AUTUMN MID SEMESTER EXAMINATION-2024

School of Computer Engineering
Kalinga Institute of Industrial Technology, Deemed to be University
Computational Intelligence
[CS30011]

Time: 1 1/2 Hours **Full Mark: 20**

Answer Any four questions including question No.1 which is compulsory.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

1. Answer all the questions. [1 X 5]

a) Differentiate between hard computing and soft computing.
Ans - Conventional or hard computing requires a precisely stated analytical model and a lot of mathematical and logical computation. Soft computing differs from conventional or hard computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximate reasoning.

b) Consider a two-input single Adaline with the weights [3 2] and inputs [-5 7]^T. Find the bias required to generate an output of 0.5.
Ans - b = 1.5 (from the linear transfer function used by Adaline)

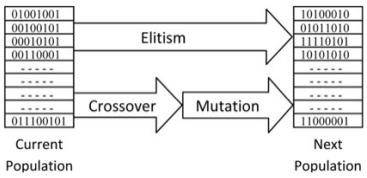
c) Test a perceptron with weights $w_1 = 1$, $w_2 = -0.8$ and bias = 0 against the following input/target pairs and update the weights and bias according to the perceptron learning rule.

x_1	x_2	t
1	2	1
-1	2	0
0	-1	0

Ans –
After 1st input, error is 1 so new parameters are w_1=1+1=2, w_2=-0.2+2=1.2, bias=0+1=1.
After 2nd input, error is -1 so new parameters are w_1=2+1=3, w_2=1.2-2=-0.8, bias=1-1=0.
After 3rd input, error is -1 so new parameters are w_1=3-0=3, w_2=-0.8+1=0.2, bias=0-1=-1.

d) Which activation function will you choose in your neuron model if you want a continuous output signal in the range of [-1, +1]? Justify your answer.
Ans – Hyperbolic tangent / bipolar sigmoid activation function.

e) Explain elitism in GA through an example.
Ans - Elitism is a strategy that preserves the best individuals from one generation to the next. This helps maintain the quality of the solutions and prevents the algorithm from converging too quickly to a suboptimal solution.

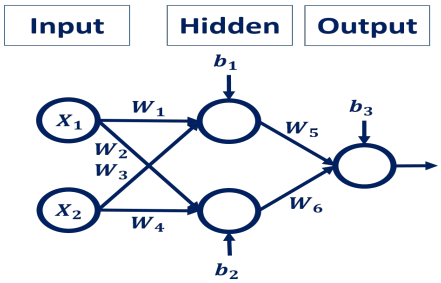


2. [2+3]

(a) Differentiate between ANN and BNN. Sketch the schematic architecture diagram of ANN and specify its components. Write its advantages and disadvantages.

Ans –

Parameters	**ANN**	**BNN**
Structure	input weight output hidden layer	dendrites synapse axon cell body
Learning	very precise structures and formatted data	they can tolerate ambiguity
Processor	complex high speed one or a few	simple low speed large number
Memory	separate from a processor localized non-content addressable	integrated into processor distributed content-addressable
Computing	centralized sequential stored programs	distributed parallel self-learning
Reliability	very vulnerable	robust
Expertise	numerical and symbolic manipulations	perceptual problems
Operating Environment	well-defined well-constrained	poorly defined un-constrained
Fault Tolerance	the potential of fault tolerance	performance degraded even on partial damage



Advantages

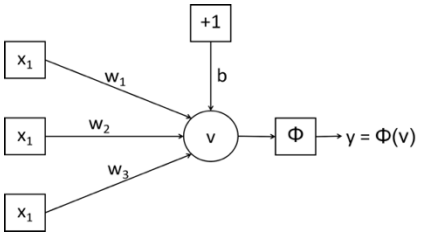
1. Large volumes of data may be utilized to train and generalize artificial neural networks. They may be trained using vast datasets, allowing them to make patterns-based predictions and judgments.
2. ANNs may be improved and employed efficiently on hardware accelerators or dedicated AI processors like ***GPUs*** and ***AI accelerators*** for quick and parallel processing.
3. Another advantage of ANN is that they continue to function even in the presence of noise or errors in data. As a result, they are appropriate in scenarios involving noisy, partial, or distorted data.
4. They are non-linear in nature as well. It enables them to represent complex data relationships and patterns. They can also be customized to handle various sorts of data and perform various activities.
5. They are capable of extracting features from data. It removes the need for manual feature editing. They can also be taught to handle many jobs at once. As a result, they may be utilized in advanced AI applications.

