Research Journal of Mathematical and Statistical Sciences ______________________________ISSN 2320-6047
Vol. **13(3),** 14-19, September **(2025)** *Res. J. Mathematical and Statistical Sci.*

A Genetic Algorithm approach for Optimization Problems

H.L. Sharma[1*], Vijayshankar Shukla[2] and Varsha Shukla[3]
[1]Department of Mathematics & Statistics, J.N. Agricultural University, Jabalpur, MP, India
[2]Computer Science and Engineering, Government Autonomous College, Satna, MP, India
[3]Comptroller Office, J.N. Agricultural University, Jabalpur, MP, India
drhlsharma_jnkvv@rediffmail.com

Available online at: www.isca.in , www.isca.me
Received 19th April 2025, revised 28th June 2025, accepted 3rd September 2025

Abstract

This paper is concerned with a genetic algorithm approach for optimization problems considering an equality whose coefficients are chosen in such a way that they would represent the bits of genetic algorithms for minimization including six chromosomes of length three applying the operator cross over and mutation while a cubic function has been considered for maximization. In both cases, the fitness value of the population seems to be adequate and found satisfactorily well at least in one generation. These have been illustrated with two numerical examples added at the end.

Keywords: Genetic algorithm, Optimization, Crossover, mutation, Offspring, Cubic function, bits.

Introduction

Genetic algorithms are considered to be a heuristic domain algorithm on the basis of Darwinian's evolutionary ideas of natural selection and genetics including the basic concept of genetic algorithms to create designing processes in natural system which is essential for evolution. Moreover, it represents an intelligent exploitation of a random search within a defined search dimension to solve a variety of problems given by John Holland[1]. He developed this idea in his book "Adaptation in natural and artificial systems". He described how to apply the principles of natural evolution to optimization problems and built the first Genetic Algorithms. Holland's theory has been further developed and now Genetic Algorithms (GAs) stand up as a powerful tool for solving search and optimization problems. Genetic algorithms are based on the principle of genetics and evolution. Which has been widely experimented studied and applied in many fields especially in engineering. The genetic algorithm not only provides an alternative method to solve the problem, but also consistently outperforms other traditional methods in most of the situations. During the last few decades, some of work reported by scientists including Andrey[2]; Bashir[3-5]; Chakraborty[6]; Dharmistha and Vishwakarma[7]; Goldberg[8]; Dana Bani[9]; Haldurai[10]; Katoch et el.[11]; and Jain[12].

The initial beginning of the evolutionary algorithm is to select the best individuals as parents from the population, making demand from them to reproduce to ultimate extend the generation. During reproduction, genes from both parents undergo crossover, and occasionally, an unintentional change occurs, known as mutation. Then the next generations are asked to reproduce their offspring and the process continues. The evolutionary algorithm is inspired on this theory of cross over and mutation where basically crossover is used to create new solutions from population's genetic information and mutation occurs to bring new information or maintain diversity within the population and prevent premature convergence to make the solution more generic. It is commonly used to find or near-optimal solutions to the problems from the search domain which otherwise would have taken a significant amount to solve.

Optimization is a very important tool in any business circle viz., finance, automobile or health care. The purpose of optimization is to find a point or set of points in the search domain by minimizing/ maximizing the loss/cost function that provides us the optimal solution for the problem in hand. Here, we try to minimize/ maximize the objective function f(x) subject to one/ multiple constraints like variables. In genetic algorithms, the points like population, chromosomes, gene, fitness values, crossover, mutation, evaluation of new population etc. are more important. These points include – i. Determine the number of chromosomes, generation, mutation rate and cross over rate value, ii. Generate chromosome - chromosome number of the population, and the initialization value of the genes chromosome- chromosome with random value, iii. Process steps 4-7until the number of generations is met, iv. Evaluation of fitness value of chromosomes by calculating objective function, v. Chromosomes selection, vi. Cross over, vii. Mutation, and viii. Solution (Best Chromosome).

Linear equation problem

In genetic algorithm chromosome coded as 0's and 1's, gene shall be represented with 2^i each genes lying in the chromosome, i.e., $2^0, 2^1, 2^2, 2^3, \ldots\ldots 2^i$.

