ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

GENETIC ALGORITHM FOR OPTIMIZATION PROBLEMS

C. Premalatha
Assistant Professor, Department of Information Technology
Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India

ABSTRACT

Decision making features occur in all fields of human activities such as science and technological and affect every sphere of our life. Normally, any engineering problem will have a large number of solutions out of which some are feasible and some are non-feasible. The designer's task is to get best solution out of the feasible solutions. The complete set of feasible solutions constitutes feasible design space and progress towards the optimal design. In such a case, genetic algorithms are good at taking larger, potentially huge search space and navigating them looking for optimal combinations of things and solutions that may not be find in a life time. Genetic algorithm unlike traditional optimization methods processes a number of designs at same time, uses randomized operators that improves search space with efficient result. This paper dealt with important aspects of GA that includes definition of objective function, representation schemas for solution variables and randomized operators. These aspects drive the problem to optimal solution.

KEYWORDS: ***Genetic algorithm, optimization methods, objective function, representation schema, randomized operator and optimal solution.***

INTRODUCTION

The most significant aspect of optimization is to make the objective function a maximum or minimum. The following questions arise in this process

- Does an optimal solution exist?
- Is it unique?
- How sensitive the optimal solution is?
- How the solution behaves for small changes in parameters?

Optimization non-traditional search techniques are deterministic and stochastic. In the former algorithms such as steepest gradient methods are employed whereas in the stochastic approach random variables are used. Both gives high level of accuracy. Genetic algorithm comes under this non-traditional search technique which provides most efficient and accurate results.

Biological background

Cells are the building blocks or fundamental units of an organism. Each cell consists of a set of chromosome that serves as a model for the whole organism. A chromosome is formed through collection of DNA and on each block of DNA [5], [8] there exists a gene, which encodes a trait. e.g. hair color. Each gene has its own positions in a chromosome search space called as locus. Genome, a complete set of genetic material in which a particular set of genes is termed as genotype.

Fig.1.1 Structure of Genome

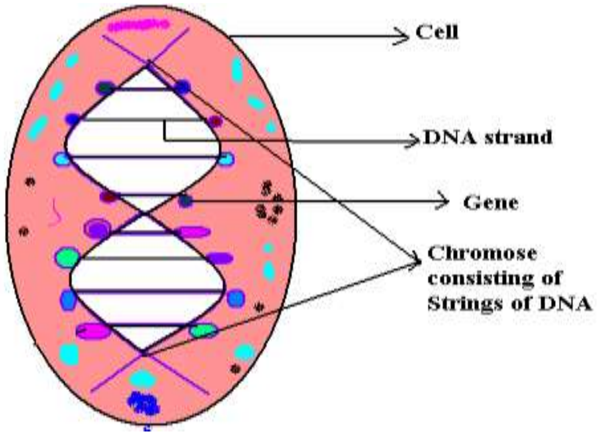


Fig.1.2 Chromosome with strings of DNA

Formation of offspring:
Recombination due to crossover is the main aspect of offspring formation[4], [6]. It is the process in which gee from parents form a new chromosome. In some cases, the element of DNA is modified through mutation. This modification happens by means of external factors like environmental changes, biological aspects and even due to errors in copying genes from parents.

Search space
Spaces [3] were all feasible solutions are placed. Solutions can be extreme (either maxima or minima) and they are represented by points in search space. In search space phenomenon, the start of solution and to look for solution is much more complicated. This is the reason for genetic algorithm preference in case of optimization problems

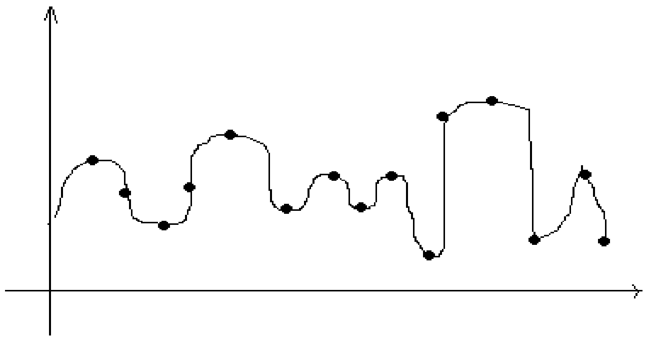


Fig.1.3 Search space

GENETIC ALGORITHM
Genetic algorithms [1], [5], [7] are computerized search and optimization algorithms based on the mechanics of natural genetics and natural science. It is a concept laid down on basis of Darwin's theory of survival of the fittest. It is started with a set of solutions called populations. Solutions are selected according to their fitness to form new population (offspring). This is repeated until some conditions for improvement of best solution are satisfied. The three most important aspect of GA are:

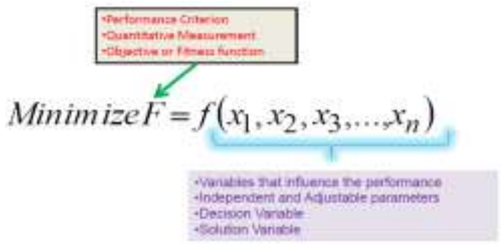
i) Definition of objective function
ii) Definition and implementation of genetic representation
iii) Definition and implementation of genetic operators

```
Algorithm
Start
{
Generate random population of
n chromosomes (suitable solutions for the problem)
}
Fitness
{
Evaluate the fitness f(x) of each chromosome x in the population
}
New population
{
Create a new population by repeating following steps until the new population is complete
[Selection] Select two parent chromosomes from a population according to their fitness
  (the better fitness, the bigger chance to be selected)
[Crossover] With a crossover parents form a new offspring (children). If no crossover
  was performed, offspring is an exact copy of parents.
[Mutation] With a mutation probability mutate new offspring at each locus (position in
  chromosome).
[Accepting] Place new offspring in a new population
Replace
{
Use new generated population for a further run of algorithm
}
Test
{
If the end condition is satisfied, stop, and return the best solution in current population
}
[Loop] Go to step Fitness
```

GENETIC ALGORITHM CRITERIA

Objective function

It involves in manipulation of solution or decision variables and produces a value based on which fitness populations are selected for recombination. The fitness function or objective function can be either maximized or minimized based on optimization criteria [2].


•Performance Criterion
•Quantitative Measurement
•Objective or Fitness function
Minimize F = f(x1, x2, x3,…,xn)
•Variables that influence the performance
•Independent and Adjustable parameters
•Decision Variable
•Solution Variable

Solution variable are the one that gives solution to an optimization problem. For e.g., consider

$$\text{Minimize } f(x) = x_1 + 2x_2 \text{ where } 0<x_1<3 \text{ and } 1<x_2<5$$

or

$$\text{Maximize } f(x) = x_1 + 2x_2 \text{ where } 0<x_1<3 \text{ and } 1<x_2<5$$

The variables x_1 and x_2 are also called Decision Variables since it decides the value of f(x).

Genetic representation schemas

There are many ways of representing individual genes. Holland (1975) worked mainly with string bits but arrays, tress, lists or any other objects can be used. Here, only bit strings are considered**.**

i. Binary encoding

It uses 1's and 0's for encoding chromosome strings. The length of the string is usually determined according to the desired solution accuracy. For example, 4-bit binary string can be used to represent 16 numbers as shown in Table II

Table I Bit strings representation in chromosome

Chromosome A	0110 0000 1110 1111
Chromosome B	1100 0011 1111 0000

Table II 4-bit strings and its numerical values

4 bit string	Numeric value	4 bit string	Numeric value	4 bit string	Numeric value	4 bit string	Numeric value
0000	0	0100	4	1000	8	1100	12
0001	1	0101	5	1001	9	1101	13
0010	2	0110	6	1010	10	1110	14
0011	3	0111	7	1011	11	1111	15

A 4-bit binary string can represent the integers from 0 to 15 and hence, (0000 0000) and (11111111) would represent the points for X_1 and X_2 as (X_1^L, X2^L); (X_1^U, X_2^U) because the substrings have minimum and maximum decoded values. Hence, an n-bit string can represent integers form 0to 2^{n-1} .The decoded value of a binary substring is calculated as

$$\sum_{k=0}^{k=ni-1} 2^k \mathrm{Sk}$$

Where S_i can be either 0 or 1 and the string S is represented as

$$S_{n-1}\ldots S_3S_2S_1S_0$$

If X_i^L X_j^U are given, the equivalent value for any 4-bit string can be obtained as follows:

$$X_i = X_i^L + ((X_j^U - X_i^L)/2^{ni}-1))*(\text{decoded value of string})$$

ii. Octal Encoding

A 4-bit octal string can represent the integers from 0 to 4095 and hence, (0000 0000) and (7777 7777) would represent the points for X_1 and X_2 as (X_1^L, X2^L);(X_1^U, X_2^U). The decoded value of a binary substring is calculated as

$$\sum_{k=0}^{k=ni-1} 8^k \mathrm{Sk}$$

The obtainable accuracy in that variable approximation is $(X_j^U - X_i^L)/8^{\,ni}$

iii. Hexadecimal Encoding

A 4-bit hexadecimal string can represent the integers from 0 to 65535and hence,(0000 0000)and (FFFF FFFF) would represent the points for X_1 and X_2 as (X_1^L,X_2^L);(X_1^U, X_2^U). The decoded value of a binary substring is calculated as

`

$$\sum_{k=0}^{k=ni-1} 16^k \mathrm{Sk}$$

The obtainable accuracy in that variable approximation is $(X_j^U - X_i^L)/16^{ni}$

iv. Permutation Encoding

In permutation encoding every numbers in string of chromosomes are sequenced. It is only useful for ordering problems.

Table III Sequenced strings representation in chromosome

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

v. Value Encoding

It includes chromosomes strings to be of numbers, real numbers, characters and objects. This type of encoding can be used in neural networks in which each value in chromosomes represents the corresponding weights.

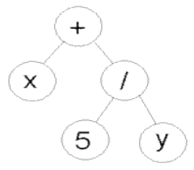
Table IV Varied strings representation in chromosome

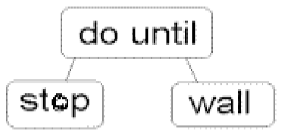
Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

vi. Tree Encoding

In this type of encoding, every chromosome is a tree of some objects such as functions and commands in programming language. Here, chromosomes are functions represented in trees.

Chromosome A **Chromosome B**





(+ x (/ 5 y)) (do_until stop wall)

Genetic selection operators

Chromosomes are selected from the population to be parents to cross over and produce offspring. As far as the fitness is concerned the mating chromosomes[6] has to be the fittest among all others. Reproduction is the first operator applied on fittest population for crossover and thus it is termed as selection operator.

The essential idea in all of them is that strings are picked from the current population and their multiple copies are inserted in mating pool in a probabilistic manner. The various methods in selecting chromosomes for parents to cross over are:

i. Roulette-wheel selection

In this selection, a string is selected from the mating pool with the probability proportional to the fitness. Thus, ith string in the population is selected with a probability proportional to F_i, where, F_i is the fitness value for that string. Since, the population size is kept fixed in simple GA; the sum of the probabilities of each string being selected for the mating pool must be one. The probability of the ith selected string is

$$P_i = F_i / (\sum_{j=1}^{n} Fj)$$

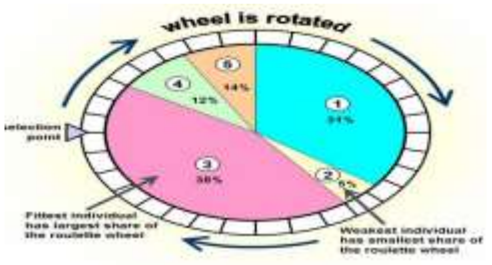

wheel is rotated
5
14%
4
12%
1
31%
selection point
3
38%
2
5%
Fittest individual has largest share of the roulette wheel
Weakest individual has smallest share of the roulette wheel

Fig. 3.1. Roulette wheel selection of the fittest

In Fig.3.1, Fittest individual that has largest share of 38% on the roulette wheel is selected for further crossover.

ii. Boltzmann Selection[5]

This method simulates the process of slow cooling of molten metal to achieve minimum function value in a minimization problem. The cooling phenomenon is simulated by controlling a temperature like parameter introduced with the concept of Boltzmann probability distribution so that a system in thermal equilibrium at a temperature T has its energy distributed probabilistically according to

$$\mathbf{P\ (E) = exp\ (-(E/kT)),\ k\text{- Boltzmann constant}}$$

This expression suggests that a system at a high temperature has almost uniform probability of being at any energy state, but at a low temperature it has a small probability of being at a high energy state. So, by controlling the temperature T and assuming search process follows Boltzmann probability distribution, the convergence of the algorithm is controlled.