Disadvantages

1. Artificial neural networks may grow overly complex due to their architecture and the massive information used to train them. They can memorize the training data. It may result in a poor generalization of new data.
2. Artificial neural networks require suitable hardware components like central processors or dedicated AI accelerators, vast storage spaces, and massive random access memory.
3. Their working principles and even outcomes can be difficult to grasp because of the complexities of ANNs. Some people may find it difficult to comprehend their decision-making processes.
4. No explicit rule determines the structure of ANN. The proper network structure is obtained by trial and error.
5. They are also susceptible to adversarial instances or slight changes in input data. These modifications may cause the artificial neural network to make wrong decisions and produce irrelevant results.

(b) Consider the single McCulloch-Pitts artificial neuron given below. The node receives three-dimensional input patterns x = (x_1, x_2, x_3), each component of which are only binary signals (either 0 or 1). Suppose that the weights corresponding to the three inputs and bias have the following values: $w_1 = 2$, $w_2 = -4$, $w_3 = 1$, b = 1, and the activation of the unit is given by the hardlimit / step function:

$$y = \begin{cases} 1, & v \geq 0 \\ 0, & otherwise \end{cases}$$



Calculate the output value y of the neuron for each of the following input patterns:

Pattern	x_1	x_2	x_3
P_1	1	0	0
P_2	0	1	1
P_3	1	0	1
P_4	1	1	1

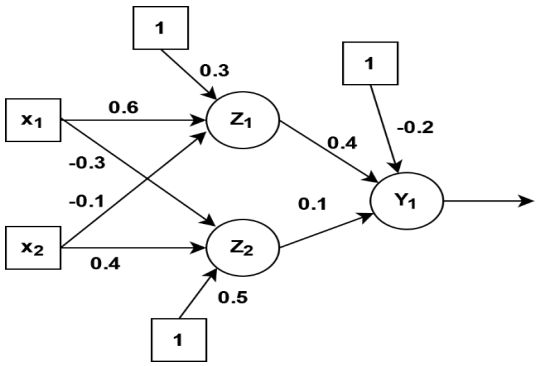
Ans –
The output of the given neuron can be expressed as follows:
y=w1x1+w2x2+w3x3+b
Given that, $w_1 = 2$, $w_2 = -4$, $w_3 = 1$, and b = 1.
For P1=x1,x2,x3=1,0,0,
yP1=2∗1 + −4∗0 + 1∗0 + 1=Φ(3)=1
For P2=x1,x2,x3=0,1,1,
yP2=2∗0 + −4∗1 + 1∗1 + 1=Φ(2)=0
For P3=x1,x2,x3=1,0,1,
yP3=2∗1 + −4∗0 + 1∗1 + 1=Φ(4)=1
For P4=x1,x2,x3=1,1,1,
yP4=2∗1 + −4∗1 + 1∗1 + 1=Φ(0)=1

3. [2+3]

(a) What is the role of transfer function in perceptron architecture? State the limitations of single-layer perceptron.
Ans –
The transfer function translates the input signals to output signals. The hard-limit transfer function gives a perceptron the ability to classify input vectors by dividing the input space into two regions.
Limitations –

- Limited to Linearly Separable Data
- Limited Memory Capacity and Range of Capabilities
- Difficulty with Nonlinear Mappings
- Vulnerable to overfitting
- Sensitive to noise

(b) Find the new weights for the following backpropagation network after presenting input [1, -1] once for the target [0]. Use log-sigmoid/Logistic Sigmoid activation function for the hidden and output layers. Assume a learning rate of 0.5.