Suppose, there is an equality $f(x) = \sum^{n-1} 2^i bi=0$ =k, say,
where k is an positive integer. The coefficients are chosen in such a way that it represents the bit of genetic algorithms. The genetic algorithm will be used to find the value of b_0,

b_1,b_2,b_3,…, and b_{n-1} that satisfy the above equation equal to k (say). Optimization is a very important concept in any business domain, it may be retail, finance, automobile or health care and its purpose is to find a point or set of points in the search space by minimizing/ maximizing the loss/cost function, that gives us the optimal solution for the problem in our hand. There are 5 phases in genetic algorithm which are as follows: (i). Initial population (ii). Fitness function (iii). Selection (iv). Crossover (v). Mutation.

Initial Population

This process starts with a set of individuals which is called a Population. An individual have it's own characteristics and theses individuals are known as Genes. Genes are combined into a string to form a Chromosome.

Fitness Function: The fitness function determines how fit an individual is it will survive in next generation or not. The fitness function plays a vital role in genetic algorithm. The fitness function gives score to each individual. The probability that an individual will be selected for next generation is based on its fitness score. The fittest ones will survive in next generation.

Selection: The idea of selection phase is to select the fittest individuals from the population and give approval to them for the next generation. Individuals are selected based on their fitness scores. Individuals with high fitness score have more chance to be selected for reproduction.

Crossover: Crossover is the most important phase in a genetic algorithm. A crossover point is randomly selected from the generation and Offspring are created by exchanging the genes of parents among themselves until the cross over point is reached. The new offspring are added to the population and new population will be generated.

Mutation: This process is used to maintain the diversity in the generation and it prevents premature convergence. In mutation genes are randomly replaced on a position with a new value. The algorithm terminates if the population has converged. On termination algorithm provides the optimal answer. The process of genetic algorithm is as follows: Step 1: Determine the number of chromosomes, generation, and mutation rate and crossover rate value for the population. Step 2: Generate chromosomes and initialization of values to the chromosomes. Step 3: Repeat steps 4-7 until the number of generations is met. Step 4: Calculation of fitness values of chromosomes by calculating the objective function. Step 5: Chromosomes selection. Step 6: Crossover. Step 7: Mutation. Step 8: Solution (Best Chromosomes).

Linear equality problem

So here is the example of applications of genetic algorithm to solve the simple mathematical linear equality problem. Suppose there is equality a + 2b + 3c + 4d+5e = 20, genetic algorithm will be used to find the value of a, b, c, d and e that satisfy the above equation for this problem the objective is minimizing the value of function f(x) where f(x) = ((a+2b+3c+4d+5e)-20). Since there are five variables in the equation, namely a,b,c,d and e we can compose the chromosome as follow: To speed up the computation, we can restrict that the values of variables a,b,c,a,d and e are integers between 0 and 20. Then, we define the number of chromosomes in population are 6, then we generate random value of gene a,b,c,d and e for 6 chromosomes.

First, we should formulate the objective function, $f(x)=b_0+2b_1+4b_2=10$. For minimizing the value of function f (x) where $f(x)= (b_0+ 2b_1+4b_2)–10)$. To speed up the computation, we can restrict that the values of variables b_0, b_1, b_2 are integers lying between 0 and 10.

Initialization: We Initialize the number of chromosomes in population as 6, then we generate random value of genes b_0, b_1,b_2 for 6 chromosomes as given below.
Chromosome [1]=[b_0,b_1,b_2]=[2;1;4]
Chromosome [2]=[b_0,b_1,b_2]=[1;2;3]
Chromosome [3]=[b_0,b_1,b_2]=[3;4;7]
Chromosome [4]=[b_0,b_1,b_2]=[2;1;6]
Chromosome [5]=[b_0,b_1,b_2]=[1;4;9]
Chromosome [6]=[b_0,b_1,b_2]=[2;5;8]

Evaluation: We compute the objective function value for each chromosome produced in initialization step as Objective function [1] = Abs (2 + 2*1 + 4*4- 10) =10 Objective function [2] = Abs (1 + 2*2 + 4*3 - 10) =7 Objective function [3] = Abs (3 + 2*4 + 4*7 - 10) = 29 Objective function [4] = Abs (2 + 2*1 + 4*6 - 10) = 18 Objective function [5] = Abs (1 + 2*4 + 4*9 - 10) = 35 Objective function [6]=Abs (2+2*5+4*8 – 10=34.
At the time of evaluation of objective function, if the values are found to be zero, then again a random is generated and function is re- determined.

