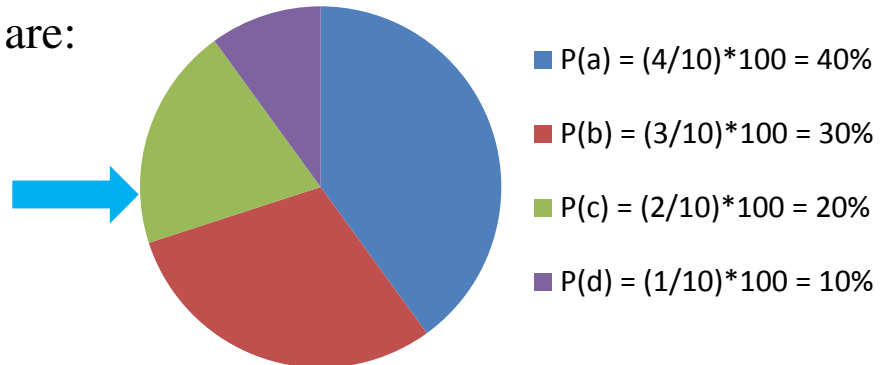
Ex. Fitness values of chromosomes are:

$$f(a) = 4;$$

$$f(b) = 3;$$

$$f(c) = 2;$$

$$f(d) = 1;$$



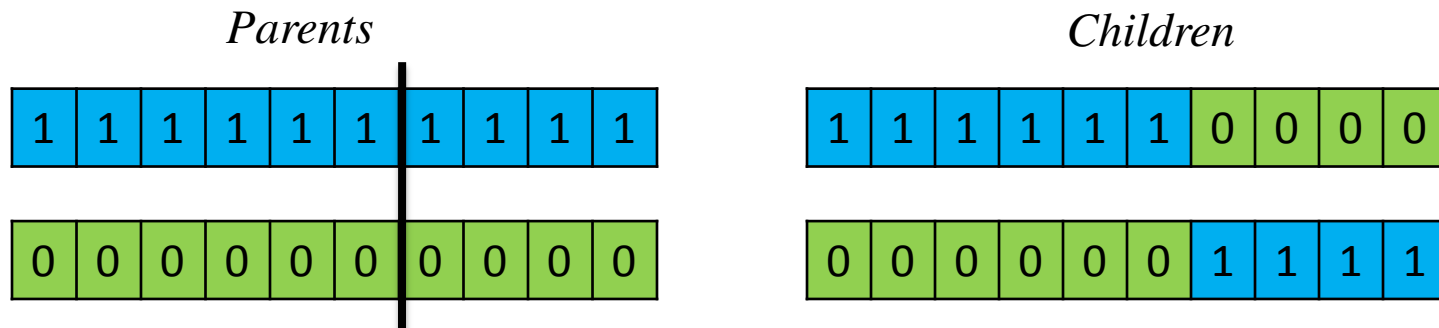
Crossover

***Crossover** is a genetic operator that combines (mates) two individuals (parents) to produce two new individuals (Children).*

- The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents.

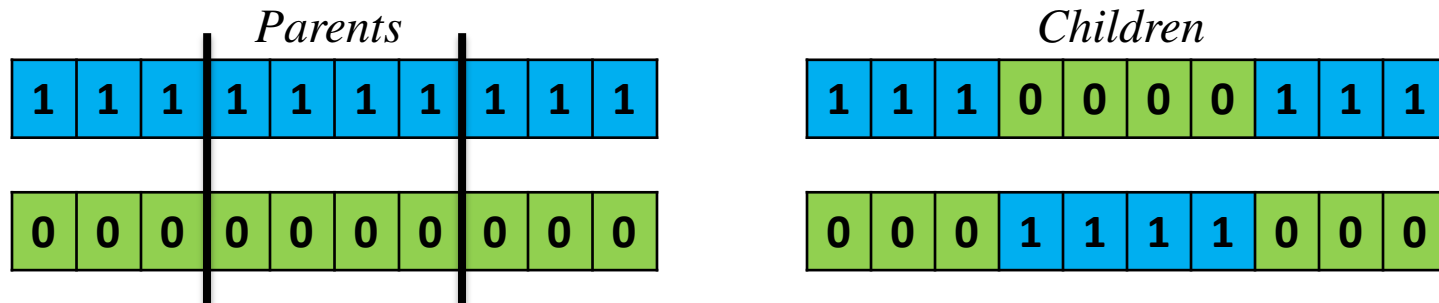
Single-Point Crossover

- Choose a random point
- Split parents at this crossover point
- Create children by exchanging tails
- Probability of crossover is typically in range (0.6, 0.9)



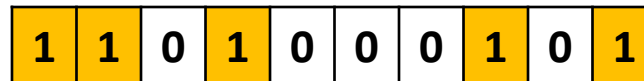
Two-Point Crossover

- Choose two random points
- Split parents at these crossover points
- Create children by exchanging middle portion

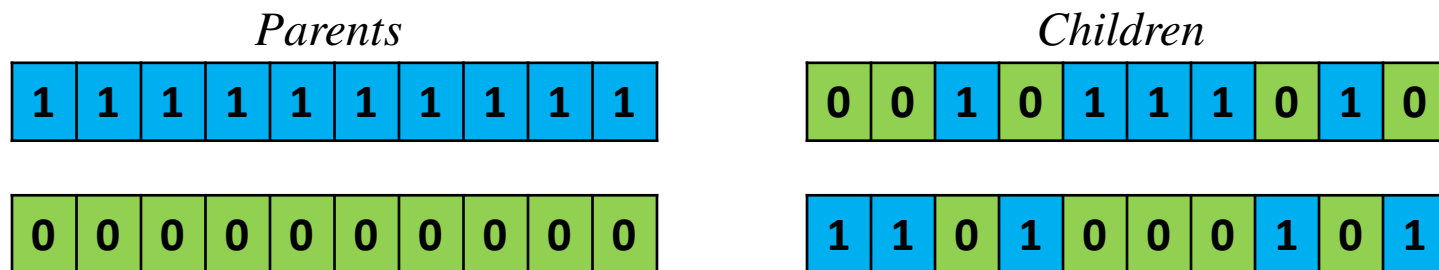


Uniform Crossover

- Randomly generated mask



- Create children by exchanging genes according to mask



Real value Crossover

- Choose two parents randomly from the selected candidates.
- Generate a random number (r) for each dimension of parents.
- Create children by combining the weighted portion of both parents.

$$b = \begin{cases} (2 * r)^{\left(\frac{1}{\mu+1}\right)} & \text{if } r \leq 0.5 \\ \left(\frac{1}{2 * (1-r)}\right)^{\left(\frac{1}{\mu+1}\right)} & \text{if } r > 0.5 \end{cases}$$

$$child_1(j) = \frac{1}{2} \left((1+b) * parent_1(j) + (1-b) * parent_2(j) \right)$$

$$child_2(j) = \frac{1}{2} \left((1-b) * parent_1(j) + (1+b) * parent_2(j) \right)$$

where

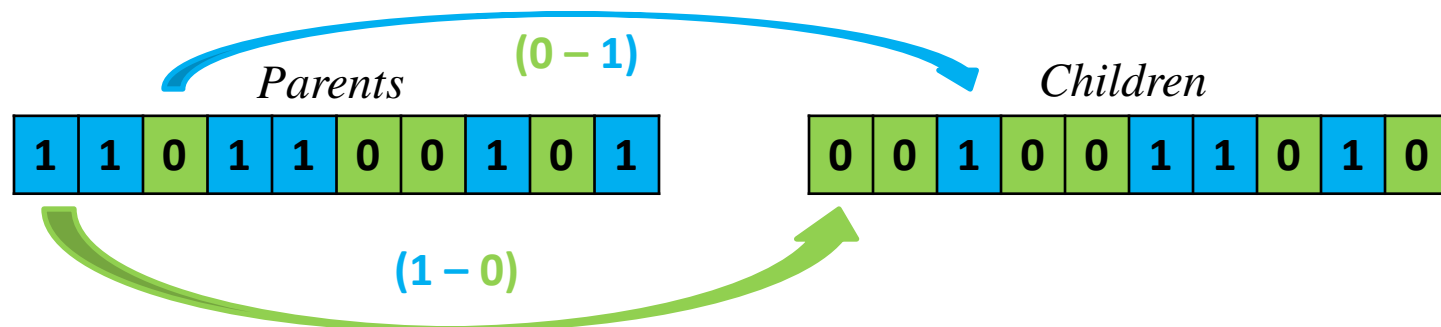
- r is a random number $\{0,1\}$
- μ is a crossover operator
- $child(j)$ represent j^{th} dimension of child.