Ans –

Input: [x1, x2] = [1, -1]

Target(t or d) = 0

Learning rate η= 0.5

Activation Function: Log-sigmoid = $f(x) = \frac{1}{1+e^{-x}}$

Choosing the notation for weight and bias

$V = \begin{bmatrix} v_{01} & v_{02} & v_{11} & v_{12} & v_{21} & v_{22} \end{bmatrix} = [0.3\ 0.5\ 0.6\ -0.3\ -0.1\ 0.4]$

$W = \begin{bmatrix} w_{01} & w_{11} & w_{21} \end{bmatrix} = [-0.2\ 0.4\ 0.1]$

Forward Pass:

Input to Hidden-Layer:

$z_{in1} = 1 * 0.3 + 1 * 0.6 + -1 * -0.1 = 1$

$z_{in2} = 1 * 0.5 + 1 * -0.3 + -1 * 0.4 = -0.2$

Output of Hidden Layer:

$z_1 = f(z_{in1}) = \frac{1}{1+e^{-z_{in1}}} = 0.7311$

$z_2 = f(z_{in2}) = \frac{1}{1+e^{-z_{in2}}} = 0.4502$

Input to Output Layer:

$y_{in1} = 1 * -0.2 + z_1 * 0.4 + z_2 * 0.1 = 0.1374$

Output of Output Layer:

$y_1 = f(y_{in1}) = \frac{1}{1+e^{-z_{in1}}} = 0.5343$

Backward Pass:

Local gradient/Error correction from output layer

$\delta_1^{(o)} \text{ or } \delta_1^{(2)} = (t_1 - y_1) f'(y_{in1})$

$\delta_1^{(o)} = (t_1 - y_1)\, y_1 (1 - y_1) = (0 - 0.5343) * 0.5343 * (1 - 0.5343) = -0.1329$

Weight correction between hidden and output layer

$\Delta b = \Delta w_{01} = \eta\, \delta_1^{(o)} 1 = 0.5 * -0.1329 * 1 = -0.0665$

$\Delta w_{11} = \eta\, \delta_1^{(o)} z_1 = 0.5 * -0.1329 * 0.7311 = -0.0486$

$\Delta w_{21} = \eta\, \delta_1^{(o)} z_2 = 0.5 * -0.1329 * 0.4502 = -0.0299$

Updated Weights

$w_{01}(new) = w_{01}(old) + \Delta w_{01}$

$w_{01}(new) = w_{01} + \Delta w_{01} = -0.2 + -0.0665 = -0.2665$

$w_{11}(new) = w_{11} + \Delta w_{11} = 0.4 + -0.0486 = 0.3514$

$w_{21}(new) = w_{21} + \Delta w_{21} = 0.1 + -0.0299 = 0.701$

Local gradient/Error correction from hidden layer

$\delta_{in1} = \delta_1^{(o)} * w_{11} = -0.1329 * 0.4 = -0.0532$

$\delta_{in2} = \delta_1^{(o)} * w_{21} = -0.1329 * 0.1 = -0.0133$

$\delta_1^{(h)} \text{ or } \delta_1^{(1)} = \delta_{in1} * f'(z_{in1}) = \delta_{in1} * z_1 * (1 - z_1) = -0.0532 * 0.7311 * (1 - 0.7311) = -0.0105$

$\delta_2^{(h)} \text{ or } \delta_2^{(1)} = \delta_{in2} * f'(z_{in2}) = \delta_{in2} * z_2 * (1 - z_2) = -0.0133 * 0.4502 * (1 - 0.4502) = -0.0033$

Weight Correction between Input and Hidden Layer:

$\Delta v_{01} = \eta\, \delta_1^{(h)} 1 = 0.5 * -0.0105 * 1 = -0.0053$

$\Delta v_{11} = \eta\, \delta_1^{(h)} x_1 = 0.5 * -0.0105 * 1 = -0.0053$

$\Delta v_{21} = \eta\, \delta_1^{(h)} x_2 = 0.5 * -0.0105 * -1 = 0.0053$

$\Delta v_{02} = \eta\, \delta_2^{(h)} 1 = 0.5 * -0.0033 * 1 = -0.0017$

$\Delta v_{12} = \eta\, \delta_2^{(h)} x_1 = 0.5 * -0.0033 * 1 = -0.0017$

$\Delta v_{22} = \eta\, \delta_2^{(h)} x_2 = 0.5 * -0.0033 * -1 = 0.0017$

Updated Weights

$$v_{01}(new) = v_{01}(old) + \Delta v_{01}$$

$$v_{01}(new) = v_{01} + \Delta v_{01} = 0.3 + -0.0053 = 0.2947$$

$$v_{11}(new) = v_{11} + \Delta v_{11} = 0.6 + -0.0053 = 0.5947$$

$$v_{21}(new) = v_{21} + \Delta v_{21} = -0.1 + 0.0053 = -0.0947$$

$$v_{02}(new) = v_{02} + \Delta v_{02} = 0.5 + -0.0017 = 0.4983$$

$$v_{12}(new) = v_{12} + \Delta v_{12} = -0.3 + -0.0017 = -0.3017$$

$$v_{22}(new) = v_{22} + \Delta v_{22} = 0.4 + 0.0017 = 0.4017$$

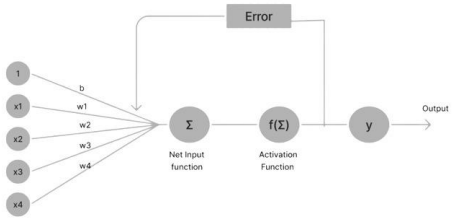
4. [2+3]

(a) Explain the Adaline model. Implement AND gate using Adaline network with a learning rate of 0.1.

Assume initial weights of [0.2 0.3] and an initial bias of 0.1.

Ans –

A network with a single linear unit is called Adaline (Adaptive Linear Neural). A unit with a linear activation function is called a linear unit. In Adaline, there is only one output unit and output values are bipolar (+1,-1). Weights between the input unit and output unit are adjustable. It uses the delta rule. The learning rule is found to minimize the mean square error between activation and target values. Adaline consists of trainable weights, it compares actual output with calculated output, and based on error training algorithm is applied.



AND GATE Implementation

Assume: learning rate = 0.1
Tolerance = 0.1

Assume a two inputs AND Gate, which is true only if all the inputs are true.
So, inputs are I1, I2 and Bias(B)

Activation Function:
Y = 1 ; y_in >= 0
= -1 ; y_in < 0

Initialization:
Set weights to small random values to each of the input.
w1 = 0.2 w2 = 0.3 b = 0.1

First Cycle:
# Run 1
Inputs = [1, 1, 1]
Y_in = 0.1 * 1 + 0.2 * 1 + 0.3 * 1 = 0.6
b(new) = 0.1 + 0.1 * (1 - 0.6) = 0.14
w1(new) = 0.2 + 0.1 * (1 - 0.6) * 1 = 0.24
w2(new) = 0.3 + 0.1 * (1 - 0.6) * 1 = 0.34
So,
Largest weight change = 0.04

First Cycle:
# Run 2
Inputs = [1, 1, -1]
Y_in = 0.04
b(new) = 0.04
w1(new) = 0.14
w2(new) = 0.44
So,
Largest weight change = 0.1

First Cycle:
# Run 3
Inputs = [1, -1, 1]
Y_in = 0.34
b(new) = -0.09
w1(new) = 0.27
w2(new) = 0.31
So,
Largest weight change = 0.13

First Cycle:
# Run 4
Inputs = [1, -1, -1]
Y_in = -0.67
b(new) = -0.27
w1(new) = 0.43
w2(new) = 0.47
So,
Largest weight change = 0.16

All the runs in the first cycle do not have largest weight change < tolerance value, we proceed to second cycle.