Selection: The fittest chromosomes are the ones those have higher probability for selection at the next generation. To compute fitness probability we must compute the fitness of each chromosome. Fitness [1] = (1/Objective function [1] 1/10 = 0.1000 Fitness [2] = (1 / Objective function [2]) = 1/7 = 0.1429 Fitness [3] = (1/Objective function [3]) = 1/29 = 0.0345 Fitness [4] = (1/Objective function [4]) = 1/18=0.0556 Fitness [5] = (1/Objective function [5]) =1/35=0.0286 Fitness [6] = 1 / Objective function [6]) = 1/34 = 0.0294.

Total= 0.1000+0.1429+0.0345+0.0556+0.0286+0.0294=0.3910
The probability for each of the chromosomes is formulated by:
P[i] = Fitness[i] / Total
P[1]=0.1000/0.3910= 0.2558
P[2]=0.1429/0.3910= 0.3655
P[3]=0.0345/0.3910= 0.0882
P[4]=0.0556/0.3910= 0.1422

P[5]=0.0286/0.3910= 0.0731
P[6]=0.0294/0.3910= 0.0752

From the above probabilities it can be seen that Chromosome 2 that has the highest fitness, has highest probability to be selected for next generation chromosomes. For the selection process we use roulette wheel, for that we should compute the cumulative probability values: C[1] = 0.2558
C[2]=0.2558+0.3655=0.6213
C[3]=0.2558+0.3655+0.0882=0.7095
C[4]=0.2558+0.3655+0.0882+0.1422=0.8517
C[5]=0.2558+0.3655+0.0882 +0.1422+0.0731=0.9248
C[6]=0.2558+0.3655+0.0882+0.1422+0.0731+0.0752= 1.0000

After the computation of the cumulative probability of selection process using roulette - wheel, then the process is used to generate random Number R(i) being chosen randomly one at least from each cumulative category in the range 0-1. The range of R(i) is given as follows:
$0 \leq R(i) \leq 0.2558$
$0.2559 \leq R(i) \leq 0.6213$
$0.6214 \leq R(i) \leq 0.7095$
$0.7096 \leq R(i) \leq 0.8517$
$0.8518 \leq R(i) \leq 0.9248$
$0.9249 \leq R(i) \leq 1.0000$
R[1]=0.370
R[2]=0.193
R[3]=0.874
R[4]=0.771
R[5]=0.297
R[6]=0.657

If random number R [1] is greater than C [1] and smaller than C [2] then select Chromosome [2] as a chromosome in the new population for next generation:
New Chromosome[1]=Chromosome[2] New Chromosome[2]= Chromosome[1] New Chromosome[3]= Chromosome[5] New Chromosome[4]= Chromosome[4] New Chromosome[5]= Chromosome[2] New Chromosome[6]= Chromosome[3] New Chromosome in the population thus becomes: Chromosome[1] = [1;2;3] Chromosome [2]=[2;1;4] Chromosome [3]=[1;4;9] Chromosome[4]=[2;1;6] Chromosome [5]=[1;2;3] Chromosome [6]=[3;4;7].

Crossover: In this example, we use one-cut point for crossover, i.e. randomly chosen a position in the parent chromosome and then exchanging sub- chromosome. Parent chromosome which will mate is randomly selected and the number of mate Chromosomes is controlled using crossover-rate (cr) parameters. Pseudo-code for the crossover process is as follows:
Begin k←0; while (k<population) do R[k] ← random (0-1);
If (R[k]<cr) then select Chromosome [k] as parent; end; k=k+1; end; end; Chromosome k will be selected as a parent if R [k] <ρc). Suppose we set that the cross over rate as 25%, then Chromosome number k will be selected for cross over if random generated value for Chromosome k is below 0.25.

The process is as follows: First we generate a random number R as the number of population
R[1]=0.090
R[2]=0.168
R[3]=0.659
R[4]=0.995
R[5]=0.073
R[6]=0.236

For random number R above, parents are Chromosome [1], Chromosome [2], Chromosome [5] and Chromosome [6] will be selected for cross over. Chromosome [1] >< Chromosome [6] Chromosome [2] >< Chromosome [5] Chromosome [5] >< Chromosome [1] Chromosome [6] >< Chromosome [2]

After chromosome selection, the next process is determining the position of the cross over point. This is done by generating and om numbers between 1 to (length of Chromosome – 1). In this case, generated random numbers should be between 1 and 2. After we get the crossover point, parents Chromosome will be cut at cross over point and its gens will be inter changed. For example we generated 3 random number and we get
C[1]=1
C[2]=1
C[3]=1
C[4]=1