Mutation

- ❖ **Mutation** is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation.
- ❖ **Mutation Probability** determines how often the parts of a chromosome will be mutated.

Bit-reversal Mutation

- Alter each gene independently
- Choose a random point
- Create children by altering the gene of chosen point
- Mutation probability is typically in range $\{(1/\text{population_size}), (1/\text{chromosome_length})\}$



Polynomial Mutation

- Select one parent randomly from the selected parents.
- Generate a random number (r) for each dimension of parent.

$$d = \begin{cases} (2 * r)^{\left(\frac{1}{\eta+1}\right)} - 1 & \text{if } r \leq 0.5 \\ 1 - (2 * (1 - r))^{\left(\frac{1}{\eta+1}\right)} & \text{if } r > 0.5 \end{cases}$$

$$child(j) = parent(j) + d$$

where

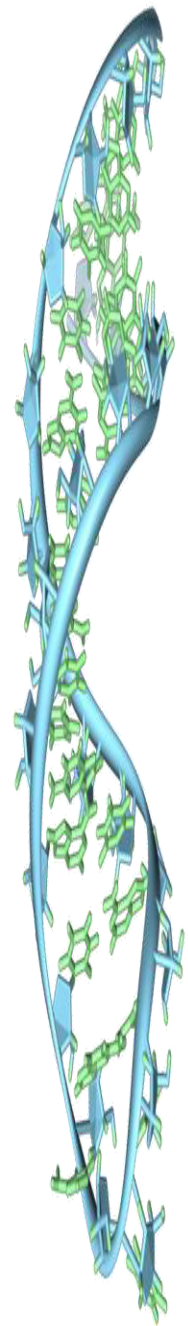
- r is a random number $\{0,1\}$
- η is a mutation operator
- $child(j)$ represent j^{th} dimension of child.

Fitness Function

- ❖ A fitness function quantifies the optimality of a solution (chromosome) so that that particular solution may be ranked against all the other solutions.
- ❖ It depicts the closeness of a given ‘solution’ to the desired result.
- ❖ Watch out for its speed.
- ❖ Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions.

Termination

- ❖ A solution is found that satisfies minimum criteria
- ❖ Fixed number of generations found
- ❖ Allocated budget (computation, time/money) reached
- ❖ The highest ranking solution's fitness is reaching or has reached



Issues with Genetic Algorithms

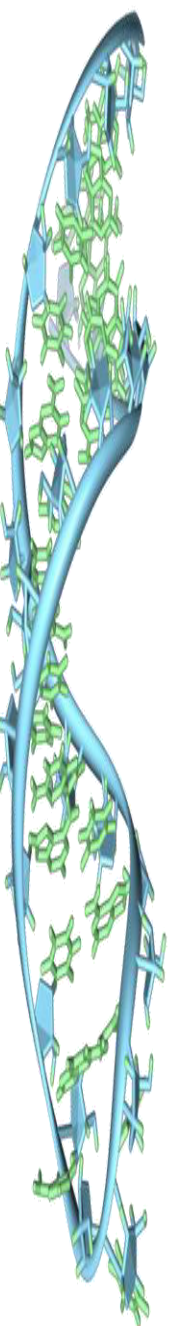
❖ Choosing parameters:

- Population size
- Crossover and mutation probabilities
- Selection, deletion policies
- Crossover, mutation operators, etc.
- Termination criteria

❖ Performance:

- Can be too slow but covers a large search space
- Is only as good as the fitness function

The Basic Genetic Algorithm



```
{ % Generate random population of chromosomes
Initialize population;
% Evaluate the fitness of each chromosome in the population
Evaluate population; [Fitness]
% Create, accept and test a new population:
while Termination_Criteria_Not_Satisfied
{ % Select according to fitness
Select parents for reproduction; [Selection]
% With a crossover probability perform crossover or copy parents
Perform crossover; [Crossover]
% With a mutation probability mutate offspring at each position in chromosome
Perform mutation; [Mutation]
Accept new generation;
Evaluate population; [Fitness]
}
}
```

Minimize – $f(x) = \sin(x)$,

where $0 < x < (2 * \pi)$

Initialize population :

- Search space is of single dimension (given).
- Let population size (N) = 10
- Initialize population with 10 chromosomes having single dimensioned real value.
- These values are randomly distributed in between $[0, 2 * \pi]$.

x

2.620
4.526
0.001
1.900
0.922
0.580
1.170
2.171
2.493
3.385

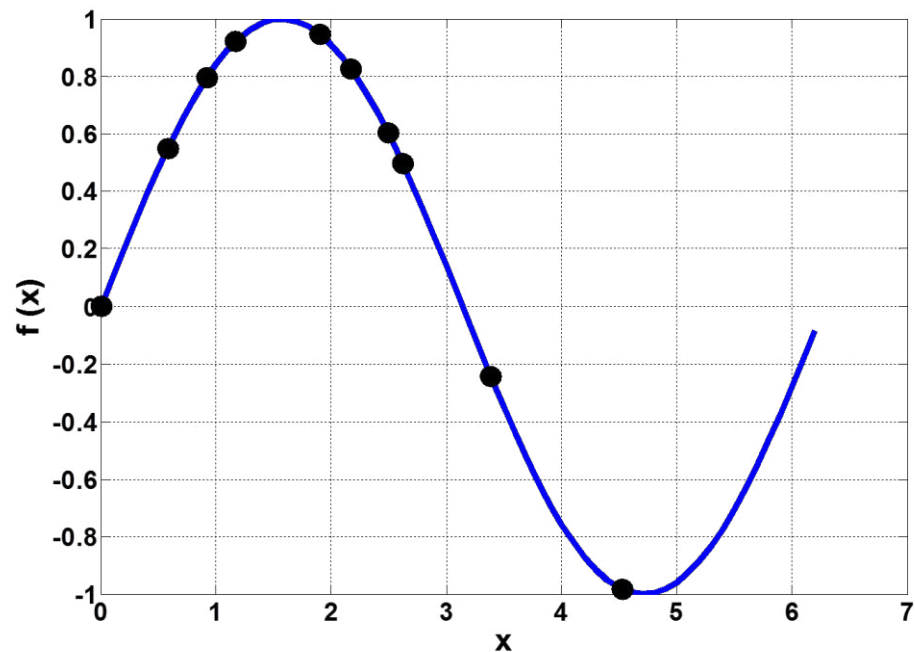
chromosomes

Evaluate function:

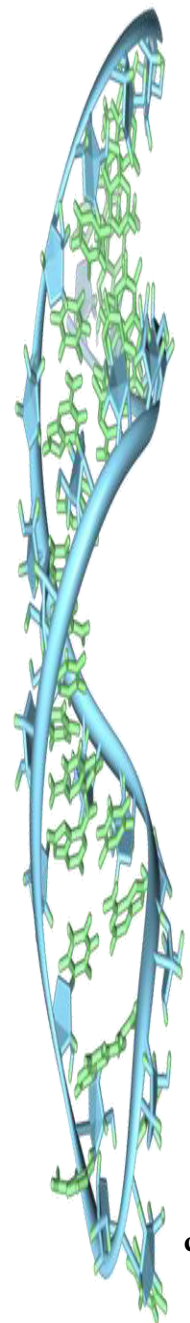
- Find out objective function values (fitness values) for all chromosomes.