Second Cycle:
# Run 1
Inputs = [1, 1, 1]
Y_in = 0.63
b(new) = -0.233
w1(new) = 0.46
w2(new) = 0.5
So,
Largest weight change = 0.03

Second Cycle:
# Run 2
Inputs = [1, 1, -1]
Y_in = -0.273
b(new) = -0.3
w1(new) = 0.39
w2(new) = 0.57
So,
Largest weight change = 0.07

Second Cycle:
# Run 3
Inputs = [1, -1, 1]
Y_in = -0.12
b(new) = -0.38
w1(new) = 0.47
w2(new) = 0.48
So,
Largest weight change = 0.09

Second Cycle:
# Run 4
Inputs = [1, -1, -1]
Y_in = -1.33
b(new) = -0.34
w1(new) = 0.43
w2(new) = 0.44
So,
Largest weight change = 0.04

In second cycle, largest weight change across all the training samples is less than the tolerance value.

So, the solution is:

B = -0.34, W1 = 0.43, W2 = 0.44

Note – Students can assume a tolerance value of their choice. At least one complete cycle needs to be shown for award of full marks.

(b) Describe with a proper diagram how the RBF network works. Mention its merits and demerits. For which kind of applications would you prefer RBF networks over MLP networks?

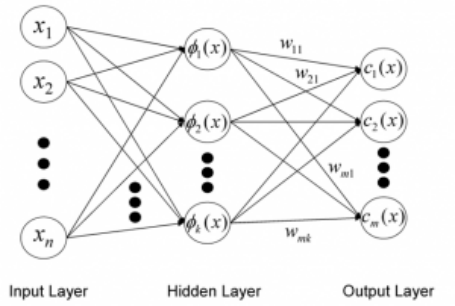
Ans –

Radial Basis Functions (RBFs) are a special category of feed-forward neural networks comprising three layers:

1. **Input Layer**: Receives input data and passes it to the hidden layer.
2. **Hidden Layer**: The core computational layer where RBF neurons process the data.
3. **Output Layer**: Produces the network's predictions, suitable for classification or regression tasks.

Here's how RBF Networks operate:

1. **Input Vector**: The network receives an n-dimensional input vector that needs classification or regression.
2. **RBF Neurons**: Each neuron in the hidden layer represents a prototype vector from the training set. The network computes the Euclidean distance between the input vector and each neuron's center.
3. **Activation Function**: The Euclidean distance is transformed using a Radial Basis Function (typically a Gaussian function) to compute the neuron's activation value. This value decreases exponentially as the distance increases.
4. **Output Nodes**: Each output node calculates a score based on a weighted sum of the activation values from all RBF neurons. For classification, the category with the highest score is chosen.



If the problem involves localized patterns or requires faster training on simpler functions, an RBF network might be the better choice. However, for more complex tasks that require deeper learning and feature extraction, a conventional multi-layer feed-forward network would likely be more effective. Ultimately, the decision should be guided by the specific characteristics of the data and the requirements of the task.

Advantages of RBF Networks

1. **Simplicity**: RBF networks often have simpler architectures and can be faster to train for specific types of problems, especially when the relationships in the data are smooth and localized.
2. **Fast Learning**: They typically involve less complex training processes since the hidden layer can be trained using unsupervised methods to determine the centers, followed by a linear adjustment in the output layer.

3. **Localized Activation**: RBF networks are particularly effective for data with localized patterns, as the radial basis functions allow them to focus on nearby input points.
4. Good for Function Approximation: They excel in approximating smooth functions and can perform well in scenarios where the input-output mapping is relatively straightforward.

5. [2+3]

(a) Briefly explain the characteristics of derivative free optimization.