Then for cross over, cross over, parent's gens will be cutatgen number 1, e.g
Chromosome [1]>< Chromosome 6]
=[1;2;3]><[3;4;7] = [1;4;7]
Chromosome [2]><Chromosome[5]
=[2;1;4]><[1;2;3] = [2;2;3]
Chromosome [5]><Chromosome[1]
=[1;2;3]><[1;2;3] = [1;2;3]
Chromosome [6]>< Chromosome [2]
=[3;4;7]><[2;1;4] = [3;1;4]

Thus, Chromosome population after experiencing a cross over process: Chromosome [1] = [2;2;3]
Chromosome [2]=[2;2;3]
Chromosome [3]=[1;4;7]
Chromosome [4]=[2;2;3]
Chromosome [5]=[3;1;4]
Chromosome [6]=[3;4;7]

Mutation: The number of chromosomes that results in a population governed by the mutation rate (mr) parameter. It is process done by replacing the gen at random position with a fresh new value. We compute the total length of gen in the population. Then, the total length of gen is equal to Total gen=number of gen in Chromosome x number of population = 3x6 = 18. Mutation process is done by generating a random integer between 1 and total gen (1to18). If generated random umber is smaller than mutation rate (mr) variable then marked the position of gen in chromosomes. Suppose we define mr as

10%, it is expected that 10% (0.1) of total gen in the population that will be mutated: number of mutations = 0.1 * 18 = 1.8 ≈ 2. Suppose generation of random number yield 10 and 14 then the chromosome which have mutation are chromosome number 4 gen number 1 and chromosome 5ge number 2. The value of mutated genes at mutation point is replaced by random number between 0-10. Suppose generated random number are 1 and 0 then chromosome composition after mutation are:
Chromosome[1]=[2;2;3]
Chromosome[2]=[2;2;3]
Chromosome[3]=[1;4;7]
Chromosome[4]=[**2**;2;3]
Chromosome[5]=[3;**1**;4]
Chromosome[6]=[3;4;7]

We, then, can now evaluate the objective function after one generation: Chromosome [1] = [2;2;3]
Objective function [1] = Abs(2+2*2+4*3) -10)=8
Chromosome [2] = [2;2;3]
Objective function [2] = Abs(2+2*2+4*3) -10)=8
Chromosome [3] = [1;4;7]
Objective function [3] = Abs(1 +2*4+4*7)- 10) =27
Chromosome [4] = [2;2;3]
Objective function [4] = Abs(2+2*2+4*3) -10)=8
Chromosome [5] = [3;1;4]
Objective function [5] = Abs(3+2*1+4*4) -10)=11
Chromosome [6] = [3;4;7]
Objective function [6] = Abs (3+2*4+4*7)-10)=29

From the evaluation of above objective function, it seems that the objective function of new prepared chromosome has been decreasing, implies that we have better chromosome or solution compared with previous chromosome generation. The senew chromosomes will undergo the same process as the previous generation of chromosomes such as evaluation, selection, crossover and mutation and at the end it produce new generation of chromosome for the next iteration. This process will be repeated until a predetermined number of generations. For this example, after running 100 generations, best chromosome is obtained: Chromosome=[4;1;1]
This means that: $b_0=4, b_1=1, b_2=1$
If we use the number in the problem equation $b_0 + 2\, b_1 + 4\, b_2 = 4 + (2 * 1) + (4 * 1) = 10$.
We can see that the value of variable b_0, b_1 and b_2 generated by genetical growth can satisfy that equality of linear equation population in Table-1.

Maximizing a function of one variable

In order to generate the number and length of the chromosomes as 10 and 6 respectively, we consider the number of Bernoulli trials as 6 and signify head as 1 and tail as 0. Let us write the combination of 1,2,3,4, and 5 successes out of 6 Bernoulli trials, we have the following combinations: 1,2,3,4,5,6 for one success. For two successes as 12,13,14,15,16,23,24, 25,26,34, 35,36,45,46 and 56.

Table-1: Values of b_0, b_1 and b_2 in linear equation population.

b_0 value	$b_1$value	b_2 value
0	1	2
2	0	2
4	1	1
6	0	1
4	3	0
2	4	0
0	5	0
0	3	1
8	1	0
10	0	0
2	2	1
4	3	0
6	2	0

For three we have 123, 234, 345, 456,156, 126, 134, 245, 356, 146,125, 236, 145, 256, 136, 124, 235,3 46, 156 and 126. Similarly, we have four combinations as 1234, 2345, 3456, 1456, 1256, 1236, 1345, 2456, 1356, 1246, 1235, 2346, 1456, 1236 and 1256. For five combinations 12345, 23456, 13456, 12456, 12356, and 12346. Thus, we have 62 combinations. Out of these 62 combinations, ten Chromosomes having length 6 would have been randomly chosen in the form of binary number as 0 and 1and proceeded on the similar lines of Sharma et.el,[13] for maximizing the function f(x) = √x subject to the condition that $1 \leq x \leq 25$.