x	$f(x)$
2.620	0.498
4.526	-0.983
0.001	0.001
1.900	0.946
0.922	0.797
0.580	0.548
1.170	0.921
2.171	0.825
2.493	0.604
3.385	-0.241

chromosomes fitness



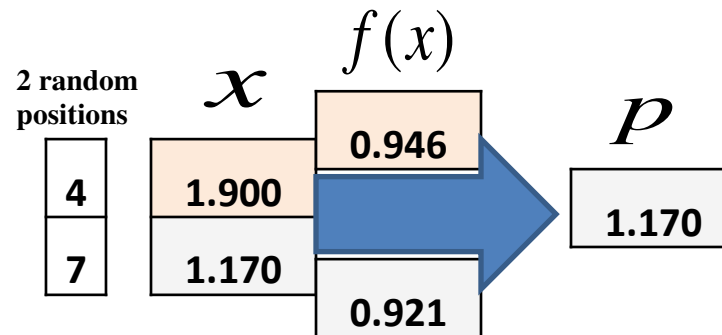
Tournament Selection:



x	$f(x)$
2.620	0.498
4.526	-0.983
0.001	0.001
1.900	0.946
0.922	0.797
0.580	0.548
1.170	0.921
2.171	0.825
2.493	0.604
3.385	-0.241

chromosomes fitness

Run this loop $N(=10)$ times



if ($f(x_1) \leq f(x_2)$)
 select - x_1 ,
 else
 select - x_2 ,

p
1.170
4.526
2.620
0.580
4.526
3.385
0.001
2.620
4.526
2.620

parents

Crossover:

Select Probability of Crossover $P_c = 0.8$

Distribution index for Crossover $(\mu) = 20$

Run this loop $P_c * N(0.8 * 10 = 8)$ times

2 random
positions

10
6

p

2.620
3.385

c_c

2.611
3.394

c_c

2.611
3.394
0.330
4.196
2.625
4.521
1.146
⋮
4.469

Generate random number – (r)

$$b = \begin{cases} (2 * r)^{\left(\frac{1}{\mu+1}\right)} & \text{if } r \leq 0.5 \\ \left(\frac{1}{2 * (1-r)}\right)^{\left(\frac{1}{\mu+1}\right)} & \text{if } r > 0.5 \end{cases}$$

$$c_{c1} = \frac{1}{2} ((1+b) * p_1 + (1-b) * p_2)$$

$$c_{c2} = \frac{1}{2} ((1-b) * p_1 + (1+b) * p_2)$$

p

1.170
4.526
2.620
0.580
4.526
3.385
0.001
2.620
4.526
2.620

parents

children

Mutation:

Select Probability of Mutation $P_m = 0.2$

Distribution index for Mutation $(\eta) = 20$

C_c

2.611
3.394
0.330
4.196
2.625
4.521
1.146
⋮
4.469

children

Run this loop $P_m * N_c * V(0.2 * 16 * 1 \cong 4)$ times

1 random
position

5

C_c

2.625

C_m

2.594

C_m

2.594

2.751

2.200

0.010

Mutated
children

Generate random number – (r)

$$d = \begin{cases} (2 * r)^{\left(\frac{1}{\eta+1}\right)} - 1 & \text{if } r \leq 0.5 \\ 1 - (2 * (1 - r))^{\left(\frac{1}{\eta+1}\right)} & \text{if } r > 0.5 \end{cases}$$

$$c_m = c_c + d$$

Selection of N chromosomes for next generation:

Combine chromosomes and children:
Intermediate population is

$$R = \begin{bmatrix} x \\ c_c \\ c_m \end{bmatrix}$$

Evaluate function

$$f(R) = \begin{bmatrix} f(x) \\ f(c_c) \\ f(c_m) \end{bmatrix}$$

Sort in
ascending
order according
to fitness value

$f(R)$

-0.985
-0.983
-0.982
-0.971
-0.912
-0.870
-0.418
-0.250
-0.241
⋮
0.946

R

4.539
4.526
4.521
4.470
4.291
4.196
3.572
3.394
3.385
⋮
1.899

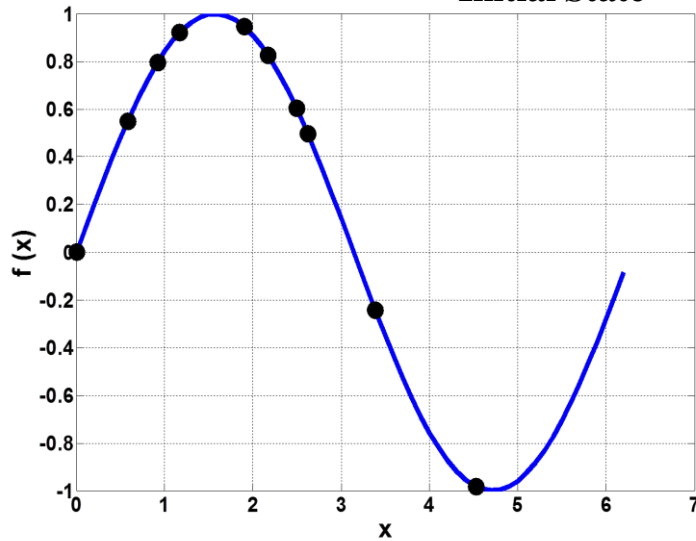
Select first $N (=10)$
chromosomes for next
generation

\mathcal{X}

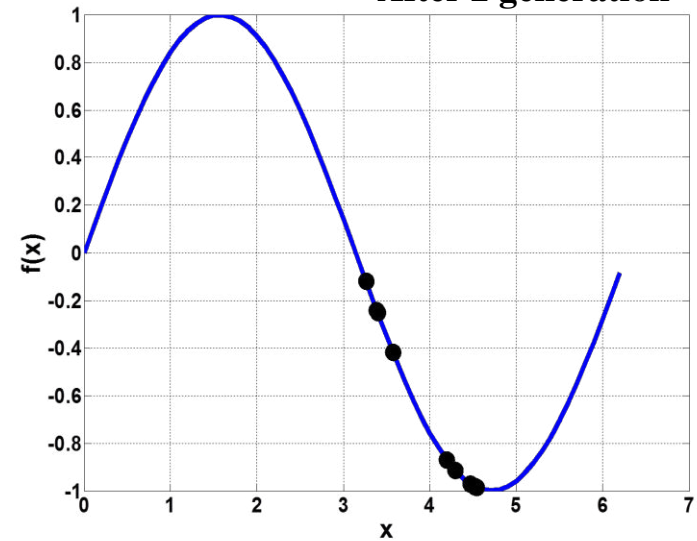
4.539
4.526
4.521
4.470
4.291
4.196
3.572
3.394
3.385
3.262