Table V Individuals with its fitness value for mating

Individual	1	2	3	4
Fitness	**1**	**2.1**	**3.1**	**4.1**
Individual	**5**	**6**	**7**	**8**
Fitness	**4.6**	**1.9**	**2**	**4.5**

iii. Tournament Selection

Population diversity and selection pressure are the two most aspects that drives GA to fittest population selection. In such a case, roulette wheel does not favor selection pressure whereas tournament does it by holding a tournament competition[5] among 'N' individuals (frequency N=2).

The best individual from the tournament is the one with highest fitness which is the winner of N. The mating pool comprising of tournament winner has higher average population fitness. The fitness differences provide selection pressure, which drives GA to improve the fitness of succeeding genes. The following steps illustrate tournament selection strategy and fitness value.

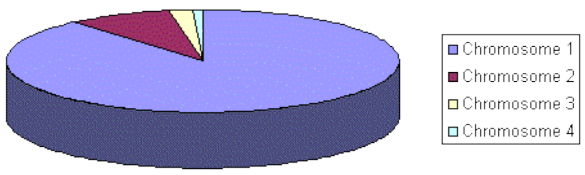
Step 1: Select individuals 2 and 4 at random, here 4 is the winner
Step 2: Select individuals 3 and 8 at random, here 8 is the winner
Step 3: Select individuals 1 and 3 at random, here 3 is the winner
Similarly, other population s are selected from the mating pool as 4 and 5 -> 5, 1 and 6-> 6, 1 and 2 -> 2, 4 and 2 -> 4 finally 8 and 3 -> 8.

From the above, the individuals 2, 3, 5 and 6 are chosen once, 4 and 8 are chosen twice, and 1 and 7 are not chosen at all. Thus for mating, all the individuals are considered once in tournament but in roulette wheel, it eliminates population with lowest fitness during first mate.

During the early genetic evolution process, there are large numbers of individuals or chromosomes that satisfy all constraints except one or two. A change in one or two design strings may produce a solution with higher fitness value. This means throwing out these solutions may result in a loss of some important information which might eventually lead to optimal solution.

iv. Rank Selection

The previous selection will have problems when the fitness differs very much. For example, if the best chromosome fitness is 90% of the entire roulette wheel then the other chromosomes will have very few chances to be selected.Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness ***1***, second worst ***2*** etc. and the best will have fitness ***N*** (number of chromosomes in population). The Fig 3.3 shows the situation change after changing the fitness to order number.



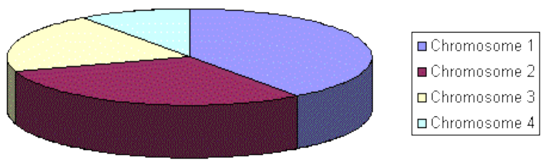


Fig.3.2 Situation before ranking (graph of fitness) ***Fig.3.3 Situation after ranking (graph of order numbers)***

After this all the chromosomes have a chance to be selected. But this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

v. Steady-State Selection

This is not particular method of selecting parents. Main idea of this selection is that big part of chromosomes should survive to next generation. In every generations, are selected a few (good - with high fitness) chromosomes for creating a new offspring. Then some (bad - with low fitness) chromosomes are removed and the new offspring is placed in their place. The rest of population survives to new generation.

vi. Elitism

Elitism is name of method, which first copies the best chromosome (or a few best chromosomes) to new population. The rest is done in classical way. Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution. The expression for the fitness becomes

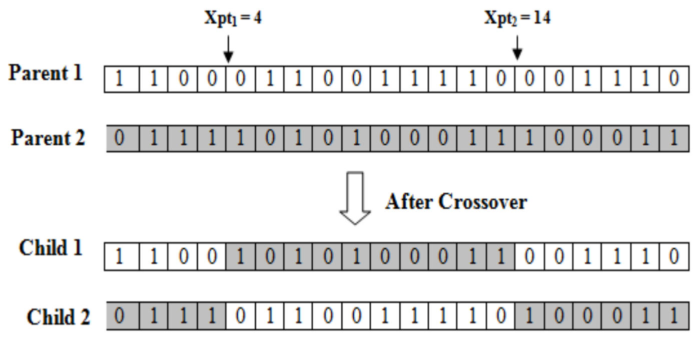
$$\Theta_i = (F_{max} - F_{min}) - F_i(X), \quad \text{for minimization problem}$$
$$\Theta_i = F_i, \quad \text{for maximization problem}$$