Ans –

1 Derivative freeness:- repeated evaluation of objective function

2 Intuitive guidelines:- concepts are based on nature's wisdom, such as evolution and thermodynamics

3 Slower

4 Flexibility

5 Randomness:- global optimizers

6 Analytic Opacity:-knowledge about them are based on empirical studies

7 Iterative nature

(b) Apply the steps of GA to maximize the function:

$y = 100 – x^2$ where x is an integer, $-10 \le x \le 10$

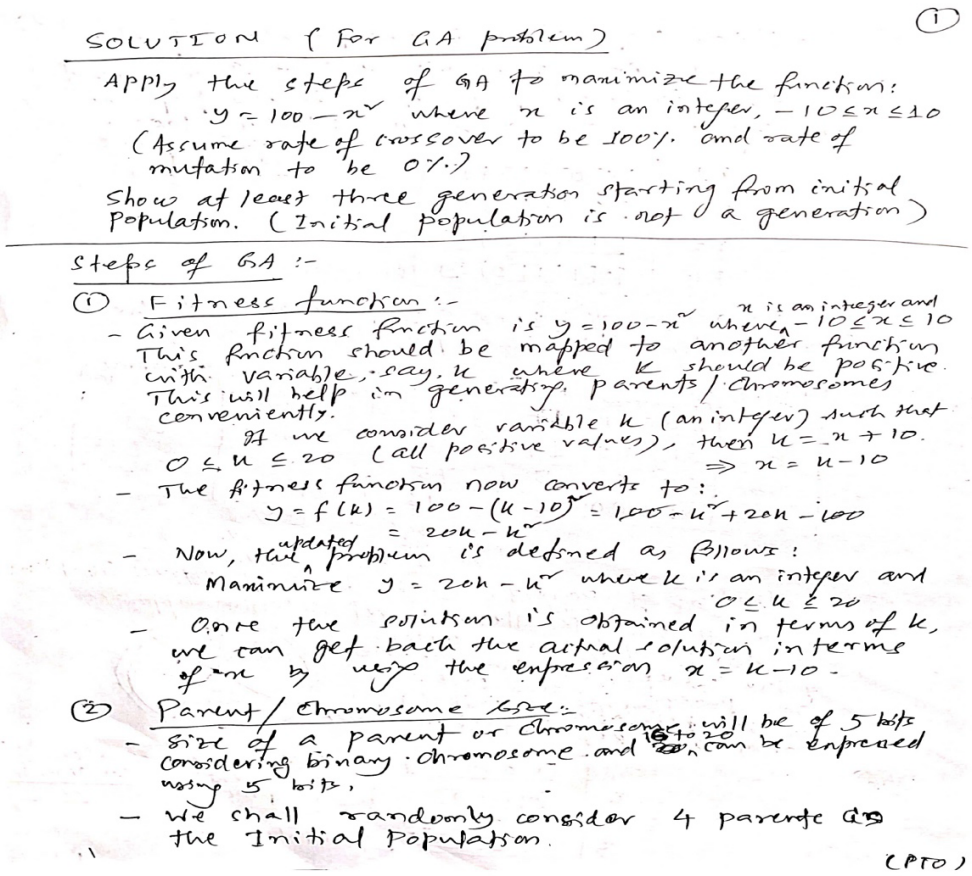
Assume : (i) rate of crossover to be 100% and rate of mutation to be 0%

(ii) 5-bit binary chromosomes

(iii) population size of 4

Show at least three generations starting from initial population. (Initial population is not a generation.)

Ans –



SOLUTION (For GA problem) (1)

Apply the steps of GA to maximize the function:
$y = 100 - x^2$ where x is an integer, $-10 \le x \le 10$
(Assume rate of crossover to be 100% and rate of mutation to be 0%.)
Show at least three generation starting from initial Population. (Initial population is not a generation)

Steps of GA :-

(1) Fitness function :-
- Given fitness function is $y = 100 - x^2$ where x is an integer and $-10 \le x \le 10$. This function should be mapped to another function with variable, say, k where k should be positive. This will help in generating parents / chromosomes conveniently.
If we consider variable k (an integer) such that $0 \le k \le 20$ (all positive values), then $k = x + 10$.
$\Rightarrow x = k - 10$
- The fitness function now converts to:
$y = f(k) = 100 - (k-10)^2 = 100 - k^2 + 20k - 100$
$= 20k - k^2$
- Now, the updated problem is defined as follows:
Maximize $y = 20k - k^2$ where k is an integer and $0 \le k \le 20$
- Once the solution is obtained in terms of k, we can get back the actual solution in terms of x by using the expression $x = k - 10$.