chromosomes

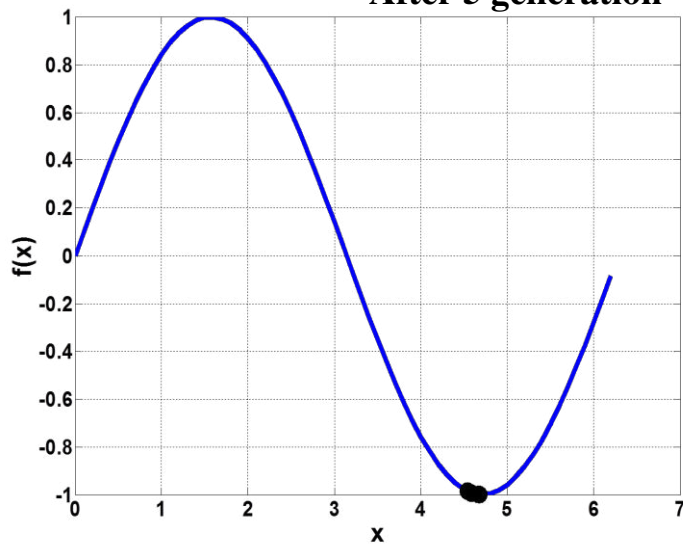
Initial State



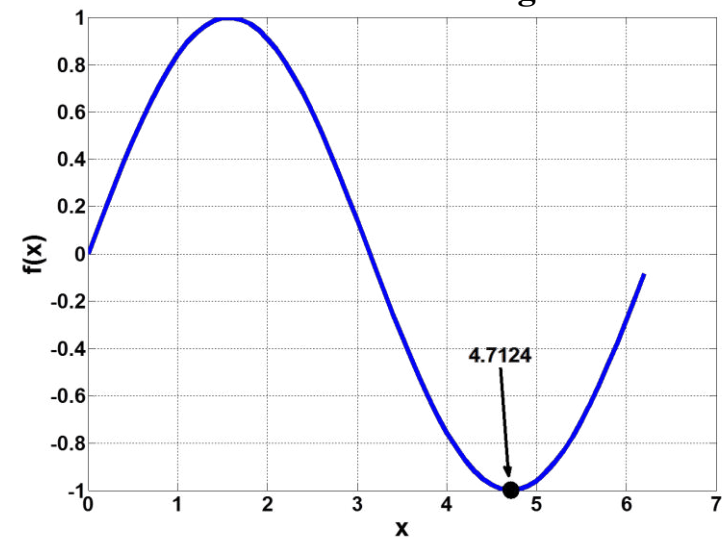
After 2 generation



After 5 generation



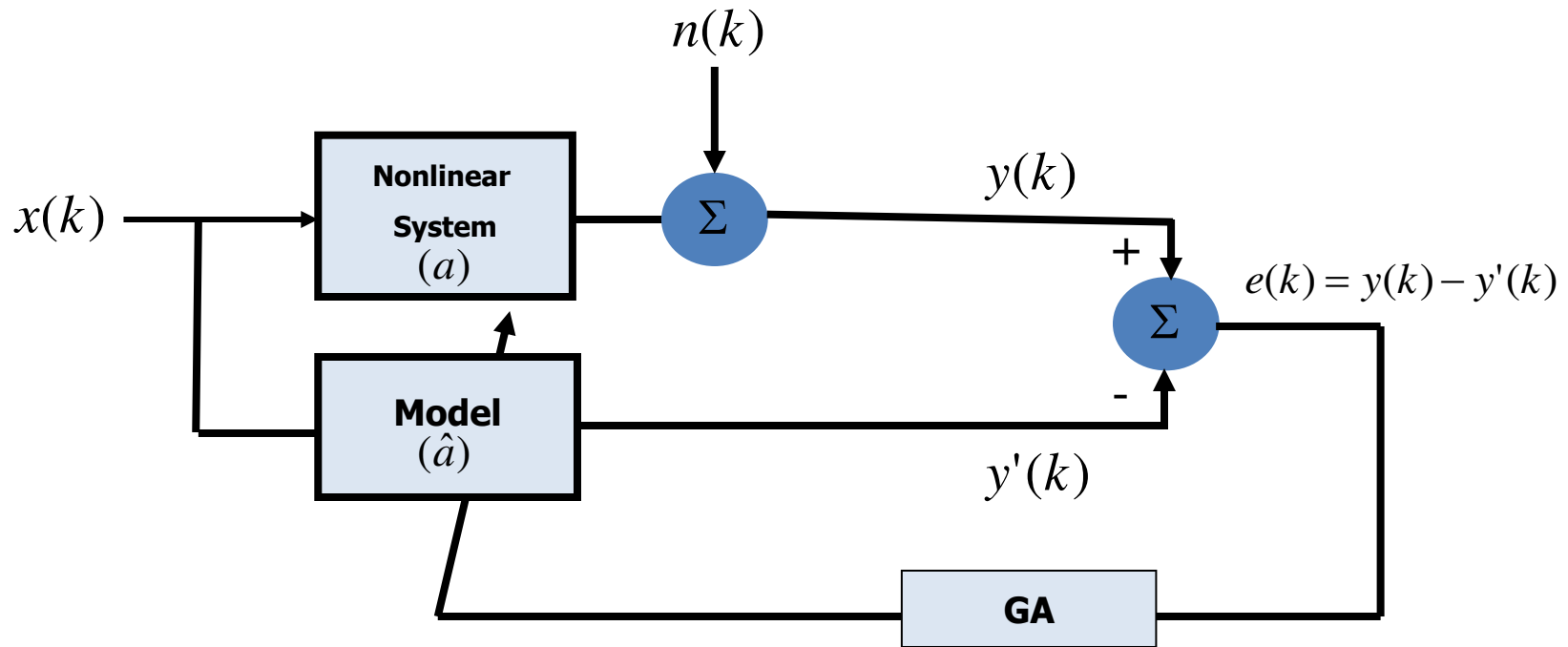
After 10 generation



Applications of Genetic Algorithms

- **Optimization**—numerical and combinatorial optimization problems, e.g. travelling salesman, routing, graph colouring and partitioning
- **Robotics**—trajectory planning
- **Machine learning**—designing neural networks, classification and prediction, e.g. prediction of weather or protein structure
- **Signal processing**—filter design
- **Design**—semiconductor layout, aircraft design, communication networks
- **Automatic programming**—evolve computer programs for specific tasks, design cellular automata and sorting networks
- **Economics**—development of bidding strategies, emergence of economics markets
- **Immune systems**—model somatic mutations
- **Ecology**—model symbiosis, resource flow

Schematic Block diagram of an adaptive identification system



x - represents the random input sequence lie in between $[-0.5, 0.5]$

y - represents the desired output sequence.

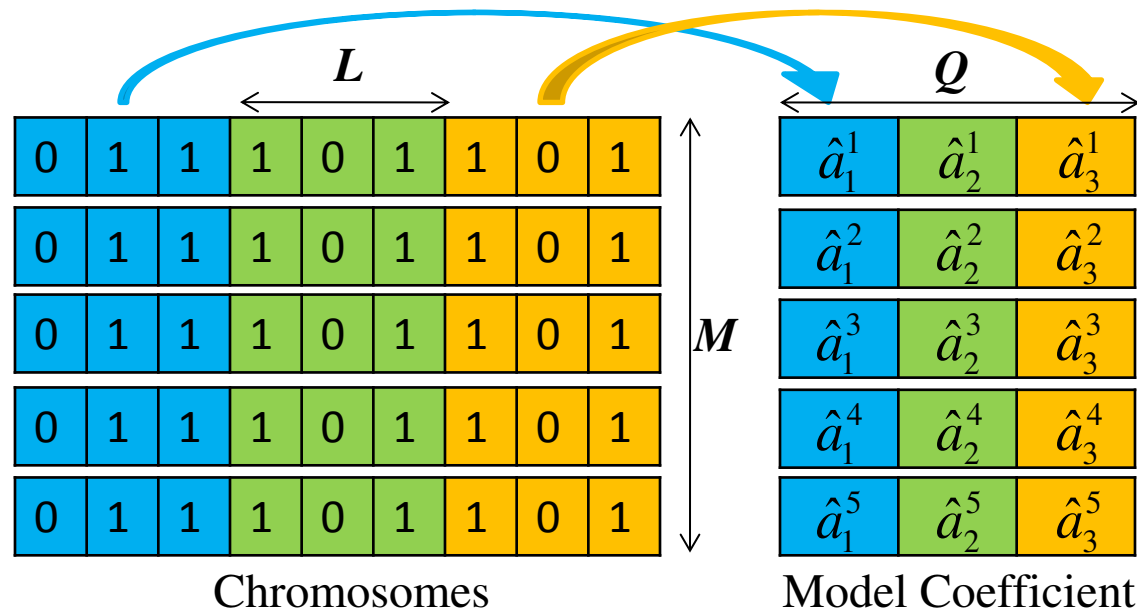
y' - represents estimated output sequence.