Genetic crossover operator

Individuals are selected form the selection operator whereas the mating between the individuals is done through cross over (recombination of good individuals). The two new offspring created from this mating are put into the next generation of the population. Cross over can be of following types:

i. Two point Crossover

- Choose two parents randomly
- Choose two crossover site randomly in any one parent
- Swap the binary string between the crossover site



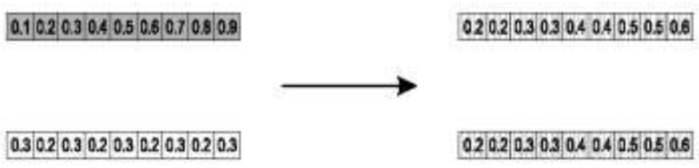
ii. Arithmetic Crossover

Most commonly used

- Parents: <x1,…,xn > and <y1,…,yn>
- child1 is:

$$\alpha \cdot \bar{x} + (1 - \alpha) \cdot \bar{y}$$

- reverse for other child. e.g. with α = 0.5



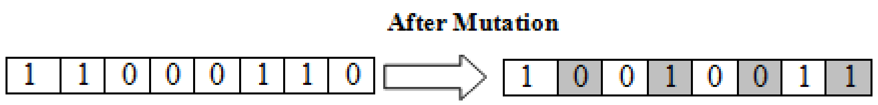
GENETIC MUTATION OPERATOR

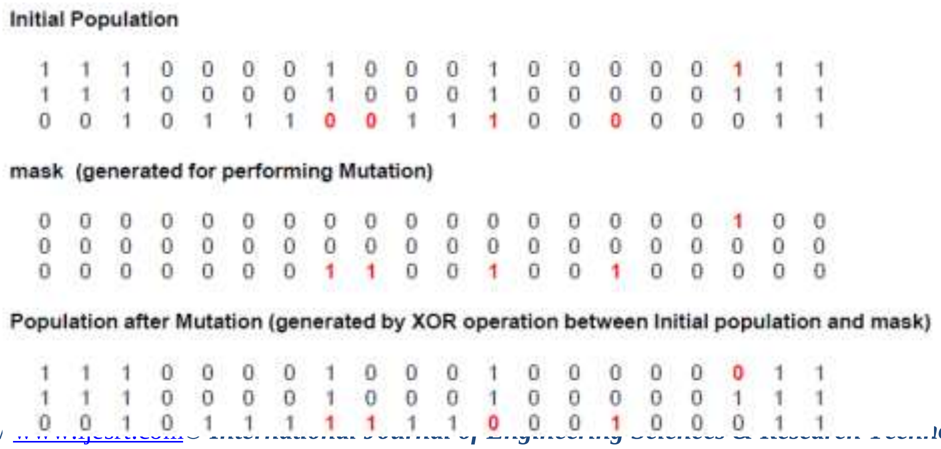
With some low probability, a portion of the new individuals will have some of their bits flipped. Its purpose is to maintain diversity within the population and inhibit premature convergence. Mutation alone induces a random walk through the search space. Mutation and selection (without crossover) create parallel, noise-tolerant, hill-climbing algorithms.

i. Bitwise Mutation[1]

The two step processes of bitwise mutation are:

- Choose few binary bits randomly
- Convert its value from 1 to 0 or from 0 to 1





Initial Population

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

mask (generated for performing Mutation)

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

Population after Mutation (generated by XOR operation between Initial population and mask)

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

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

ii. Uniform Mutation[2]

An individual is drawn randomly from the population and it is replaced by a new individual generated within its range.

$$\bar{x} = \langle x_1, \ldots, x_l \rangle \rightarrow \bar{x}' = \langle x_1', \ldots, x_l' \rangle$$

$$x_i, x_i' \in [LB_i, UB_i]$$

New individual = rand* (Max.Range - Min.Range) + Min.Range

Let 'rand'=0.7797, Max.Range=5, Min.Range-2

New individual = 0.7797*(5-2)+2 = 4.331

IMPLEMENTATION OF GENETIC ALGORITHM

Consider the following problem for optimization
Maximize f(x) = $x_1 + 2x_2$ where $0<x_1<3$ and $1<x_2<5$
Genetic algorithm can be implemented in the following steps:

Step 1: Representation of Solution Variables

Given: Maximize f(x) = $x_1 + 2x_2$ where $0<x_1<3$ and $1<x_2<5$

Solution variables: x_1, x_2

Commonly followed representation schemes are

- Binary Numbers (0 1 0 1 0 0 0 1 1 0…..)
- Real Numbers (1.2 5.4 3.8 7.9 …..)
- Integer Numbers (4 3 7 9 12 15…..)
- Character / Object (A C Y H B T …..)