This example adapts the method of an example presented in Goldberg[8]. Consider the problem of maximizing the function
$f(x) = -\frac{x^2(x+1)}{100}$ 10 x
where x is allowed to vary between 0 and 31.
This function is displayed in Figure-1. To solve this using a genetic algorithm, we must encode the possible values of x as chromosomes. For this example, we will encode x as a binary integer of length 5. Thus the chromosomes for our genetic algorithm will be sequences of 0's and 1's with a length of 5bits, and have a range from 0 (00000) to 31 (11111). To begin the algorithm, we select an initial population of 10 chromosomes at random. We can achieve this by tossing a fair coin 5 times for each chromosome, letting heads signify 1 and tails signify 0. The resulting initial population of chromosomes is shown in Table-2. Next we take the x-value that each chromosome represents and test its fitness with the fitness function. The resulting fitness values are recorded in the third column of Table-2.

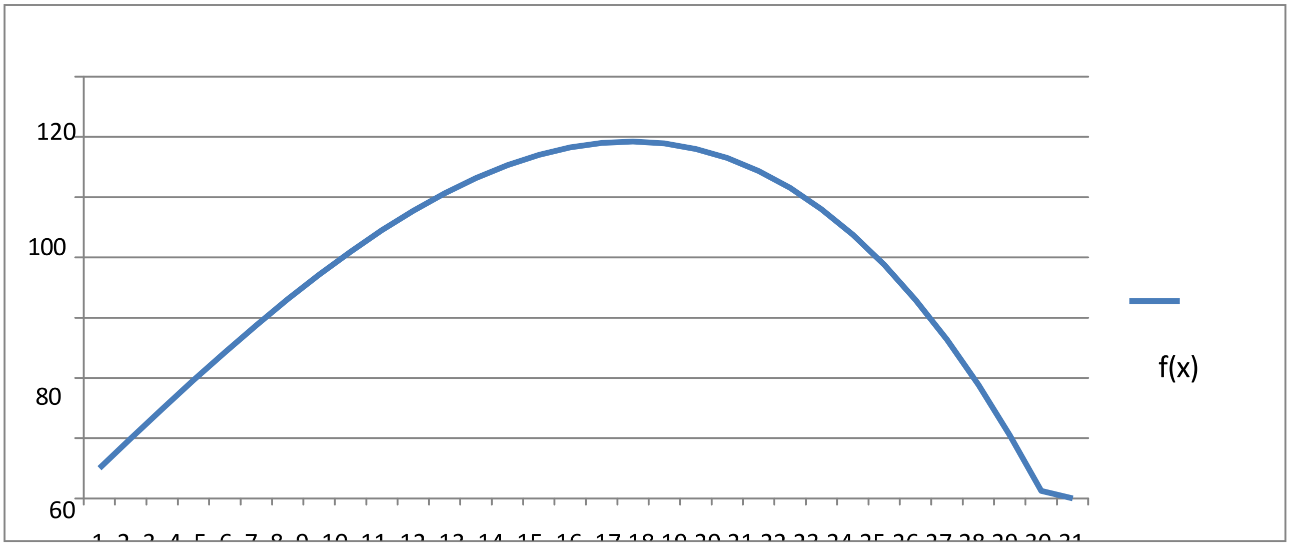

Figure-1: Graph of f(X) = $-\frac{x^2(x+1)}{100} + 10\,x$

Table-2: Initial Population.