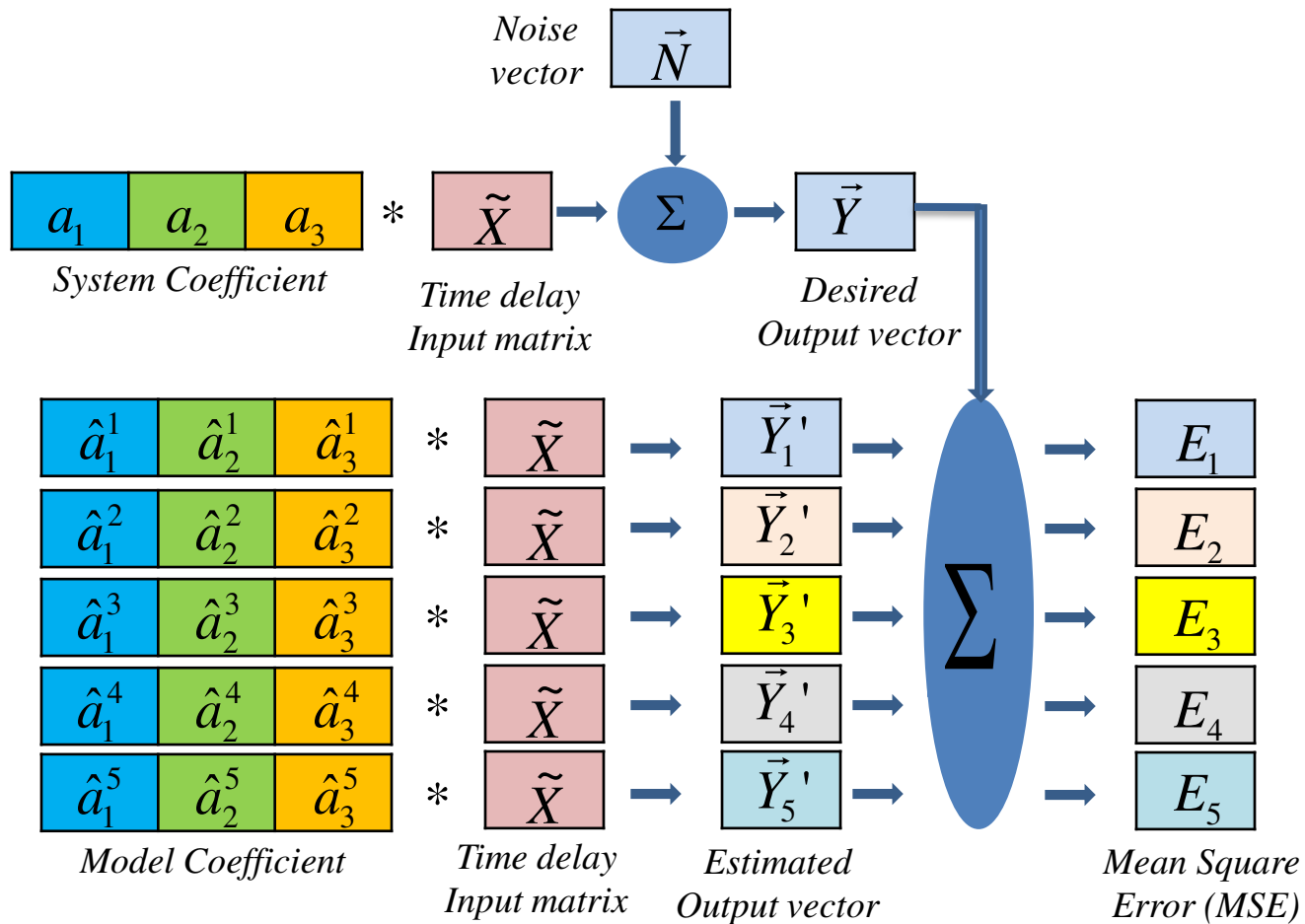
n - is additive white Gaussian noise.

a - represent coefficients of system.

Genetic Algorithm For System Identification

- ❖ Coefficients of the model (\hat{a}) are initially chosen from a population of M - chromosomes.
- ❖ Each chromosome constitutes ($Q * L$) number of random binary bits where each sequential group of L - bits represent one coefficient of the adaptive model, where Q is the number of parameters of the model.





$$\tilde{X} = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ 0 & x_1 & x_2 & \dots & x_{K-1} \\ 0 & 0 & x_1 & \dots & x_{K-2} \end{bmatrix}$$

$$\vec{Y} = [y_1, y_2, y_3, \dots, y_K]$$

$$\vec{N} = [n_1, n_2, n_3, \dots, n_K]$$

$$E_m = mse(e_1^m, e_2^m, \dots, e_K^m)$$

Simulation & Results

- ❖ Input Signal (x) :- zero mean uniformly distributed random sample lying between $[-0.5, 0.5]$
- ❖ Noise (N) :- White & Gaussian distributed random samples
- ❖ Coefficients of the FIR filter (Q):- 3
- ❖ Number of input signal samples (K) :- 500
- ❖ Binary coded GA is used
 - Number of chromosomes taken (M) :- 60
 - Number of bits in each chromosome ($Q*L$) :- 60
 - Using one point cross over
 - Using Tournament selection
 - Probability of Crossover (P_c) :- 0.8
 - Probability of Mutation (P_m) :- 0.1

Systems

Linear System

$$A(z) = 0.3040 + 0.9029z^{-1} + 0.3040z^{-2}$$

$$NSR = -30dB \text{ \& } NSR = -20dB$$

Nonlinear System

For simulation purpose the Nonlinear system can be obtained by passing the output of the linear system through various nonlinearity functions.