(2) Parent / Chromosome size :-
- Size of a parent or chromosome will be of 5 bits considering binary chromosome and 0 to 20 can be expressed using 5 bits.
- We shall randomly consider 4 parents as the Initial Population.

(PTO)

(3) Initial Population in the population pool :-

- We shall randomly select four parents as the initial Population in the population/mating pool), say, as follows : (We may toss a coin 20 times, considering Head = 1, Tail = 0)

P1 | 1 | 0 | 0 | 0 | 1 |
P2 | 0 | 1 | 1 | 1 | 0 |
P3 | 0 | 0 | 1 | 0 | 1 |
P4 | 0 | 1 | 1 | 0 | 0 |
(Population Pool)

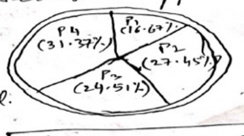
(4) Evaluation of fitness of parents and their contribution in the population pool :

Parent #	Binary chromosome	K	Fitness value $(20K-K^2)$	% contribution of fitness	REMARK (Range Acceptance)
P1	10001	17	51	16.67%	✓
P2	01110	14	84	27.45%	✓
P3	00101	5	75	24.51%	✓
P4	01100	12	96	31.37%	✓

Total : 306 → 100%
Max : 96 → 31.37%
Avg : 76.5 → 25%

(5) Selection of parents on the basis of fitness contribution :

- We shall use Biased Roullette wheel selection approach
- considering the biased sectors of the wheel, we shall not reject any parent in this case i.e. all the Parents of initial population are kept intact i.e selected.



(6)
P1 | 1 | 0 | 0 | 0 | 1 |
P2 | 0 | 1 | 1 | 1 | 0 |
P3 | 0 | 0 | 1 | 0 | 1 |
P4 | 0 | 1 | 1 | 0 | 0 |
(Population pool)

(*) These are also parents going for 1st generation processing

(PTO)

1st Generation Processing :-

(6) CROSSOVER :-

Parent #	Selected chromosome	K	Fitness value $(20K-K^2)$	% contribution of fitness	Crossover	offspring/children	
P_1'	10:001	17	51	16.67%	10:001	10110	(C_1)
P_2'	01:110	14	84	27.45%	01:110	01001	(C_2)
P_3'	001:01	5	75	24.51%	001:01	00100	(C_3)
P_4'	011:00	12	96	31.37%	011:00	01101	(C_4)

TOTAL: 306 → 100%

[(*) Crossover points (CP) are selected randomly for pair of parents. We select, say, CP2 for Parents P_1' & P_2' and say, CP3 for Parents P_3' & P_4'. We go for single point crossover as the chromosome sizes are small]

- Children generated becomes new parents of population pool.

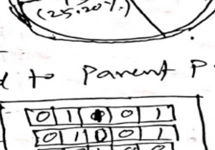
Children #	Parent #	Binary Chromosome	K	Fitness value $(20K-K^2)$	% contribution of fitness	REMARK (Range Acceptance)
C1	P1	10110	22	(Not Applicable)	(Rejected as it is beyond range)	X (REJECTED)
C2	P2	01001	9	99	38.97%	✓
C3	P3	00100	4	64	25.20%	✓
C4	P4	01101	13	91	35.83%	✓

(*) Parent P1 is rejected as its value is beyond the range

Total : 254 → 100%
Max : 99 → 38.97%
Avg : 84.67 → 33.33%

(7) Selection of Parents on the basis of fitness contribution

- We shall use Biased Roullettee wheel selection approach
- considering the biased sectors of the wheel, we shall not reject any parent in this case.
- Since we need 4 parents whereas there are now 3 parents available in the population pool we shall randomly make a copy of either parent P_2 or P_4 as their fitness values are better than compared to parent P3. Let us copy Parent P4.