Chromosome number	Initial population	x-value	Fitness value f(x)	Selection probability
1.	10000	16	116.48	0.0487
2.	01000	8	74.24	0.0310
3.	00100	4	39.20	0.0164
4.	00010	2	19.88	0.0083
5.	00001	1	9.98	0.0042
6.	11000	24	96.00	0.0401
7.	01100	12	101.28	0.0423
8.	00110	6	57.48	0.0240
9.	00011	3	29.64	0.0124
10.	10001	17	117.98	0.0493
11.	10100	20	116.00	0.0485
12.	01010	10	89.00	0.0372
13.	00101	5	48.5	0.0203
14.	10010	18	118.44	0.0495
15	01001	9	81.9	0.0342
16	11100	28	52.64	0.0220
17	01110	14	110.60	0.0462
18	00111	7	66.08	0.0276
19	10011	19	117.80	0.0492
20	11001	25	87.50	0.0366
21	11010	26	77.48	0.0324
22	01101	13	106.34	0.0444
23.	10110	22	108.68	0.0454
24	01011	11	95.48	0.0399
25	10101	21	112.98	0.0472
26	11110	30	21.00	0.0088
27	01111	15	114.00	0.0476
28	10111	23	103.04	0.0431
29	11011	27	65.88	0.0275
30	11101	29	37.70	0.0158
		Sum	2393.20	
		Average	79.7733	
		Max	118.44	

Table-3: Contined for re-production and second generation.

Chromosome number	Mating pairs	New population	X-value	Fitness value f(X)
1.	100|00	1 0 0 1 1	19	117.80
19.	100|11	1 0 0 0 0	16	116.48
7.	011 |00	0 1 1 0 1	13	106.34
22.	011|01	0 1 1 0 0	12	101.28
10.	10|001	1 0 1 1 0	22	108.68
23.	10|110	1 0 0 0 1	17	117.98
11.	101|00	1 0 1 0 1	21	112.98
25.	101|01	1 0 1 0 0	20	116.00
14.	100|10	1 0 0 1 1	19	117.80
27.	011|11	0 1 1 1 0	14	110.60
17.	01|110	0 1 1 1 1	15	114.00
28.	10|111	1 0 1 1 0	22	108.68
			Sum	1348.62
			Average	112.385
			Max	117.98

Conclusion

We select the chromosomes that will reproduce based on their fitness values, using the following probability:

$$\text{P (Chromosome produces)} = \frac{(\square\square)}{\sum_{\square=1}(\square\square)}$$

Goldberg has decided this process to spinning a weighted roulette wheel. Since our population has 30 chromosomes and each 'mating' produces 2 off spring, we need 15 matings to produce a new generation of 30 chromosomes. The selected chromosomes are displayed in Table-3 with the priority given on the basis of selection probability which has greater probability to be 0.04 20 and more. To create their off spring, across over point is chosen at random, which is shown in the table as a vertical line. It is to be noted that the sum of the chromosomes number selected format in g pairs is equal to 2393.20 from Table-2 while it is 1348.62 for fitness value f(X) in Table-3 for 12 chromosomes randomly selected with the priority of higher probability. It indicates that fitness value of the population has been increased at least after only one generation.

References

1. John, H. (1975). Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

2. Popov, A. (2005). Genetic algorithms for optimization. Programs for MATLAB. *User manual for MATLAB*.

3. Bashir, L. Z. (2015). Find Optimal Solution for Eggcrate Function Based on Objective Function. *World Scientific News*, (16), 53-72.

4. Bashir, Lubna Zaghlul and Raja Salih (2015). Solving Banana (Rosenbrock) function based on fitness function‖. *World Scientific News,* 6, 41-56.

5. Bashir, Lubna Zaghlul (2015). Solve simple linear equation using evolutionary algorithm. *World Scientific News*, 19, 148-167.

6. Chakraborty, R. C. (2010). Fundamentals of genetic algorithms. *Reproduction*, 22, 35.

7. Vishwakarma, M. D. D. (2012). Genetic algorithm based weights optimization of artificial neural network. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 1(3), 206-11.

8. Goldberg, D. (1989). Algorithms in search, optimization and machine learning. Reading. *MA: Addison-Wesley*.

9. Bani-Hani, D. (2020). Genetic algorithm (ga): A simple and intuitive guide. *Towards Data Science*.

10. Haldurai, L., Madhubala, T., & Rajalakshmi, R. (2016). A study on genetic algorithm and its applications. *Int. J. Comput. Sci. Eng*, 4(10), 139-143.

11. Katoch, Sourabh; Chauhan, S.S. & Kumar, Vijay (2021). A review on genetic algorithm: Past, present and future. *Multimedia tools and Applications*, 80(5), 8091-8126.

12. Jain, Shubham (2020). Introduction to Genetic Algorithm & Their Application in Data Science. Analytics Vidhya.

13. Sharma H.L., Shukla, Vijayshanker and Shukla, Varsha (2024). A modified genetic algorithms using balanced incomplete block designs and balanced ternary designs. *Gujarat Journal of Statistics and Data Science,* 40(1), 80-87.