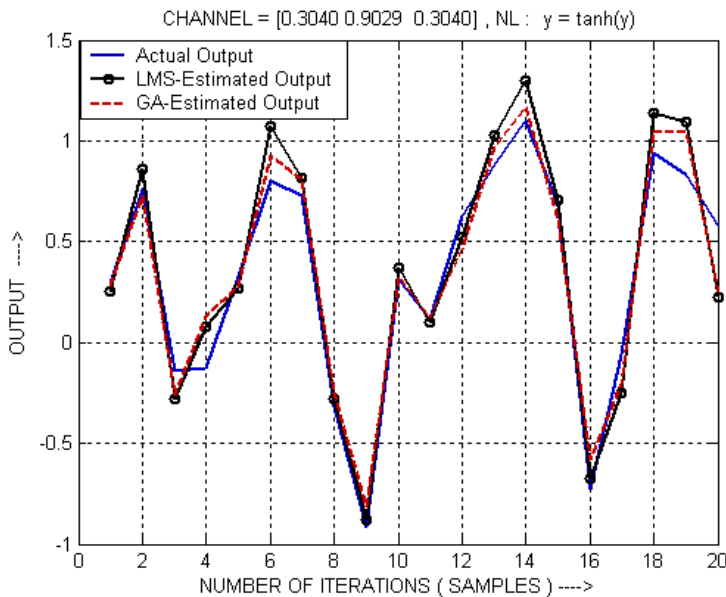
- EXP : - 1

$$Y = \tanh(Y),$$

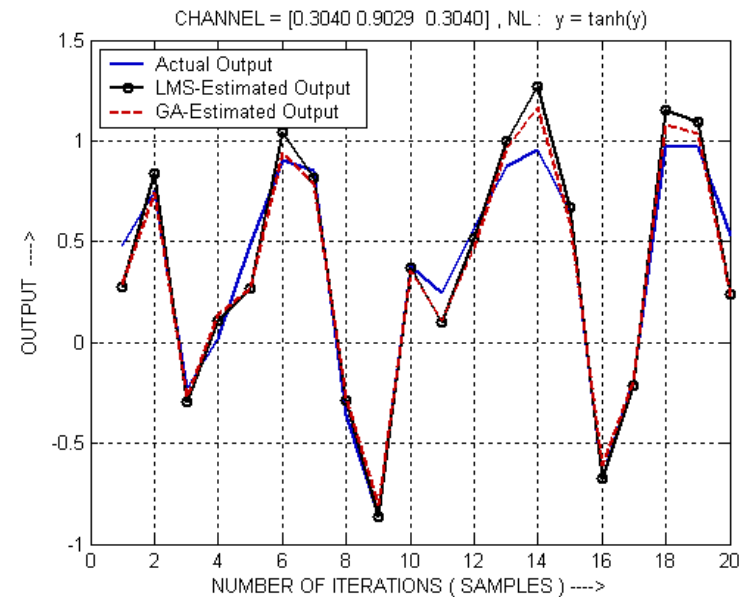
- EXP : - 2

$$Y = Y + 0.2Y^2 - 0.1Y^3$$

Comparison of output response of (Exp-1) at -20dB NSR & -30dB NSR

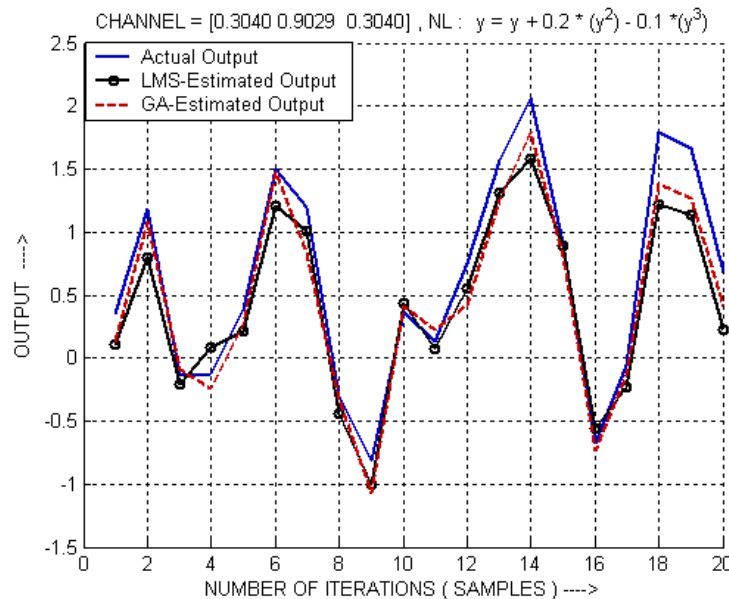


NSR = -20dB

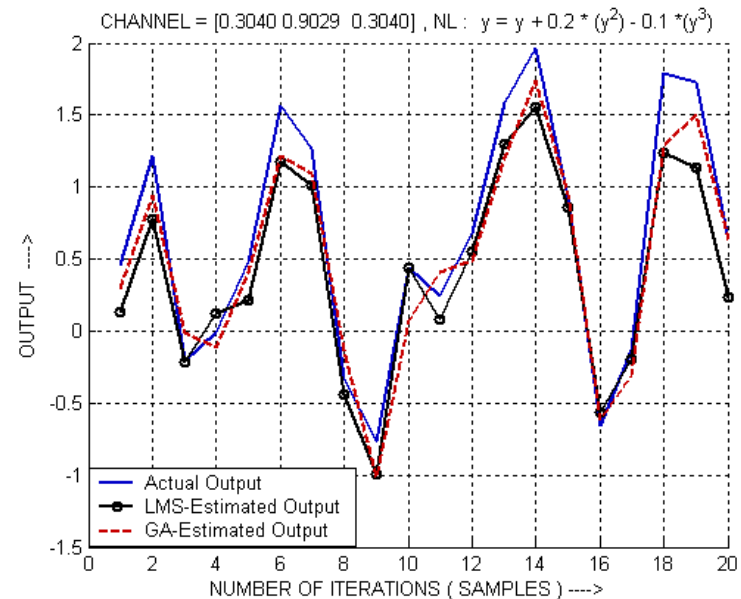


NSR = -30dB

Comparison of output response of (Exp-2) at -20dB NSR & -30dB NSR

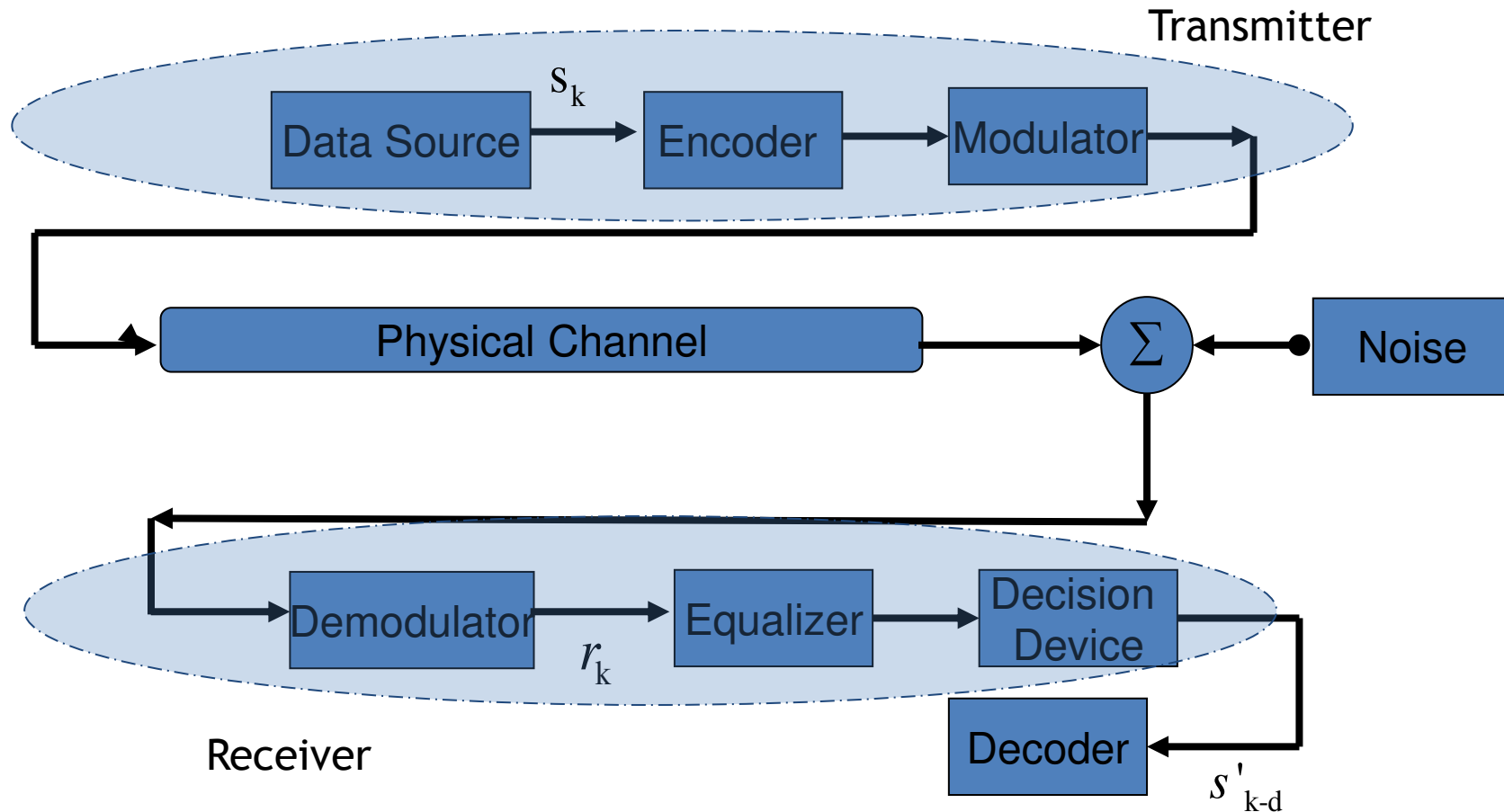


NSR = -20dB



NSR = -30dB

Overview of a Digital Communication System



Block Diagram of a Digital Communication System

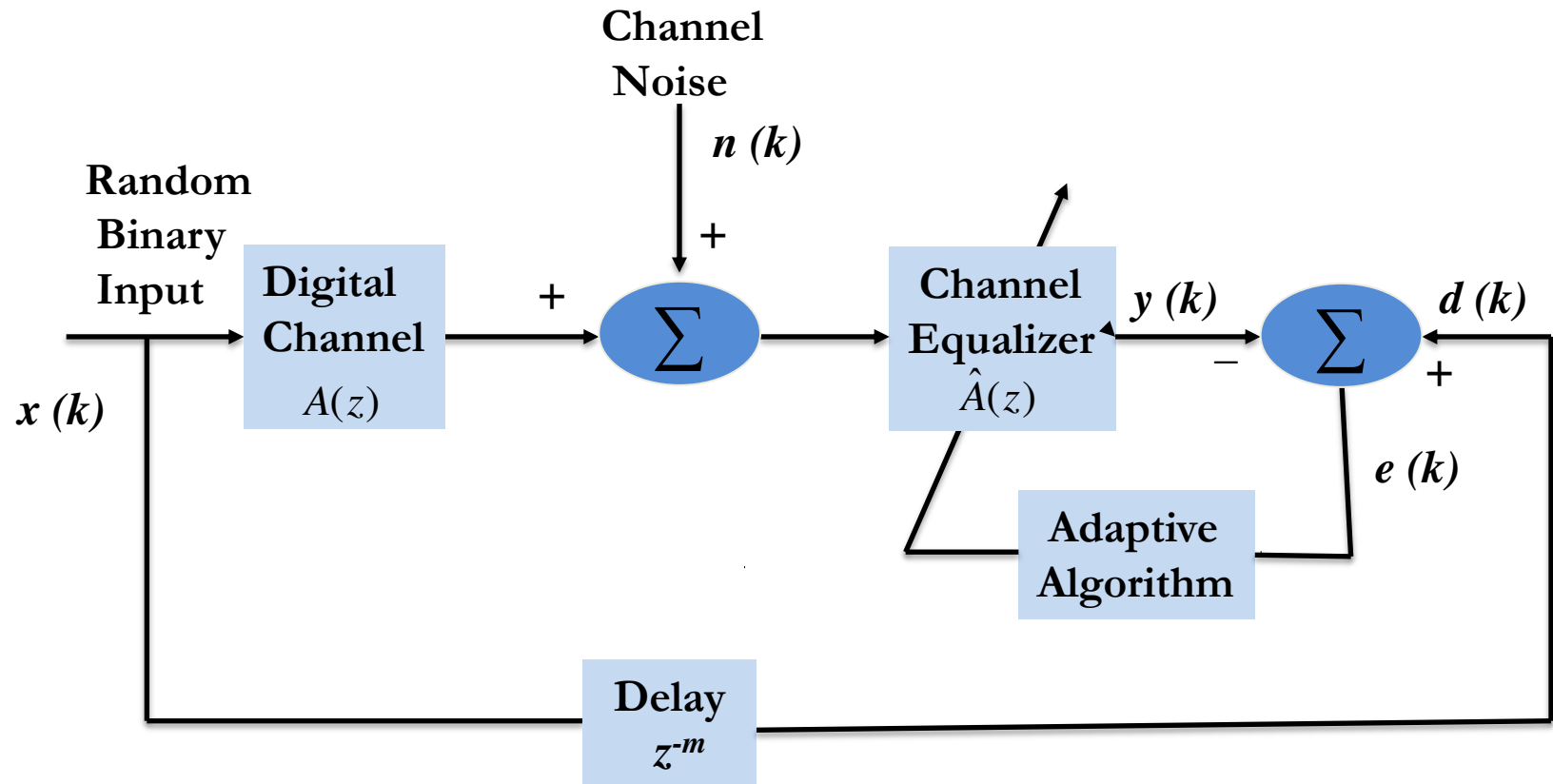
Digital Channel Equalizers

- ❖ Located in the front end of the receivers (in series with the channel)
- ❖ Inverse system of the channel model (Transfer function of the equalizer is inverse to the transfer function of the channel)

Use to reduce : -

- Inter-Symbol Interference (ISI)
- Inter User Interference in the form of Co-channel Interference (CCI)
- Adjacent Channel Interference (ACI) in the presence of Additive White Gaussian Noise (AWGN).

Block Diagram For Channel Equalization



Simulation & Results

- ❖ Binary coding is used
- ❖ Number of chromosomes taken (M) :- 60
- ❖ Number of bits used for each parameter (L) :- 20
- ❖ One point cross over is used
- ❖ Tournament selection is used
- ❖ Probability of Crossover (P_c) :- 0.8
- ❖ Probability of Mutation (P_m) :- 0.1
- ❖ Experiments on Nonlinear Channel

Nonlinear Channels

Linear System

$$A(z) = 0.3040 + 0.9029z^{-1} + 0.3040z^{-2}$$

$$NSR = -10dB$$

Nonlinear System

For simulation purpose the Nonlinear system can be obtained by passing the output of the linear system through various nonlinearity functions.