- So the selected parents are ⇒
- These parents now go to 2nd generation processing.

P4 | 0 | 1 | 1 | 0 | 1 |
P2 | 0 | 1 | 0 | 0 | 1 |
P3 | 0 | 0 | 1 | 0 | 0 |
P4 | 0 | 1 | 1 | 0 | 1 |

(PTO)

2nd Generation Processing :- (4)

(8) CROSSOVER :-

Parent #	Binary chromosome	K	Fitness value ($20K - K^2$)	% Contribution of fitness	Crossover	Offsprings/ Children	
P_1^2	01101	13	91	26.38%	01|101	01001	(C1)
P_2^2	01001	9	99	28.70%	01|001	01101	(C2)
P_3^2	00100	4	64	18.55%	001|00	00101	(C3)
P_4^2	01101	13	91	26.37%	011|01	01100	(C4)

TOTAL: 345 → 100%

[(*) we select, say, CP 2 for Parents P_1^2 & P_2^2 and say, CP3 for parents P_3^2 & P_4^2]

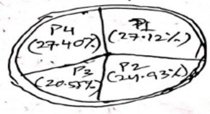
- children generated becomes new parents of population pool.

Children #	Parent #	Binary chromosome	K	Fitness value ($20K-K^2$)	% contribution of fitness	REMARK (Range Acceptance)
C1	P1	01001	9	99	27.12%	✓
C2	P2	01101	13	91	24.93%	✓
C3	P3	00101	5	75	20.55%	✓
C4	P4	01100	10	100	27.40%	✓

Total: 365 → 100%
Max: 100 → 27.40%
Avg: 91.25 → 25%

(9) Selection of parents on the basis of fitness contribution:

- we shall use Biased Roulette wheel selection approach
- considering the biased sectors of the wheel we shall not reject any parent in this case.
- so the selected Parents are:



P1	01001
P2	01101
P3	00101
P4	01100

- These parents now go to 3rd Generation Processing

(PTO)

3rd Generation Processing :- (5)

(10) CROSSOVER :- (we can randomly change the parent pairs in order to avoid similar results)

Parent #	Binary Chromosome	K	Fitness value ($20K-K^2$)	% contribution of fitness	Crossover	Offsprings/ Children	
P_1^3	01001	9	99	27.12%	01|001	01100	(C1)
P_4^3	01100	10	100	27.40%	01|100	01001	(C2)
P_3^3	00101	5	75	20.55%	001|01	00101	(C3)
P_2^3	01101	13	91	24.93%	011|01	01101	(C4)

TOTAL: 365 100%

[(*) we select, say, CP2 for parents P_1^3 & P_4^3 and say, CP3 for parents P_3^3 & P_2^3]

- Children generated becomes new parents of population pool.

Children #	Parent #	Binary Chromosome	K	Fitness value ($20K-K^2$)	% contribution of fitness	REMARK (Range Acceptance)
C1	P1	01100	10	100	27.40%	✓
C2	P2	01001	9	99	27.12%	✓
C3	P3	00101	5	75	20.55%	✓
C4	P4	01101	13	91	24.93%	✓

Total: 365 → 100%
Max: 100 → 27.40%
Avg: 91.25 → 25%

(9) Selection of parents on the basis of fitness contribution

- The outcome of Generation (2) & Generation (3) remains same after using the Biased Roulette wheel approach.
- All the parents can be selected for next generation

OBSERVATION :- The solution for this optimal problem ($20K-K^2$) happens at $K=10$ i.e. Maximization of $20K-K^2$

- In other words, solution for the given problem i.e. maximization of $100-x^2$ happens at $x=0$
- This solution has already achieved in both 2nd & 3rd generation.

$x = K - 10 = 10 - 10 = 0$

- NOTE :- In actual practice, solution may not be reached within these small iterations. Marks shall be given on the basis of approach.