Step 2: Random Initialization of set of possible values for solution variables

N'=number of possible values for the solution variables.
If N=5, then for f(x)= $x_1 + 2x_2$ where $0<x1<3$ and $1<x2<5$
For Binary Number Representation, with 4 bits for each variable,

x_1	x_2
0 1 1 0	0 1 0 1
1 0 1 0	1 1 1 0
0 0 0 0	1 0 1 0
1 1 1 1	1 1 0 0
0 0 1 1	1 0 1 0

For Real Number Representation,

x_1	x_2
1.2	1.9
0.3	2.5
2.8	3.4
1.7	4.5
0.6	3.8

Step 3: Evaluate each value of solution variable by calculating objective function

Linear Mapping Rule is used for Binary Representation whereas no mapping rule followed for real number representation.

$$X_i = X_i^L + ((X_j^U - X_i^L)/2^{ni}-1))*(\text{decoded value of string})$$

Table VI Objective Function Value

Encoded Value		Decoded Value		Objective Function Value
X_1	X_2	X_1	X_2	F(x)
0 1 1 0	0 1 0 1	1.2	2.33	5.86
1 0 1 0	1 1 1 0	2	4.7	11.46
0 0 0 0	1 0 1 0	0	3.66	7.33
1 1 1 1	1 1 0 0	3	4.2	11.4
0 0 1 1	1 0 1 0	0.6	3.66	7.93

Step 4: Display the best value of objective function

- If the problem is **maximize $f(x)= x_1 + 2x_2$** where 0<x1<3 and 1<x2<5, then the maximum value of f(x) is to be displayed.
- If the problem is **minimize $f(x)= x_1 + 2x_2$** where 0<x1<3 and 1<x2<5, then the minimum of f(x) or maximum of 1/f(x) is to be displayed.
- Objective function tests the fitness of the randomly generated value of the individual solution variable
  - For this example, the **Best Value is 11.46**

Step 5: Creating new set of possible values for solution variable by applying four major operations

Operation 1: Process of selecting the better two individuals and keeping it separately without allowing them to go for some other operations

Table VII Objective Function Value for Operation 1

Encoded Value		Decoded Value		Objective Function Value
X_1	X_2	X_1	X_2	F(x)
1 0 1 0	1 1 1 0	2	4.7	11.46
1 1 1 1	1 1 0 0	3	4.2	11.4

Operation 2: Process of producing more number of good values of solution variable and removing the bad value of solution variable

Table VIII Objective Function Value for Operation 2

Encoded Value		Decoded Value		Objective Function Value
X_1	X_2	X_1	X_2	F(x)
1 0 1 0	1 1 1 0	2	4.7	11.46
1 0 1 0	1 1 1 0	2	4.7	11.46
0 0 0 0	1 0 1 0	0	3.66	7.33
1 1 1 1	1 1 0 0	3	4.2	11.4
0 0 1 1	1 0 1 0	0.6	3.66	7.93

Operation 3: Process of producing the new value for the solution variable by swapping two already assigned value of the solution variable randomly.

Table IX Objective Function Value for Operation 3

Old value	New value	X_1	X_2	F(x)
1 1 1 1 1 1 0 0	1 1 1 1 1 0 1 0	3	3.66	10.33
0 0 1 1 1 0 1 0	0 0 1 1 1 1 0 0	0.6	4.22	9

Operation 4: Process of producing new value for the solution variable by reversing any one bit of the
already assigned value of the solution variable randomly

Table X Objective Function Value for Operation 4

Old value	New value	X_1	X_2	F(x)
1 1 1 1 1 0 1 0	1 1 1 1 1 1 1 0	3	4.7	12.46

Table XI New Set of Possible values for solutions variable

Encoded Value		Decoded Value		Objective Function Value
X_1	X_2	X_1	X_2	F(x)
1 0 1 0	1 1 1 0	2	4.7	11.46
1 0 1 0	1 1 1 0	2	4.7	11.46
0 0 0 0	1 0 1 0	0	3.66	7.33
1 1 1 1	1 1 1 0	3	4.7	12.46
0 0 1 1	0 0 1 1	0.6	4.22	9