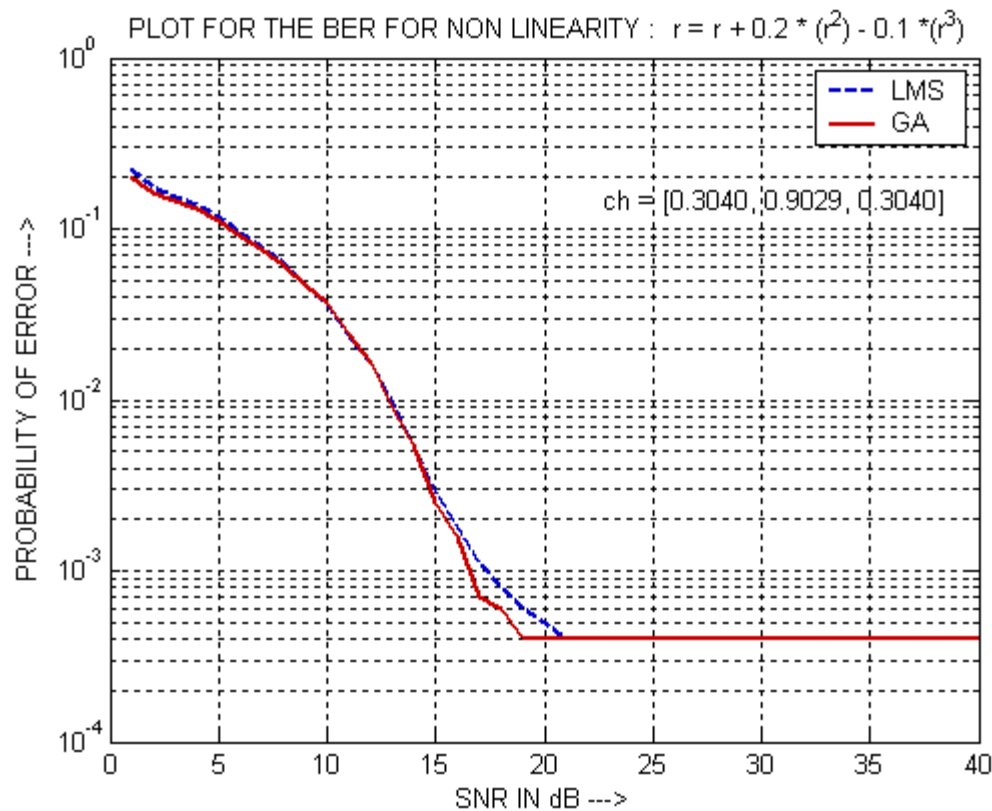
•EXP : - 1

$$r(k) = r(k) + 0.2 * r^2(k) - 0.1 * r^3(k),$$

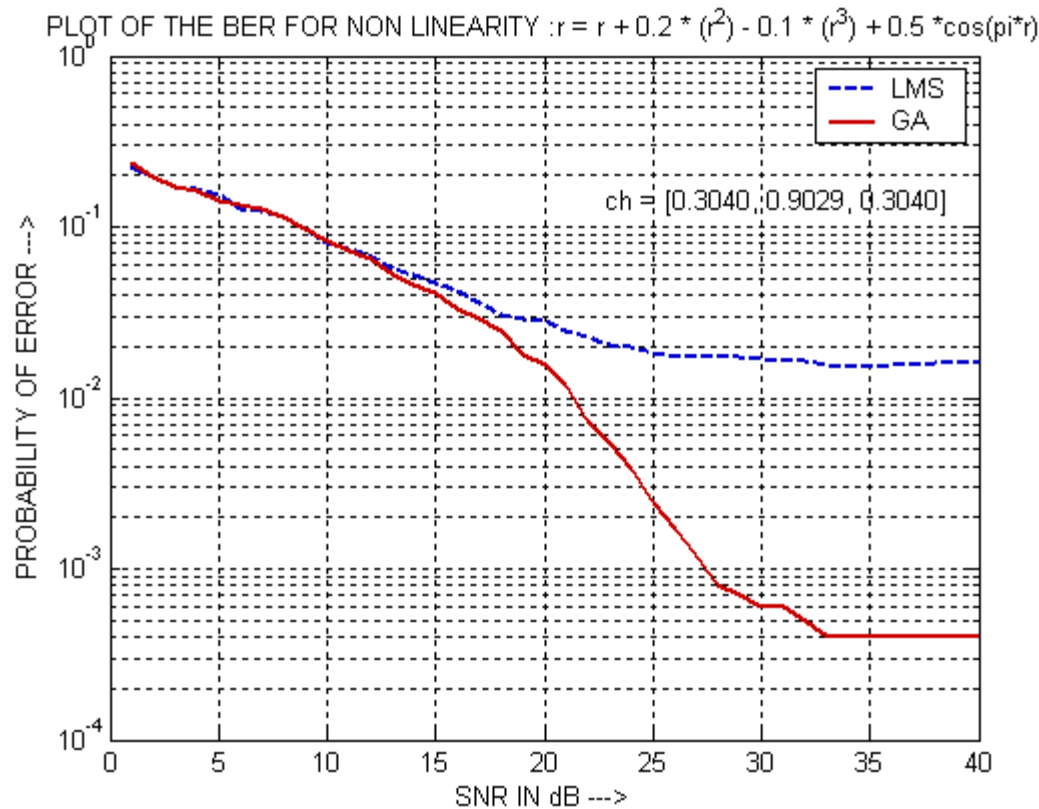
•EXP : - 2

$$r(k) = r(k) + 0.2 * r^2(k) - 0.1 * r^3(k) + 0.5 * \cos(\pi * r),$$

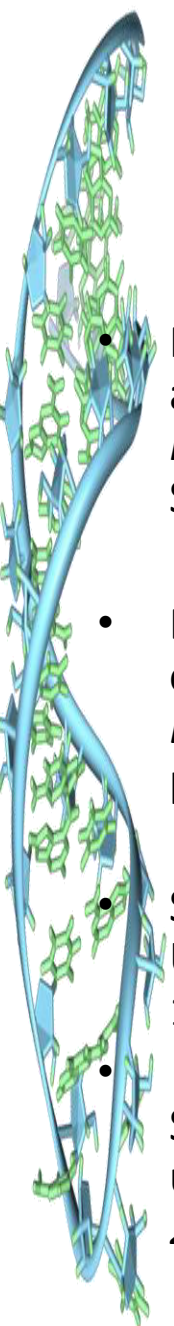
Comparison of BER of Nonlinear Channel NCH1 between LMS & GA based Equalizer at -10dB noise



Comparison of BER of nonlinear Channel NCH2 between LMS & GA based equaliser at -10dB noise



References

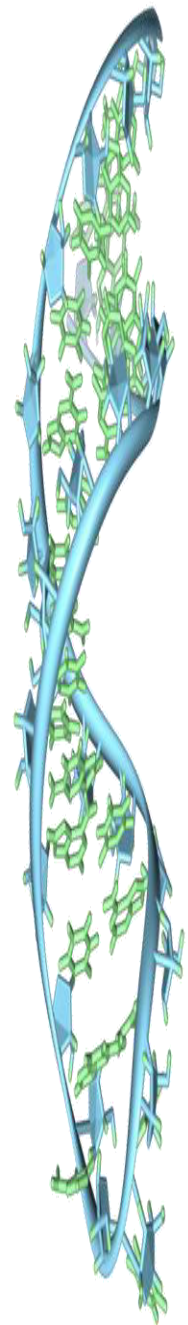
- 
- Mishra, S.K.; Panda, G.; Meher, S.; Majhi, R.; Singh, M., "Portfolio management assessment by four multiobjective optimization algorithm," *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE* , vol., no., pp.326,331, 22-24 Sept. 2011
 - Baghel, V.; Panda, G.; Srihari, P.; Rajarajeswari, K.; Majhi, B., "An efficient multi-objective pulse radar compression technique using RBF and NSGA-II," *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* , vol., no., pp.1291,1296, 9-11 Dec. 2009
 - Sahoo, A.K.; Panda, G.; Pradhan, P.M., "Generation of Pulse Compression Codes Using NSGA-II," *India Conference (INDICON), 2009 Annual IEEE* , vol., no., pp.1,4, 18-20 Dec. 2009
 - Sahoo, A.K.; Panda, G.; Pradhan, P.M., "Efficient design of pulse compression codes using multiobjective genetic algorithm," *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* , vol., no., pp.324,329, 9-11 Dec. 2009

References

- Acharya, D.P.; Panda, G.; Lakshmi, Y. V S, "Constrained genetic algorithm based independent component analysis," *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* , vol., no., pp.2443,2449, 25-28 Sept. 2007

■ Books:

- “The design of innovation : lessons from and for competent genetic algorithms “ by David E. Goldberg
- “Optimization for Engineering Design : Algorithms and Examples “ by Kalyanmoy Deb



Thank You

Civil Engineering

- Parameters Identification of Coupled Seepage and Stress Field Based on Genetic Algorithms
- An Adaptive Chromosome Technique of Genetic Algorithms to Identify Structural Damage
- Vehicle Velocity Recognition by Bridge Responses with a Pattern Search Algorithm
- Improved Seismic Control of Structure with Variable Friction Dampers by GA

Mechanical Engineering

- A hybrid genetic algorithm for constrained optimization problems in mechanical engineering
- Genetic algorithm and gradient-based algorithm optimization of vehicle turning mechanism
- Robot path planning using genetic algorithms
- Intelligent Learning Algorithms for Active Vibration Control