Step 6: Do steps 3, 4 and 5 iteratively until no other improvement in the best value of objective function

Iteratively solving the problem results in best value **12.46**

Table XII Terminologies of Mathematics vs. Genetic algorithm

S.No.	Mathematical Terminology	Example	Genetic algorithms Terminology
1.	Solution variable	x1 (0010 or 2 or 5.4 or A)	Gene
2.	set of Solution variable	x1,x2 (0 1 1 0, 0 1 0 1 or 2,5 or 3.5,6.2)	Chromosome
3.	Set of Possible Values for the solution variable	0 1 1 0 0 1 0 1 1 0 1 0 1 1 1 0 0 0 0 0 1 0 1 0	Population
4.	Number of Possible Values for the		

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

	solution variable	N = 3 or 30 or 100…	Population Size
5.	Objective Function/Fitness Function	minimize $f(x)= x1 + 2x2$ where $0<x1<3$ and $1<x2<5$	Environment
6.	Operation 1 : Process of selecting the better two individuals and keeping it separately without allowing them to go for some other operations		Elitism
7.	Operation 2 : Process of producing more number of good value of solution variable and removing the bad value of solution variable		Selection/Reproduction
8.	Operation 3 : Process of producing the new value for the solution variable by swapping two already assigned value of the solution variable randomly.		CrossOver/Recombination
9.	Operation 4 : Process of producing new value for the solution variable by reversing any one bit of the already assigned value of the solution variable randomly		Mutation
10.	Iteration	I =100 or 300 or 500….	Generation

CONCLUSION

The concept of GA is ease, modular and good for noisy environment. GA produces better solutions, inherently parallel, easy to exploit for alternate solutions and supports multi-objective optimization. It can be used in the case of hybridization with an existing solution, need of exploratory tool to examine new approaches and for too complicated or too slow solutions. It has certain issues to be considered in which the most significant one is population size, selection, deletion policies, cross over, mutation, representation, termination criteria, performance and scalability.

REFERENCES

1. D. Devaraj, P.Ganesh Kumar, “Mixed Genetic Algorithm approach for Fuzzy Classifier Design”, International Journal of Computational Intelligence and Applications, Vol.9, No.1, pp.49-67, 2010.
2. Deb, k.(1999), An tnroduction to Genetic algoriths,sadhana, V0l.24, Parts 4 and 5, Aug and oct., pp. 293-315
3. Gen, M. and R. cheng(1997), Genetic Algorithm and Engineering Design, John Wiley, Newyork.
4. Goldberg, D.E. and J. Richardson (1987), Genetic Algorithms with sharing for multi-modal function Optimization, proc. Of the second intl. conf. on Genetic Algorithms, (Ed.) J.J. Greffenstette, pp. 41-49
5. S.Rajasekaran, G, A. vijayalakshmi pai “Neural networks, Fuzzy logic and Genetic algorithms synthesis and applications”, PHP learning pvt. Ltd., 201
6. http://lancet.mit.edu/`mbwall/presentations/Introto GA
7. http://lsll.www.informatik.uni-dortmund. /de /evenet /coordinator /resources /softwareanddemos .html
8. http://www.eng.buffalo.edu/Research/MODEL/wcsmo3/proceedings/numlist.html
9. J. Han and M. Kamber, (2001) ‘Data Mining, Concepts and Techniques’, 2nd ed. San Francisco, CA: Morgan Kaufmann.
10. C. Premalatha, D. Devikanniga, “Analysis of Cancer Gene Expression Profiling in DNA Microarray Data using Clustering Technique”, International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014, pp.131-137.

11. Premalatha, C., B. Suganyadevi, and S. Chitra. "Assessment of AP, STEMI, NSTEMI and therapy Prescription based on vascular age-A Decision tree approach", International Journal of Engineering Research and General Science Volume 2, Issue 2, Feb-Mar 2014,pp.56-67.
12. Premalatha, C., X. Rexeena, and S. Saranya. "Assessment of critical behavior for MI, PCI and CABG events.",Internationa journal of Scientific & Engineering Research, Volume 4, Issue 6, June-2013,pp.17-21